

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))


Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

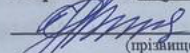
на тему:

«Інформаційна технологія надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем»

Виконав: студент 2-го курсу, гр. 1КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

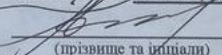
 Сапіташ В.С.
(прізвище та ініціали)

Керівник: д-р техн.наук, проф. каф. КН

 Іванчук Я. В.
(прізвище та ініціали)

« 07 » 12 2023 р.

Оponent: к.т.н., доцент каф. АІТ

 Коцюбинський В. Ю.
(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.
(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 Комп'ютерні науки
Освітньо - професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

Д.т.н. проф. Яровий А.А.

29.08

2023 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сапіташ Владислав Степанович

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

керівник роботи: д.т.н., професор кафедри КН Іванчук Я.В.

затвержені наказом вищого навчального закладу від «13» 09 2023 року № 244



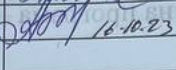

2. Строк подання студентом роботи 13 11 2023 р.

3. Вхідні дані до роботи: максимальна кількість елементів комплектуючих апаратного забезпечення – 7 од; максимальна вартість комплектуючих – 200 тис. грн.; максимальний рейтинг вказаного елемента апаратного забезпечення – 5 ум. од.;

4. Зміст текстової частини: вступ; аналіз предметної області; проектування інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем; програмна реалізація інформаційної технології формування рекомендацій для підбору апаратного забезпечення; економічна частина; висновок; перелік використаних джерел; додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Схема етапів розробки експертних систем; принципова схема функціонування системи нечіткого висновку; типова архітектура системи нечіткого висновку; графічне представлення функцій належності; схема алгоритму роботи інформаційної технології надання рекомендацій для підбору апаратного забезпечення; схема алгоритму роботи програмного модуля надання рекомендацій.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Іванчук Я. В., д.т.н., проф. каф. КН	 29.08.23	 10.11.23
4	Адлер О.О., доц. каф. ЕПВМ	 16.10.23	 20.10.23

7. Дата видачі завдання 29.08.2023 року

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ з/п	Назва та зміст етапу	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	01.09.23 - 01.09.23	
2	Проектування інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	08.09.23 - 24.09.23	
3	Програмна реалізація інформаційної технології формування рекомендацій для підбору апаратного забезпечення	25.09.23 - 15.10.23	
4	Економічна частина	16.10.23 - 20.10.23	
5	Апробація та/або впровадження результатів дослідження	21.10.23 - 05.11.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	

Студент


(підпис)

Сапігаш В.С.

Керівник роботи


(підпис)

Іванчук Я.В.

АНОТАЦІЯ

УДК 004.8

Сапіташ В.С. Інформаційна технологія надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем.

Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма– системи штучного інтелекту. Вінниця: ВНТУ, 2023. 112 с.

На укр. мові. Бібліогр.: 25 назв; рис.: 19; табл. 14.

Магістерська кваліфікаційна робота присвячена розробці інформаційної технології для надання рекомендацій з підбору апаратного забезпечення комп'ютерних систем. У загальній частині роботи були розглянуті та проаналізовані існуючі програмні рішення і їх функціональні можливості, досліджено предметну область вибору комп'ютерного обладнання, враховуючи потреби різних категорій користувачів.

Розроблено модель системи рекомендацій, яка включає прогнозування потреб користувача та оцінювання відповідності доступного обладнання. Запропоновано алгоритми нечіткого логічного виведення для визначення оптимального вибору на основі переліку технічних характеристик і вимог користувача. Після цього була розроблена програмна реалізація інформаційної технології з використанням мови програмування JavaScript і проведено її тестування.

Ілюстративна частина складається з семи плакатів та креслень. В економічному розділі наведено комерційний та технологічний аудит науково-технічної розробки, спрогнозовано витрати, проаналізовано системи-аналогів, розраховано економічну ефективність та комерційний потенціал розробки, який демонструє абсолютну ефективність вкладених інвестицій при періоді окупності близько 0,9 років.

Ключові слова: експертні системи, комп'ютерні комплектуючі, апаратне забезпечення, алгоритм

ABSTRACT

Sapitash V.S. Information technology for providing recommendations for the selection of computer system hardware. Master's thesis in the specialty 122 «Computer sciences», educational program «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 112p.

In Ukrainian. Bibliography: 25 titles; fig: 19; table. 14.

The master's thesis is devoted to the development of information technology for providing recommendations for the selection of computer system hardware. In the general part of the work, existing software solutions and their functionality were reviewed and analyzed, the subject area of computer hardware selection was investigated, taking into account the needs of different categories of users.

A model of the recommendation system is developed, which includes predicting user needs and assessing the suitability of available equipment. Fuzzy inference algorithms are proposed to determine the optimal choice based on the list of technical characteristics and user requirements. After that, a software implementation of the information technology was developed using the JavaScript programming language and tested.

The illustrative part consists of seven posters and drawings. The economic section provides a commercial and technological audit of the scientific and technical development, forecasts costs, analyzes analog systems, calculates the economic efficiency and commercial potential of the development, which demonstrates the absolute efficiency of the investment with a payback period of about 0.9 years.

Keywords: expert systems, computer components, hardware, algorithm.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Огляд проблемної області підбору апаратного забезпечення комп'ютерних систем	7
1.2 Аналіз сучасних методів підбору комплектуючих комп'ютерних систем	10
1.3 Аналіз методів надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	19
1.4 Висновок до розділу 1	27
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ДЛЯ ПІДБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ.....	29
2.1 Розробка методу формування рекомендацій для підбору апаратного забезпечення комп'ютерних систем	29
2.2 Розробка експертної системи надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	32
2.3 Розробка алгоритму надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	48
2.4 Висновок до розділу 2	52
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ДЛЯ ПІДБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ	53
3.1 Обґрунтування вибору засобів розробки програмного забезпечення	53
3.2 Програмна розробка інформаційної технології.....	56
3.3 Тестування інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	61

	3
3.4 Аналіз результатів тестування програми.....	65
3.5 Висновок до розділу 3	66
4 ЕКОНОМІЧНА ЧАСТИНА	67
4.1 Проведення комерційного та технологічного аудиту інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	67
4.2 Розрахунок витрат на здійснення науково-дослідної роботи	68
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	76
4.4 Висновок до розділу 4	80
ВИСНОВКИ.....	81
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
ДОДАТКИ.....	86
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	87
Додаток Б (обов'язковий) Лістинг програми.....	88
Додаток В (обов'язковий) Ілюстративна частина.....	101
Додаток Г (довідниковий) Інструкція користувача	109

ВСТУП

Актуальність теми дослідження. Тема використання інформаційних технологій для надання рекомендацій щодо підбору апаратного забезпечення комп'ютерних систем є дуже актуальною у сучасному світі. З розвитком технологій та збільшенням кількості доступних на ринку продуктів, користувачам все важче зробити правильний вибір, що відповідає їх потребам та бюджету. Згідно з даними, представленими різними дослідницькими агентствами, значна кількість користувачів втрачає час і ресурси через неправильний вибір апаратного забезпечення [1-3].

Процес підбору апаратного забезпечення включає в себе декілька етапів: вибір компонентів, вивчення їх характеристик, порівняння цін, врахування вимог до продуктивності і надійності. Основними проблемами цього процесу є відсутність єдиної бази даних з повною інформацією про компоненти та важкість отримання достовірних рекомендацій.

Економічний аспект підбору апаратного забезпечення є важливим як для організацій, так і для звичайних користувачів. Витрати на апаратне забезпечення можуть становити значну частину бюджету, особливо якщо вибір зроблено неправильно [4].

Впровадження інформаційної технології для підбору апаратного забезпечення може сприяти оптимізації витрат та покращенню ефективності використання комп'ютерних систем.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Метою магістерської кваліфікаційної роботи є підвищення ефективності надання рекомендацій підбору комплектуючих комп'ютерних

систем за допомогою розробки інформаційної технології в основі якої покладена система прийняття рішень.

Для досягнення поставленої мети, необхідно розв'язати такі основні **завдання:**

- провести аналіз предметної області процесу підбору апаратного забезпечення для комп'ютерних систем;
- проаналізувати існуючі методи надання рекомендації в експертних системах підбору апаратного забезпечення комп'ютерних систем;
- розробити метод формування рекомендацій для підбору апаратного забезпечення комп'ютерних систем;
- розробити систему надання рекомендацій підбору апаратного забезпечення для комп'ютерних систем на основі методу нечітких множин;
- розробити алгоритм функціонування програмного модуля апаратного забезпечення комп'ютерних систем;
- виконати програмну реалізацію інформаційної технології підбору апаратного забезпечення комп'ютерних систем;
- провести тестування програмного модуля апаратного забезпечення комп'ютерних систем;
- провести розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації.

Об'єкт дослідження – процес надання рекомендацій в експертних системах підбору апаратного забезпечення комп'ютерних систем.

Предметом дослідження є нечіткі моделі, алгоритми та програмні засоби оптимізації підбору апаратного забезпечення, зокрема через використання експертних систем.

Методи дослідження включають методи і моделі подання знань, теорію експертних систем, методи інженерії знань, що допомагають в розробці системи рекомендацій на основі штучного інтелекту, моделі та методи нечіткого логічного виведення, а також методи та підходи до розробки WEB-орієнтованих

програмних додатків.

Наукова новизна одержаних результатів: Розроблено математичну модель рекомендаційної системи, яка, на відміну від існуючих, базується на формалізованих, у вигляді нечітких баз знань, експертних правилах, що дає змогу визначати взаємозв'язок між параметрами комплектуючих апаратного забезпечення з якісними показниками комп'ютерних систем.

Практичне значення одержаних результатів полягає у наступному:

- розроблено алгоритм для прогнозування вимог до апаратного забезпечення на основі використання нечіткого підходу, який дозволяє уточнити та персоналізувати потреби користувача у комп'ютерному обладнанні.

- розроблено програмний модуль інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем. Це значно спрощує процес підбору апаратного забезпечення, роблячи його більш дружнім для користувача та ефективним з точки зору точності надання рекомендацій.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, глибоким аналізом стану справ в предметній області, порівнянням результатів з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок автора. Всі дослідження проведені в даній магістерській кваліфікаційній роботі. У тезах написаних у співавторстві, автору належать такі результати: [1] - Інформаційна технологія надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем.

Апробація результатів роботи. Результати були апробовані на науково-технічній конференції «LI Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023)» [1].

Публікації. За результатами досліджень опубліковано тези доповіді [1, 2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд проблемної області підбору апаратного забезпечення комп'ютерних систем

Сучасний світ інформаційних технологій постійно розвивається. За останні роки ми стали свідками небаченого розвитку в сфері апаратного забезпечення для комп'ютерних систем. Сучасний ринок апаратного забезпечення комп'ютерних систем є висококонкурентним і різноманітним, пропонуючи величезний вибір продуктів для споживачів. Нові моделі, з удосконаленими характеристиками і новими технологіями, постійно виходять на ринок, роблячи його ще більш насиченим. Це включає в себе не лише велику кількість нових пристроїв, що регулярно виходять на ринок, але й широкий спектр технологічних вдосконалень, які впливають на продуктивність, ефективність енергоспоживання та інші важливі характеристики обладнання [5].

Цей швидкий розвиток ринку може створювати виклики для користувачів, особливо тих, хто не має глибоких технічних знань. Комп'ютерне обладнання може мати багато технічних параметрів, які слід враховувати при виборі, включаючи, але не обмежуючись, продуктивністю, споживанням енергії, розмірами, вагою, сумісністю з іншими компонентами тощо. Така складність може утруднити процес прийняття рішень, особливо для недосвідчених користувачів.

Важливість правильного вибору апаратного забезпечення не можна недооцінювати. Вибір неправильного обладнання може призвести до підвищених витрат, зниженої продуктивності та інших негативних наслідків. Наприклад, вибір комп'ютера з недостатньою продуктивністю може обмежити здатність користувача виконувати певні завдання, такі як графічне моделювання або робота з великими наборами даних. Крім того, вибір недостатньо енергоефективного обладнання може призвести до підвищених витрат на електроенергію [6].

Апаратне забезпечення відіграє важливе значення в житті сучасної людини, тому важливо, щоб користувачі мали доступ до надійної, точної та актуальної інформації про доступне обладнання. Незважаючи на наявність інформації на веб-сайтах виробників та інтернет-магазинах, ця інформація часто є технічно складною, а зважаючи на великий обсяг доступних на даний момент комплектуючих користувачу складно зрозуміти, які саме комплектуючі йому потрібні для виконання необхідних йому завдань.

Отже, існує потреба в системах, що можуть аналізувати різні аспекти апаратного забезпечення і представляти цю інформацію в зручній для користувачів формі, допомагаючи їм прийняти обґрунтовані рішення про покупку.

Більше того, проблема вибору апаратного забезпечення ускладнюється тим, що у кожного користувача є своє індивідуальне розуміння того, який комп'ютер вважається "ефективним" чи "потужним". Для одного користувача "ефективним" комп'ютер – це той, який здатний запускати найновіші ігри на максимальних налаштуваннях, для іншого – це комп'ютер, який має високу енергоефективність та може працювати цілодобово без перерв. Ще хтось може вважати, що найкращий комп'ютер – це той, який має найменшу вагу та розміри для комфортної роботи в дорозі.

Ще одним важливим аспектом є різне сприйняття потужності комп'ютера серед різних користувачів. Для когось потужний комп'ютер – це та машина, яка здатна без затримок і лагів відтворювати вимогливі ігрові процеси. Але професіонал у сфері графічного дизайну або відеомонтажу може вважати той самий комп'ютер "повільним", якщо він не в змозі швидко обробляти великі графічні файли або відеоряд.

Ця відмінність у сприйнятті виникає через різні вимоги до роботи обладнання. Ігрові додатки, наприклад, можуть вимагати високої продуктивності від графічної карти, але при цьому не ставити жорстких вимог до об'єму оперативної пам'яті. У той час як програми для роботи з 3D-моделями потребують великих ресурсів від обох цих компонентів.

Комп'ютер, який відмінно справляється з офісними завданнями і переглядом відео в Інтернеті, може виявитися абсолютно неефективним для наукових розрахунків або рендерингу великих об'ємів даних. Водночас, потужний сервер, призначений для обробки великих даних, може бути "перебором" для звичайного користувача, який використовує комп'ютер виключно для розваг.

Отже, проблема полягає в тому, що один і той же комп'ютер може в одних ситуаціях демонструвати відмінну продуктивність, а в інших – бути недостатньо ефективним. Тому при виборі комп'ютерного обладнання важливо враховувати конкретні завдання, для яких він буде використовуватися.

Ці відмінності в особистих уподобаннях і потребах користувачів роблять задачу рекомендацій щодо вибору апаратного забезпечення ще більш складною. При виборі компонентів для свого персонального комп'ютера (ПК) користувачі часто опираються на особистий досвід, рекомендації друзів чи відгуки в Інтернеті. Однак, через різницю в індивідуальних потребах, те, що ідеально підходить одному користувачу, може бути неоптимальним вибором для іншого.

Тому необхідно розробляти системи рекомендацій, які б могли враховувати індивідуальні потреби та уподобання користувачів. Ці системи повинні бути гнучкими, адаптивними та здатними аналізувати великий обсяг інформації про продукцію, щоб допомогти користувачам знайти найкращий варіант, який відповідав би їхнім потребам і бюджету.

Для досягнення поставленої мети необхідно виконати наступні задачі:

1. Вивчення загальних принципів вибору комп'ютерних комплектуючих. Дослідження різних характеристик комплектуючих, таких як продуктивність, сумісність, ціна та інші фактори, що впливають на вибір.

2. Формулювання правил вибору комп'ютерних комплектуючих. Створення набору правил, які враховують різні критерії і параметри для визначення оптимального варіанту.

3. Вибір підходу до розв'язання задачі та його обґрунтування. Визначення методів нечіткої логіки, які будуть використовуватись для розробки системи вибору комплектуючих, та обґрунтування цього вибору.

4. Виконання структуризації даних. Розробка системи класифікації комплектуючих та створення бази даних, яка включатиме інформацію про різні моделі та їх характеристики.

5. Вибір моделі формалізації та його обґрунтування. Визначення моделі нечіткої логіки, яка найкраще підходить для задачі вибору комплектуючих, та обґрунтування.

6. Виконання формалізації проблеми. Побудова математичної моделі, яка враховуватиме нечіткість і неоднозначність при виборі комплектуючих.

7. Розробка структури програмного забезпечення для нечіткої інтелектуальної системи вибору комп'ютерних комплектуючих. Розробка модулів для збору та обробки вхідних даних.

8. Розробка алгоритму програмного забезпечення.

9. Виконання аналізу результатів тестування програми.

1.2 Аналіз сучасних методів підбору комплектуючих комп'ютерних систем

Розглянемо сервіси, які використовують користувачі для вибору комплектуючих для ПК:

1. Веб-сайти виробників комп'ютерного обладнання надають найбільш оновлену та достовірну інформацію про свої продукти. Зазвичай, вони мають розділи, присвячені кожному продукту, де можна знайти технічні специфікації, відомості про сумісність, керівництво користувача, драйвери та програмне забезпечення, а також загальні рекомендації щодо використання та обслуговування продуктів.

Наприклад, на сайті Intel можна знайти інформацію про всі процесори компанії, включаючи їх технічні характеристики, особливості та рекомендації

щодо сумісних комплектуючих [7]. Сайт AMD надає подібну інформацію для своїх процесорів та відеокарт, а на сайті NVIDIA можна знайти детальні дані про відеокарти цього виробника [8, 9].

Важливою особливістю веб-сайтів виробників є те, що вони часто надають відповіді на технічні запитання, FAQ та форуми підтримки, де користувачі можуть отримати допомогу в розв'язанні проблем з продуктами.

2. Інтернет-магазини, такі як Amazon, Rozetka, також є важливими джерелами інформації [10, 11]. Вони надають ціни продуктів, включаючи інформацію про акції та знижки, огляди користувачів, рейтинги продуктів, а також, в деяких випадках, додаткову інформацію від експертів.

Ці відгуки та рейтинги можуть допомогти користувачам зрозуміти, як продукт веде себе в реальних умовах, а не лише на основі специфікацій виробника. Деякі сайти дозволяють користувачам задавати питання про продукт, які інші користувачі або продавці можуть відповісти.

Особливість використання онлайн магазинів полягає в тому, що вони допомагають користувачам зрозуміти реальний вартісний контекст продуктів, порівняти ціни між різними продавцями, а також оцінити доступність продукту на ринку.

3. Форуми комп'ютерних технологій та комп'ютерного апаратного забезпечення - це динамічні онлайн-спільноти, де користувачі обмінюються знаннями та досвідом щодо продуктів. Це включає в себе підсумки та дискусії щодо використання, підтримки, налаштування, усунення неполадок та оптимізації апаратного забезпечення комп'ютерних систем. Важливість форумів полягає в тому, що вони дають користувачам можливість отримувати реальні відгуки від інших користувачів.

Форуми, такі як Reddit мають велику кількість активних учасників, які діляться своїми відгуками, вподобаннями, досвідом та навіть досвідом усунення неполадок. Ці форуми дозволяють користувачам ставити запитання, отримувати відповіді, і навіть отримувати рекомендації щодо купівлі від досвідчених користувачів [12].

4. Веб-сайти оглядів - це ще один важливий ресурс для вибору комп'ютерних комплектуючих. Вони надають професійні та детальні аналізи продуктів, включаючи бенчмарк-тести, тести на надійність, порівняльні аналізи та ін. Сайти, такі як TechRadar, PCMag, та ін., проводять випробування продуктів в контрольованих умовах і надають об'єктивні відгуки, що дозволяє користувачам робити інформовані рішення про покупку [13, 14]. Однак, варто зауважити, що деякі з цих сайтів можуть отримувати фінансування від виробників або рекламодавців, тому важливо критично ставитися до такої інформації та звертатися до кількох джерел для найбільш об'єктивного розуміння.

5. Онлайн конфігуратори ПК – це веб-сервіси, що дозволяють користувачам вибрати та спланувати комплектуючі для будування або модернізації свого персонального комп'ютера. Ці інструменти надають зручний інтерфейс для вибору комплектуючих з широкого спектра виробників та моделей, а також перевіряють їх сумісність між собою, що допомагає уникнути можливих конфліктів.

6. Програми для бенчмаркінгу - це спеціалізовані програми, розроблені для оцінки продуктивності різних компонентів комп'ютера. Ці інструменти виконують серію різноманітних тестів, щоб визначити швидкість, потужність, ефективність та стабільність обладнання. Бенчмарки використовуються як в індустрії, так і серед споживачів для вимірювання та порівняння продуктивності обладнання.

Бенчмаркінг використовується для вимірювання та оцінки продуктивності різних комплектуючих, таких як процесори (CPU), графічні процесори (GPU), оперативна пам'ять (RAM), диски твердого стану (SSD), та ін. Це допомагає користувачам визначити, наскільки ефективно обладнання може обробляти різні задачі, включаючи обчислення, графіку, обробку даних, та ін.

Бенчмаркінг також використовується для порівняння різних моделей обладнання. За допомогою стандартних тестів можна оцінити, яка модель видає найкращу продуктивність в різних умовах використання. Це може бути особливо

корисно при виборі нового обладнання або оновленні існуючого.

Окрім оцінки та порівняння, програми для бенчмаркінгу можуть бути використані для перевірки, чи ваше обладнання працює на максимальній продуктивності. Наприклад, якщо ви встановили нову графічну карту або оновили драйвери, ви можете використати бенчмарк для перевірки, чи вони правильно встановлені та налаштовані. та бюджет користувача. Але вищезгадані сервіси мають також і недоліки:

1. Веб-сайти виробників:

- Перебільшення щодо якості своєї продукції (виробники можуть надавати інформацію в спосіб, який представляє їх продукти в найкращому світлі, часто приховуючи потенційні проблеми або слабкі сторони).
- Актуальність даних (іноді сайти виробників не оновлюються належним чином, що може призвести до неповної або застарілої інформації);
- технічний жаргон (інформація на сайтах виробників часто написана технічною мовою, яка може бути незрозумілою для звичайних користувачів).

2. Інтернет-магазини:

- Об'єктивність відгуків (часом відгуки можуть бути сплаченими або суб'єктивними, також існують випадки, коли конкуренти або самі продавці пишуть неправдиві відгуки);
- Недостатня технічних опис (іноді описи продуктів в онлайн-магазинах можуть бути неповними або занадто спрощеними);
- Наявність реклами (онлайн-магазини часто просувають певні продукти, які можуть бути не найкращим вибором для користувача).

3. Форуми:

- Відсутність модерації (Не всі форуми мають строгу модерацію, що може призвести до поширення неточної або недостовірної інформації);
- Суб'єктивність (Користувачі форумів висловлюють свої особисті думки, які можуть бути забарвлені особистим досвідом або упередженнями).
- Пошук інформації (У зв'язку з великою кількістю дописів і тем на

форумах, може бути складно знайти потрібну інформацію).

4. Веб-сайти оглядів:

- Зацікавлена сторона (Не всі сайти оглядів незалежні. Деякі з них можуть отримувати спонсорську підтримку або інші вигоди від виробників, що може впливати на об'єктивність їхніх рецензій);
- Нерівномірна якість оглядів (Якість аналізу може варіюватися від одного рецензента до іншого, а також від одного сайту до іншого);
- Поверхневність аналізу (Деякі огляди можуть бути занадто поверхневими та не забезпечувати детального аналізу продукту).

5. Онлайн конфігуратори ПК:

- Обмежена сумісність компонентів (Не всі можливі комбінації компонентів можуть бути представлені в конфігураторі, що може обмежити вибір користувача).
- Актуальність інформації (Ціни та наявність компонентів можуть швидко змінюватися, і конфігуратор може не завжди відображати актуальну інформацію).
- Відсутність практичних рекомендацій (Конфігуратор може підказувати компоненти на основі технічних характеристик, але не завжди надає рекомендації з точки зору практичного використання та сумісності компонентів).

6. Програми для бенчмаркінгу:

- Відсутність стандартизації (Різні програми для бенчмаркінгу можуть використовувати різні тести і методології, що може ускладнити порівняння результатів).
- Застосування в реальних умовах (Результати бенчмаркінгу можуть не завжди відображати реальну продуктивність системи під час її звичайного використання).
- Потенційні технічні проблеми (Деякі програми для бенчмаркінгу можуть бути несумісними з певними системами або компонентами, або ж можуть спричинити непередбачені технічні проблеми).

Усі ці ресурси можуть бути корисними для вибору комп'ютерних

комплектуючих. Однак, вони мають певні обмеження та можуть вимагати від користувача значних зусиль і технічних знань. Тому існує потреба в більш автоматизованому та орієнтованому на користувача рішенні для вибору апаратного забезпечення, яке б враховувало індивідуальні потреби та бюджет користувача, а також яке б не вимагало глибоких технічних знань в даній області.

Серед розглянутих підходів функціонально найближчими є онлайн-конфігуратори ПК, тому розглянемо їх детальніше.

PCPartPicker є веб-сервісом, що дозволяє користувачам вибрати та спланувати комплектуючі для будівництва або модернізації свого персонального комп'ютера (рис. 1.1). Інструмент надає зручний інтерфейс для вибору комплектуючих з широкого спектра виробників та моделей, а також перевіряє їх сумісність між собою, що допомагає уникнути можливих конфліктів.

PCPartPicker надає користувачам змогу вибрати компоненти з численних міжнародних та місцевих ритейлерів, переглядати останні ціни та перевіряти сумісність компонентів [15].

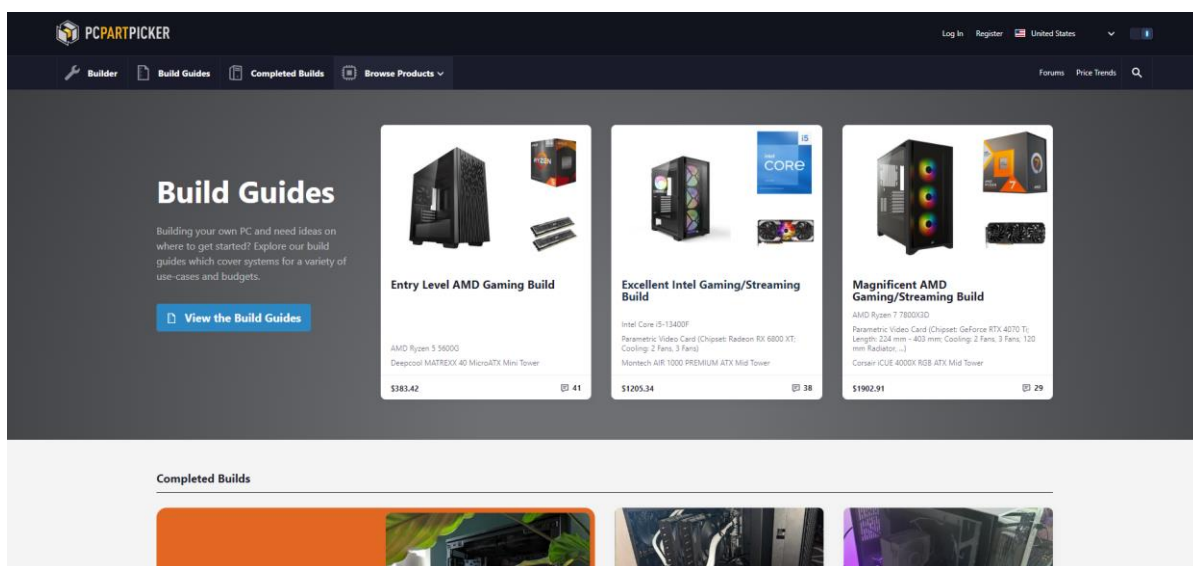


Рисунок 1.1 – Загальний вигляд інтерфейсного вікна сервісу PCPartPicker

Інший популярний сервіс – BuildMyPC (рис. 1.2). Він пропонує схожі функції, але має деякі унікальні особливості, такі як підбір компонентів на основі бюджету, наявності або специфічних брендів. Він також має розділ

"Інструменти", який включає в себе калькулятор потужності, перевірку сумісності RAM та ін [16].

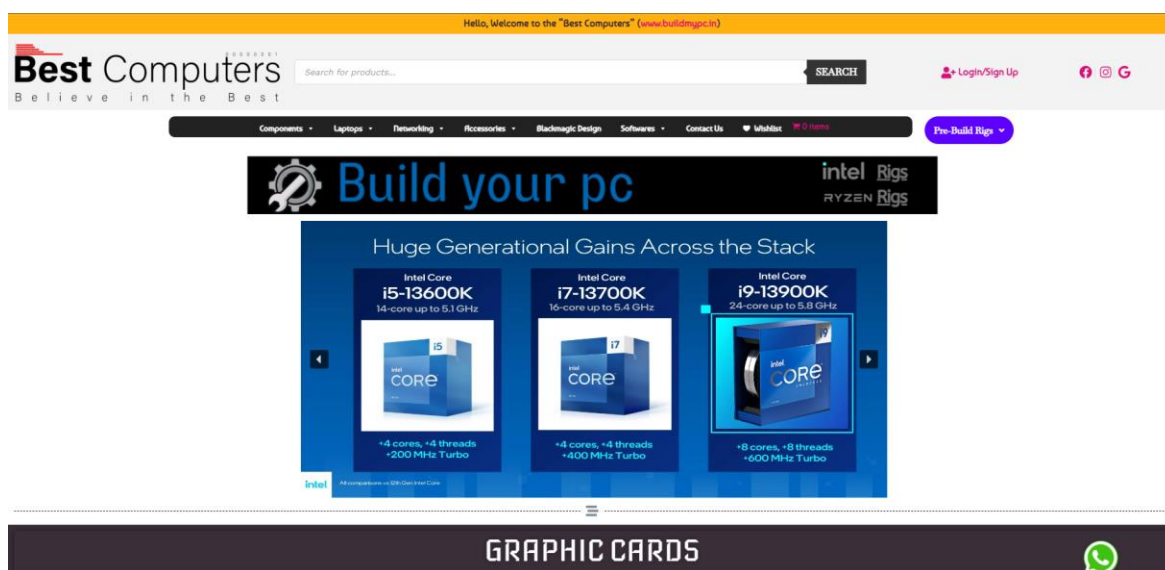


Рисунок 1.2 – Загальний вигляд інтерфейсного вікна сервісу BuildMyPC

Origin PC - це відомий виробник комп'ютерів та ноутбуків, який спеціалізується на високопродуктивних геймерських та професійних системах . Вони пропонують широкий вибір налаштувань для своїх систем, дозволяючи користувачам вибрати та налаштувати компоненти відповідно до своїх вимог та бюджету.

Сайт Origin PC має інтуїтивно зрозумілий конфігуратор систем, який дозволяє користувачам вибирати з базових моделей систем та налаштовувати їх за допомогою вибору різних компонентів (рис. 1.3). Користувач може вибрати процесор, графічний адаптер, оперативну пам'ять, накопичувач та інші компоненти. Конфігуратор також автоматично перевіряє сумісність компонентів, що допомагає уникнути потенційних проблем .

Окрім конфігурації обладнання, Origin PC також пропонує вибір різних опцій корпусу, включаючи різні кольори та дизайни. Вони також пропонують різні послуги, такі як ручне розгонювання, професійне кабельне керування, та ін.

Слід зазначити, що ціна систем в Origin PC може бути вищою, ніж при самостійній збірці, але вони пропонують високу якість зборки, гарантійне

обслуговування та зручність для тих, хто не бажає самотійно збирати свій комп'ютер [17].

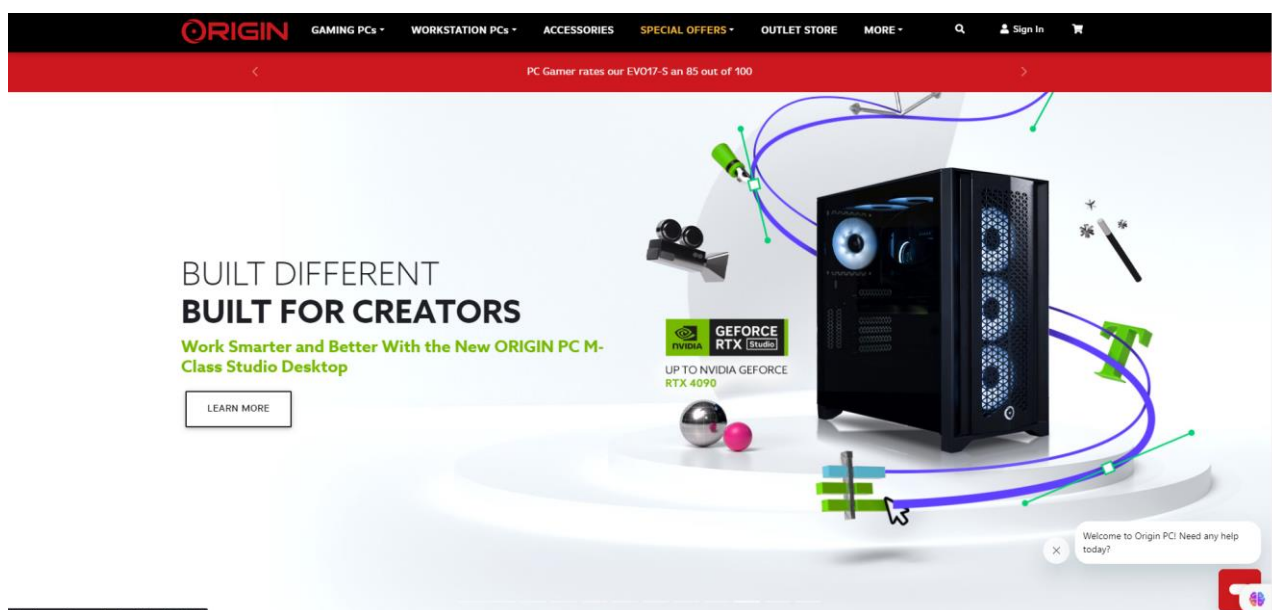


Рисунок 1.3 – Загальний вигляд інтерфейсного вікна сервісу Origin PC

У таблиці 1.1 було вказано відомі програмні рішення для підбору апаратного забезпечення комп'ютерних систем

Таблиця 1.1 – Порівняння відомих програмних рішень підбору апаратного забезпечення комп'ютерних систем

Назва системи	Переваги	Недоліки
PCPartPicker	<ul style="list-style-type: none"> - велика база даних компонентів - перевірка сумісності - інтеграція з ритейлерами - зручний інтерфейс 	<ul style="list-style-type: none"> - не включає різні додаткові послуги, такі як збірка - не може гарантувати актуальність цін

Продовження таблиці 1.1

Origin PC	<ul style="list-style-type: none"> - високоякісна збірка - інтуїтивно зрозумілий інтерфейс - широкий функціонал 	<ul style="list-style-type: none"> - ціна систем може бути вищою, ніж при самостійній збірці. - невеликий вибір компонентів
BuildMyPC	<ul style="list-style-type: none"> - підбір компонентів на основі бюджету, наявності або специфічних брендів. - доступ до великої бази даних для порівняння та вибору. 	<ul style="list-style-type: none"> - немає функції для зберігання та поділу списків комплектуючих. - можливі помилки в оцінці бюджету або потужності

Зазначені інструменти дозволяють користувачам спланувати та зібрати свій власний ПК, обрахувати приблизну вартість та уникнути помилок у виборі компонентів. Вони є важливими інструментами для будь-кого, хто цікавиться збіркою власного ПК. Але у них також є свої недоліки, тому було прийнято рішення про розробку конфігуратора для PC, принцип роботи якого базувався б на експертних системах, був простий та зручний у використанні навіть для недосвідчених користувачів за рахунок використання експертних систем для надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем.

Основні вимоги до інформаційної системи для підбору апаратного забезпечення:

1. Розуміння користувацьких потреб: Система має вміти аналізувати інформацію від користувача про його потреби, включаючи тип використання (наприклад, ігри, офісні роботи, професійне відео або графічне моделювання), бюджет, потреби в енергоефективності та інші важливі фактори.

2. Актуальна база даних апаратного забезпечення: Інформаційна система повинна мати доступ до актуальної бази даних компонентів апаратного

забезпечення з детальною інформацією про їх характеристики, ціни та сумісність між собою.

3. Алгоритми налізу: Система має використовувати алгоритми для порівняння та відбору оптимальних компонентів на основі введених користувачем параметрів.

4. Сумісність з різними платформами: Інформаційна система повинна бути доступна для використання на різних платформах і пристроях, включаючи настільні комп'ютери, планшети та мобільні телефони.

5. Користувацький інтерфейс: Система повинна мати інтуїтивно зрозумілий користувацький інтерфейс, який забезпечує простоту використання технології навіть для людей, які не мають технічних знань.

1.3 Аналіз методів надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Зростаючий обсяг доступної інформації ускладнює можливість її ефективної ручної обробки та засвоєння людиною, що стимулює розробку комп'ютерних засобів для автоматизації роботи з інформацією, включно з її збереженням, упорядкуванням, пошуком та обробкою. Значний інтерес представляють системи, здатні синтезувати та аналізувати інформацію, адаптуватися до змін, взаємодіяти з користувачами природною мовою та приймати рішення, незважаючи на неповноту чи суперечливість даних. Це інтелектуальні системи, що базуються на знаннях. Експертні системи, як підклас цих систем, використовують спеціалізовані знання у певній області для ефективного рішення задач на професійному рівні. Варто відзначити, що хоча всі експертні системи є заснованими на знаннях, не всі інтелектуальні системи є експертними. Інтелектуальні системи - це широкий клас систем, які виявляють інтелектуальну поведінку через використання евристик; системи на основі знань відділяють ці знання від інших компонентів системи; експертні системи - підклас, орієнтований на розв'язання складних життєвих завдань. Експертні

системи використовуються для задач, які є недостатньо добре визначеними або вимагають символічного міркування, мають нечіткі цілі, не мають відомого алгоритмічного рішення, або не можуть бути реалізовані через обмеження ресурсів. Ці задачі часто характеризуються неточністю, неповнотою, неоднозначністю та конфліктністю вихідних даних та знань.

Перевагами експертних систем є:

- Загальна доступність. Експертні системи дозволяють масово поширювати експертні знання, надаючи доступ багатьом користувачам.
- Тривале збереження знань: Знання, зібрані в експертних системах, зберігаються на тривалий час, на відміну від людської експертизи, яка може з часом слабнути або зникати через різні обставини, такі як відхід на пенсію або втрату працівників.
- Легкість розповсюдження знань. Передавання знань між людьми вимагає часу та витрат, в той час як копіювання та розповсюдження інформації в експертних системах є простим та ефективним процесом.
- Консистентність результатів. На відміну від людей, які можуть приймати різні рішення в однакових обставинах через емоційні чи фізичні фактори, експертні системи пропонують стабільні та незмінні відповіді.
- Інтеграція експертних знань. За допомогою експертних систем можна зібрати та використовувати знання від різних експертів, працюючи над задачами неперервно та ефективно, що може перевищити рівень знань одного експерта.
- Швидкість реакції. Експертні системи здатні реагувати швидше, ніж люди, особливо у випадках, що вимагають термінових рішень.
- Підвищена надійність: Використання експертних систем забезпечує додаткову достовірність рішень, особливо у випадку розбіжностей між думками різних експертів.
- Можливість обґрунтування рішень. Експертні системи можуть детально пояснити логіку своїх висновків, на відміну від людей, які можуть бути не здатні або не бажати це робити.

- Застосування в освіті. Експертні системи можуть служити як інтерактивні навчальні інструменти, передаючи знання та методику роботи.
- Формалізація та перевірка знань. У процесі створення експертних систем знання експертів перетворюються на чіткі форми, що дозволяє перевіряти їх на точність, послідовність та повноту.
- Інтелектуальний доступ до баз даних. Експертні системи можуть функціонувати як розширені бази даних з зручним інтелектуальним доступом.
- Економічність. Хоча початкова вартість розробки експертних систем може бути високою, їх експлуатація є відносно дешевшою порівняно з витратами на висококваліфікованих експертів.

Експертні системи погано вміють:

- Обмежені в здійсненні часового та просторового аналізу. Вони виявляються неефективними у роботі з часовими та просторовими відносинами, використанні здорового глузду для розуміння, визначенні власних обмежень компетентності та обробці суперечливої інформації.
- Недоліки інструментів побудови експертних систем. Інструментарій для створення експертних систем часто має обмеження в набутті нових знань, удосконаленні баз знань та використанні різних схем представлення знань.
- Складність побудови систем для кінцевого користувача, що не є екпертом у даній предметній області. Зазвичай кінцеві користувачі, які не мають глибоких знань у проблемній області, не можуть самостійно створювати експертні системи.
- Необхідність залучення людського експерта. Побудова ефективної експертної системи вимагає участі людини-експерта для надання та формалізації знань.
- Складність співпраці експертів з фахівцями-когнітологами. Можуть виникнути труднощі в процесі взаємодії експертів з когнітологами, які займаються формалізацією отриманих знань.
- Потреба в переробці програмного забезпечення. Іноді існуючі інструменти та моделі представлення знань у експертних системах виявляються

непридатними для конкретної проблемної області, що вимагає їхньої повної переробки.

Часові витрати на витягування та формалізацію знань: Процеси отримання знань від експертів, їх формалізації та перевірки на несуперечність можуть бути тривалими та складними.

Експертні системи вимагають значних інвестицій та людських зусиль. При спробах вирішення завдань, які є занадто складними, погано зрозумілими або непридатними для наявної технології, можуть виникнути витратні та невдалі результати. Ефективність застосування експертної системи залежить від можливості, доцільності та виправданості її розробки. Розробка експертної системи є можливою, якщо: задача не потребує загальновідомих знань, базується на інтелектуальних навичках, експерти можуть чітко описати свої методи, існують кваліфіковані експерти з узгодженими думками, задача не є надзвичайно складною, зрозумілою і має чітку структуру без необхідності використання здорового глузду або широких загальних знань про світ.

Розробка експертної системи є виправданою у випадках, коли: рішення є високорентабельним, існує ризик втрати людського досвіду, експертів недостатньо, досвід потрібен одночасно в різних місцях, досвід необхідний у умовах, непридатних для людини, або коли людський досвід відсутній в критичних ситуаціях.

Розробка експертної системи є виправданою у випадках, коли: рішення приносить високу віддачу, існує ризик втрати людського досвіду, експертів недостатньо, досвід потрібен одночасно в різних місцях, досвід необхідний у умовах, непридатних для людини, або коли людський досвід відсутній в критичних ситуаціях.

Інструменти для створення експертних систем – це програмні засоби, які використовуються інженерами знань або програмістами. Відрізняючись від звичайних мов програмування, ці інструменти спрощують представлення складних концепцій. Важливо розрізняти між інструментом створення та самою експертною системою. Інструменти включають мови для доступу та

представлення знань, а також допоміжні програми для взаємодії з компонентами, що вирішують проблеми.

Підхід «швидкого прототипування» у розробці експертних систем полягає у створенні первісних моделей, які не відразу стають кінцевим продуктом. Ці прототипи мають бути ефективними у вирішенні типових завдань, але їх розробка не повинна бути часомісткою, щоб максимізувати процес накопичення та відлагодження знань. Різноманітні інструментальні засоби часто використовуються для прискорення проектування. Успішний прототип може бути розширений або, у разі невдачі, перероблений або відхилений. З часом, прототип, що ефективно вирішує всі завдання, може бути трансформований у кінцевий продукт, часто з перепрограмуванням на мови нижчого рівня для покращення продуктивності та оптимізації використання пам'яті.

Методологія розробки експертних систем включає такі етапи (рис. 1.4).

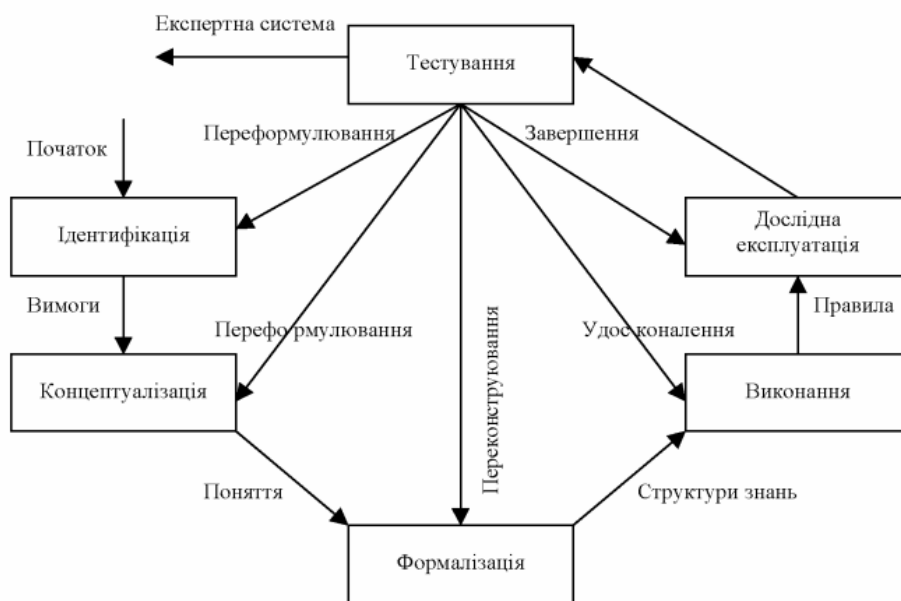


Рисунок 1.4 – Схема етапів розробки експертних систем

Механізм логічного виведення, також відомий як машина логічного виведення, є ключовою складовою експертної системи. Він аналізує інформацію з бази знань, використовуючи стратегії, що тісно пов'язані з методом представлення знань у системі та з особливостями розв'язуваних задач. На основі

цього механізм генерує рекомендації для вирішення задач та містить:

Логічне виведення - це процес використання існуючих знань для виведення нових фактів або висновків. Це основний елемент багатьох систем штучного інтелекту, зокрема експертних систем, де воно використовується для імітації людського процесу міркування.

Типи логічного виведення виділяють наступні:

– Дедукція – це форма логічного виведення, у якому висновки повинні випливати з відповідних ним посилок.

– Індукція – це метод виведення загальних правил, закономірностей або теорій на основі спостережень, даних або конкретних прикладів – один з основних методів машинного навчання, у якому комп'ютери навчаються без утручання людини. До цих методів відноситься кластер-аналіз і нейронні та нейро-нечіткі мережі, що самоорганізуються.

– Інтуїція - це процес сприйняття або розуміння ідей, інформації або рішень без свідомого використання логічного аналізу. Це швидке внутрішнє усвідомлення або відчуття, яке часто виникає без явних аргументів або раціонального обґрунтування.. Логічне виведення такого типу ще не реалізовано в експертних системах.

– Евристика – емпіричні (отримані на основі досвіду) правила виведення.

– Метод породження і перевірки – це підхід, який використовується для формування та перевірки гіпотез або теорій, тобто метод проб і помилок. Часто використовується в сполученні з плануванням для досягнення максимальної ефективності.

– Абдукція – це форма логічного виведення, що формує судження у зворотному напрямку, від істинного висновку до посилок, що могли привести до одержання цього висновку. Абдуктивне міркування дозволяє системам формувати гіпотези або висновки на основі неповних або обмежених даних.

– Судження, застосовувані за замовчуванням – є формою логічного виведення, яка дозволяє робити припущення в умовах відсутності повної інформації та приймати за замовчуванням загальноприйняті чи загальновідомі

знання.

– Аутоепістемічні судження – це судження, пов'язані з власним знанням суб'єкта. Вони відносяться до роздумів індивіда про його власні знання або переконання, включаючи їхню достовірність, обмеження, а також про власну невпевненість або невідомість..

Немонотонні судження - це логічні виведення або міркування, в яких введення нової інформації може змінювати або спростовувати вже існуючі висновки. У немонотонній логіці, на відміну від монотонної, додавання додаткових даних або фактів може призвести до того, що раніше зроблені висновки вже не будуть вважатися вірними або достовірними.

– Судження за аналогією – це метод міркування, при якому висновки про один об'єкт, явище або ситуацію робляться на основі їхньої подібності з іншими об'єктами, явищами або ситуаціями. У даній роботі буде використано абдукційний метод логічного виведення, тож розглянемо його детальніше.

Абдукція є необґрунтованим правилом виведення $((A \rightarrow B) \wedge B) \rightarrow A$ і означає, що висновок не є обов'язково істинним для кожної інтерпретації, при якій істинні передумови. Абдуктивні судження часто називають найкращим поясненням даних B . Традиційна логіка має обмеження на використання в умовах неповної і невизначеної інформації і є монотонною.

Монотонна логіка ґрунтується на наборі аксіом, які вважаються істинними, і з яких виводяться висновки. В цій системі, включення нових даних може лише розширити множину істинних тверджень, але не змінює вже існуючі висновки. Це створює складнощі при моделюванні суджень, які базуються на змінних припущеннях та довірі.

Немонотонна логіка вирішує дозволяє моделювати судження, засновані на довірі та припущеннях, оскільки вона допускає зміну висновків у відповідь на нову інформацію. Вона не обмежується жорсткими аксіомами, як у монотонній логіці, тому рівень довіри та висновки можуть адаптуватися з накопиченням даних.

У системах немонотонних суджень основною задачею є управління

невизначеністю, формування оптимальних припущень на основі наявної інформації. При цьому висновки, зроблені з цих припущень, можуть бути переглянуті або змінені, якщо з'являється нова інформація, що впливає на ступінь довіри до цих припущень.

Методи абдуктивного виведення охоплюють такі підходи, як Стенфордська теорія коефіцієнта впевненості, нечітка логіка, теорія Демпстера-Шафера, Байєсівський метод і метод Нейлора. Стенфордська теорія коефіцієнта впевненості базується на комбінації неформальних оцінок фактів та висновків. Коефіцієнт впевненості представляє собою числову оцінку, яка відображає імовірність або рівень достовірності, з якою факт або правило може бути вважане достовірним чи істинним. При визначенні коефіцієнтів впевненості використовуються як методи математичної статистики, так і суб'єктивні оцінки експертів. Ці коефіцієнти знаходять застосування у продукційних моделях.

Коефіцієнт упевненості антецедента $CF(X, Y)$, що містить посилки X та Y , обчислюється за формулою:

$$CF(X, Y) = \begin{cases} 1, CF(X) = 1 \text{ або } CF(Y) = 1; \\ CF(X) + CF(Y) - CF(X)CF(Y), CF(X) > 0, CF(Y) > 0; \\ \frac{CF(X) + CF(Y)}{1 - \min\{|CF(X)|, |CF(Y)|\}}, CF(X)CF(Y) \leq 0, CF(X) \neq \pm 1, CF(Y) \neq \pm 1; \\ CF(X) + CF(Y) + CF(X)CF(Y), CF(X) < 0, CF(Y) < 0; \\ -1, CF(X) = -1 \text{ або } CF(H) = -1, \end{cases} \quad (1.1)$$

де $CF(X)$ – коефіцієнт упевненості посилки X , $CF(Y)$ – коефіцієнт упевненості посилки Y . Коефіцієнт упевненості консеквента $CF(C)$, визначається на основі коефіцієнта впевненості антецедента $CF(X, Y)$ і коефіцієнта впевненості правила $CF(R)$ як $CF(C) = CF(X, Y) CF(R)$. Коефіцієнт впевненості приймає значення в діапазоні $[-1, +1]$. Якщо він дорівнює $+1$, то це означає, що при дотриманні всіх обговорених умов укладач правила абсолютно упевнений у правильності висновку, а якщо він дорівнює -1 , то виходить, що при дотриманні всіх обговорених умов існує абсолютна впевненість у помилковості

цього висновку. Відмінні від +1 позитивні значення коефіцієнта вказують на ступінь впевненості в правильності висновку, а негативні значення – на ступінь впевненості в його помилковості. Коефіцієнт упевненості є комбінованою оцінкою. Його основне призначення полягає в тому, щоб керувати ходом виконання програми при формуванні суджень, керувати процесом пошуку мети в просторі станів (якщо коефіцієнт упевненості гіпотези виявляється в діапазоні $[-0,2, +0,2]$, то пошук блокується), ранжувати набір гіпотез після обробки всіх ознак. Якщо обидві гіпотези підтверджують висновок або, навпаки, обидві гіпотези його спростовують, то коефіцієнт упевненості їхньої комбінації зростає за абсолютною величиною. У випадку множинності гіпотез, можна послідовно застосовувати цю формулу до кожної з них. Завдяки комутативності цієї формули, порядок, в якому гіпотези обробляються, не впливає на результат. Незважаючи на відсутність суворого теоретичного підґрунтя, коефіцієнти впевненості широко використовуються в продукційних експертних системах. Це пояснюється їх простотою у сприйнятті та інтерпретації, а також хорошою узгодженістю з реальними умовами.

Для підбору апаратного забезпечення комп'ютерних систем експертні системи можна використати таким чином:

- Вхідні дані. Нечіткі дані від користувача, такі як: "потрібний швидкий процесор", "обмежений бюджет", "призначення для геймінгу".
- Аналіз. Експертна система аналізує вхідні дані з використанням своєї бази знань та правил. Вона може використовувати методи нечіткої логіки для опрацювання нечітких вхідних даних.
- Висновок. На виході система надає рекомендації щодо потрібного апаратного забезпечення з коефіцієнтом впевненості. Наприклад, система може рекомендувати конкретну модель процесора з коефіцієнтом впевненості 55%.

1.4 Висновок до розділу 1

У даному розділі було проведено детальний аналіз сучасних

інформаційних технологій, які використовуються в системах рекомендацій для підбору апаратного забезпечення комп'ютерних систем.

Наголошено на особливостях вибору апаратного забезпечення, враховуючи динаміку технологічних новинок, широкий спектр характеристик та високий рівень технічних вимог до сучасних програмних систем. Зазначено, що для прийняття ефективних рішень користувачами в умовах такої нестабільності потрібна високоякісна і точна система рекомендацій.

У процесі огляду існуючих рекомендаційних систем для підбору апаратного забезпечення було з'ясовано, що багато з них мають недоліки, такі як недостатня точність прогнозування, складність використання або відсутність потрібного функціоналу. Також було розглянуто експертні системи, методи логічного виведення. На основі цього аналізу було сформульовано задачу дослідження – розробити інформаційну технологію, яка б дозволила розширити функціонал надання рекомендацій шляхом створення системи відповідності апаратного забезпечення заданим користувачем критеріям.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ДЛЯ ПІДБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

2.1 Розробка методу формування рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Підбір апаратного забезпечення для комп'ютерних систем є істотним аспектом, що впливає на продуктивність та ефективність використання технологій. Метод формування рекомендацій щодо вибору апаратного забезпечення передбачає аналіз ряду параметрів і факторів, які впливають на роботу системи та вимог користувача:

1. Визначення потреб користувача. Спочатку, необхідно ясно визначити і зафіксувати потреби та вимоги користувача, що може включати збір інформації про призначення системи, обсяг та характер оброблюваних даних, а також бажану швидкість і продуктивність. Специфікації можуть залежати від того, чи система буде використовуватися для графічного дизайну, обробки даних, геймінгу чи інших цілей.

2. Технічний аналіз. Важливим етапом є технічний аналіз доступних на ринку комплектуючих, їхніх технічних характеристик, цінових категорій, та відгуків користувачів. Аналіз включає в себе дослідження процесорів, відеокарт, ОЗУ, накопичувачів, блоків живлення, та інших елементів апаратного забезпечення.

3. Розробка критеріїв вибору. Встановлення критеріїв вибору, що можуть включати продуктивність, надійність, сумісність компонентів, енергоефективність, та інші параметри. Ці критерії допомагають структурувати процес вибору та робити його більш об'єктивним та консистентним.

4. Алгоритм вибору апаратного забезпечення. Розробка алгоритму, який на основі визначених критеріїв та технічного аналізу, допоможе автоматизувати процес вибору апаратного забезпечення, пропонуючи

користувачеві оптимальні варіанти в рамках його бюджету та технічних вимог.

5. Валідація та адаптація методу. Після розробки методу, його необхідно валідувати на реальних задачах та отримати зворотний зв'язок від користувачів для подальшого вдосконалення і оптимізації алгоритму.

6. Навчання та підтримка користувачів. Проведення навчань і воркшопів для користувачів з метою забезпечення ефективного використання методу, а також надання технічної підтримки під час його впровадження та використання.

7. Оцінка ефективності методу. Регулярний аналіз ефективності методу, включаючи його спроможність задовольняти потреби користувачів, а також його адаптивність до змін на ринку комп'ютерних комплектуючих.

Врахування математичних підходів може значно підвищити об'єктивність та точність методу підбору апаратного забезпечення. Ось деякі приклади, як можна втілити математичний підхід у вищезазначений процес:

– Розробка критеріїв вибору

Змоделюємо критерії вибору за допомогою вагових коефіцієнтів, де W_i представляє вагу i -го критерію, і сума всіх ваг дорівнює 1:

$$\sum_{i=1}^n W_i = 1. \quad (2.1)$$

– Алгоритм вибору апаратного забезпечення

Можна використовувати нормалізовану оцінку ефективності E кожного варіанту апаратного забезпечення j з використанням вагових коефіцієнтів W_i і показників P_{ij} (таких як швидкість процесора, обсяг пам'яті тощо):

$$E_j = \sum_{i=1}^n W_i \times P_{ij}, \quad (2.2)$$

де n - кількість критеріїв.

Така формула дозволяє ранжувати всі доступні варіанти апаратного

забезпечення за їхньою ефективністю, враховуючи визначені користувачем пріоритети (ваги критеріїв).

- Оцінка ефективності методу.

Ми можемо оцінити ефективність вибраного апаратного забезпечення, використовуючи середньоквадратичне відхилення від оптимального варіанту (якщо відомо):

$$R = \sqrt{\frac{1}{m} \sum_{j=1}^m (E_{j,actual} - E_{j,predicted})^2}, \quad (2.3)$$

де m - кількість варіантів апаратного забезпечення, які були розглянуті при валідації, $E_{j,actual}$ - реальна ефективність j -го варіанту апаратного забезпечення, $E_{j,predicted}$ - прогнозована ефективність j -го варіанту апаратного забезпечення, отримана за допомогою нашого алгоритму.

Коли мова йде про нечіткі параметри, такі як "потужний комп'ютер" або "процесор середньої вартості", ми можемо використовувати нечітку логіку і системи нечітких виводів для побудови моделі, яка може перетворити ці нечіткі вхідні параметри у конкретні та обґрунтовані рекомендації з певним коефіцієнтом впевненості [18]:

- Фазифікація вхідних параметрів - змінні, такі як "потужність" чи "вартість", поділяються на декілька нечітких множин, наприклад: [низька, середня, висока]. Для кожної множини визначаються функції належності, які визначають ступінь належності конкретного значення до кожної з множин.

- Правила нечіткої логіки – визначаються правила для мапінгу вхідних параметрів на вихідні рекомендації, наприклад:

- Якщо "Потужність" висока і "Вартість" середня, тоді "Рекомендований процесор" є висококласним.

Правилам також можна призначити ваги для регулювання їх значимості в рішенні.

- Виведення – використанням системи нечітких виводів і заданих

правил, на основі вхідних даних генеруються рекомендації з коефіцієнтом впевненості.

– Дефазифікація – рекомендації і коефіцієнти впевненості конвертуються з нечітких множин у конкретні варіанти вибору та їх асоційовані числові значення впевненості.

Допустимо, у нас є вхідний параметр "Потужний комп'ютер", який по суті є нечіткою величиною. Після фазифікації та визначення правил, ми можемо отримати рекомендацію у вигляді:

1. Процесор: Intel Core i9, Коефіцієнт впевненості: 0,8.
2. Відеокарта: Nvidia RTX 3080, Коефіцієнт впевненості: 0,7.
3. Оперативна пам'ять: 32GB RAM, Коефіцієнт впевненості: 0,85.

Тут коефіцієнт впевненості вказує на те, наскільки система впевнена в рекомендованому варіанті на основі вхідних параметрів і встановлених правил. Це може допомогти користувачеві оцінити достовірність рекомендації і в разі потреби провести додатковий аналіз чи адаптацію вибору.

Такий підхід може бути ефективно застосований для надання рекомендацій у ситуаціях, коли на вхід подаються нечіткі значення.

Ці математичні моделі можуть бути адаптовані та оптимізовані згідно з конкретними вимогами і умовами в певних контекстах використання, щоб надати користувачам найбільш обґрунтовані та ефективні рекомендації щодо вибору апаратного забезпечення.

2.2 Розробка експертної системи надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Розглянемо необхідні складові для створення експертної системи:

– База знань, яка складається з набору продукційних правил, фактів, та дозволених значень, що відповідають певним умовам. Ця база знань забезпечує основу для процесу прийняття рішень.

- Механізм логічного виведення, що використовує інформацію з бази знань

для висновків або визначення дій, що мають бути виконані.

- Інтерфейс користувача, що забезпечує доступ користувачів до системи та взаємодії з нею. Інтерфейс користувача включає введення та корекцію вхідних даних, а також можливість коригувати базу знань.

- Підсистема пояснень, яка допомагає користувачам розуміти, на основі яких правил та фактів було прийнято конкретне рішення.

- Підсистема поповнення знань, яка дозволяє оновлювати та розширювати базу знань з врахуванням нових інформацій або випадків.

- Синтаксичний аналізатор для перевірки правильності та послідовності вхідних даних.

- Підсистема введення/коректування даних для обробки та підготовки вхідних даних для аналізу.

Розглянемо приклад експертної бази знань для вибору процесора на таблиці 2.1.

Таблиця 2.1 – Приклад експертної бази знань для вибору процесора

№ з/п	Кількість ядер	Тактова частота	Вендор	Рекомендований процесор
Правило 1	4	2-3 ГГц	Intel	Intel Core i5-6400
Правило 2	4	2-3 ГГц	AMD	AMD a10-9700e
Правило 3	4	2-3 ГГц	Intel	Intel Core i3-12100f
Правило 4	4	2-3 ГГц	AMD	AMD ryzen 3-1200
Правило 5	8	3.1-4 ГГц	Intel	Intel Core i7-11700
Правило 6	8	3.1-4 ГГц	AMD	AMD Ryzen 7-1700
Правило 7	8	3.1-4 ГГц	Intel	Intel Core i9-11900kf
Правило 8	8	3.1-4 ГГц	AMD	ADM Ryzen 7-5700X

У таблиці 2.1 розміщені наступні особливості, а саме: елементи правила <умова> і <висновок> можуть бути складеними і з'єднаними оператором "І", що є кон'юнкцією; визначено "ІІ", а не "І", тому що з букви "І" можуть

починатися інші елементи <умови> і <висновку>. У правилі застосовується коефіцієнт достовірності (КД). У процесі логічного виведення КД змінюється (згладжується, усереднюється) і наприкінці видається його оновлене значення для поставленої мети. Якщо КД відсутній, то вважається, що він дорівнює 100%.

Налаштування нечіткої моделі - це процес, що передбачає послідовну зміну параметрів моделі з метою зменшення розбіжності між логічними висновками, зробленими моделлю, та експериментальними даними. Цей процес є ітераційним, де на кожній ітерації параметри моделі змінюються залежно від рівня відхилення, зокрема використовуються методи мінімізації для досягнення найбільш точних результатів.

Вибір апаратного забезпечення комп'ютерних систем є процесом, який вимагає уважного аналізу різних характеристик та потреб користувача. Вибір не зводиться до простого порівняння технічних характеристик, а вимагає глибокого розуміння того, як окремі компоненти взаємодіють між собою та як вони впливають на загальну продуктивність системи.

Тактова частота процесора, наприклад, відіграє важливу роль у визначенні продуктивності системи. Висока тактова частота говорить про здатність процесора швидко обробляти інформацію, що є критично важливим для виконання ресурсоемних завдань, таких як 3D-моделювання, відео редагування або геймінг. Проте, нижча тактова частота може вказувати на енергоефективність процесора, що є бажаним для портативних або енергозберігаючих систем.

Об'єм оперативної пам'яті також є важливим аспектом. Великий об'єм оперативної пам'яті дозволяє системі одночасно виконувати більше задач, що може збільшити продуктивність використання комп'ютера. З іншого боку, система з меншим об'ємом оперативної пам'яті може бути більш економічною і енергоефективною.

Що стосується накопичувача, то його об'єм впливає на кількість даних, яку можна зберегти на комп'ютері. Великі накопичувачі зазвичай потрібні для зберігання великої кількості файлів або для роботи з великими даними. Проте,

менший накопичувач може бути більш придатний для тих, хто в основному використовує хмарні сервіси. Швидкість накопичувача визначає, наскільки швидко дані можуть бути прочитані або записані на пристрій. Це важливий аспект, який впливає на загальну продуктивність системи.

Вибір відеокарти залежить від того, як користувач планує використовувати свій комп'ютер. Потужні відеокарти, які можуть обробляти високоякісну графіку та відео, необхідні для геймерів та професіоналів з графічного дизайну. З іншого боку, якщо комп'ютер використовується в основному для офісних робіт або для перегляду веб-сторінок, потужна відеокарта може бути не потрібна.

Описану задачу вибору апаратного забезпечення комп'ютерних систем можна вирішити за допомогою методів аналізу образів, Бассівських імовірнісних методів, логічного програмування, планування експериментів та нечіткої логіки. Кожен з даних методів має свої переваги та недоліки для даної предметної області, але загальним недоліком перших чотирьох перелічених методів є створення великих баз даних, що містять велику кількість параметрів.

Підхід, заснований на використанні нечіткої логіки, дозволяє вирішити дану проблему за рахунок скорочення обсягу бази даних при використанні лінгвістичних змінних. Великою проблемою для раніше зазначених методів є також нечіткість у процесі вибору комплектуючих через невизначеність характеристик для певних комплектуючих. Це призводить до неточностей у процесі вибору. Для усунення цих неточностей можливим є залучення певної кількості експертів, що потребує додаткових розрахунків при визначенні значущості їхніх оцінок. Набагато простішим та при цьому надійним методом буде застосування нечіткої логіки з використанням системи нечіткого входу

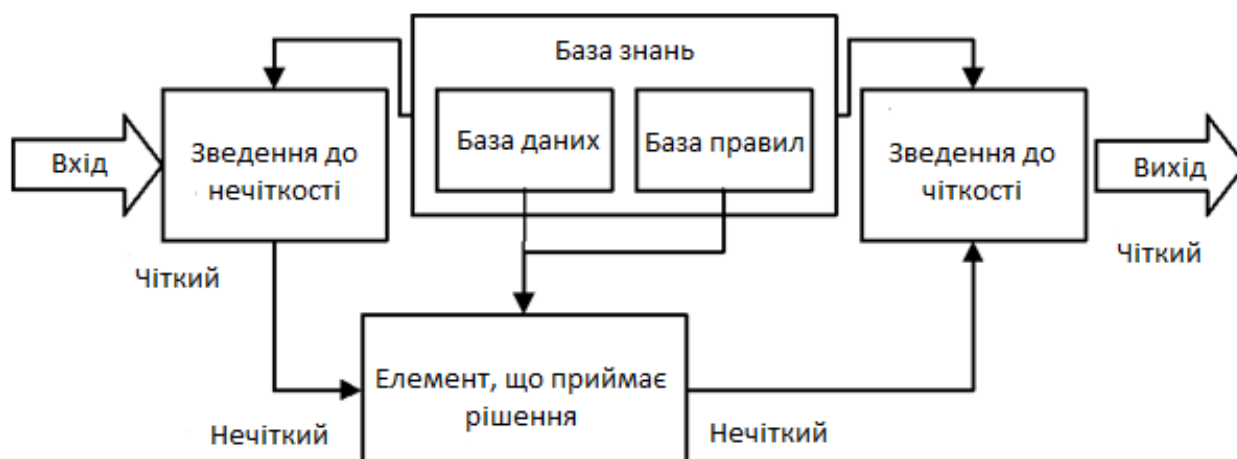


Рисунок 2.1 – Принципова схема функціонування системи нечіткого висновку

Отже, згідно із теоретичними даними про можливі параметри комп'ютерних комплектуючих, можливим є визначення лінгвістичних змінних, якими оперує розроблюваний метод на основі нечіткої логіки.

При великій кількості параметрів стану доцільно використовувати ієрархічно пов'язані бази знань, в яких вихідна змінна однієї бази знань є вхідною для бази знань вищого рівня ієрархії. Моделювання на основі нечітких баз знань здійснюється через нечіткий логічний висновок. Загальна архітектура системи нечіткого висновку показана на рисунку 2.2.



Рисунок 2.2 – Типова архітектура системи нечіткого висновку

Блоки системи на рисунку 2.2 виконують такі функції: – «Фазифікатор»

перетворює вектор (X) значень параметрів стану в вектор ($X\%$) ступенів належності параметрів стану до нечітких термів; – «Функції належності» складають бібліотеку аналітичних залежностей та параметрів функцій належностей, якими формалізуються лінгвістичні терми з нечітких правил; – «Нечітка база знань» це сукупність експертних лінгвістичних правил типу , які формалізовані засобами теорії нечітких множин; – «Машина нечіткого висновку» визначає значення вихідної змінної у вигляді нечіткої множини ($Y\%$); – «Дефазифікатор» перетворює нечіткий розв’язок ($Y\%$) в чіткий (Y).

Лінгвістична змінна – це множина нечітких змінних, використовувана для надання словесного опису деякому нечіткому числу, отриманому в результаті деяких операцій, і визначається кортежем:

$$\langle \beta, T(\beta), X, G, M \rangle, \quad (2.4)$$

де β – назва лінгвістичної змінної; $T(\beta)$ – базова терм-множина значень β , тобто множина її лінгвістичних (вербальних) значень; $X = \{x\}$ – множина визначення нечіткої змінної (універсальна множина); G – синтаксична процедура, яка описує процес створення з множини $T(\beta)$ нових змістовних для даної задачі значень $\alpha \in T^*$ лінгвістичної змінної. При цьому множину $T^* = T \cup G(T)$ називають розширеною терм-множиною лінгвістичної змінної; M – семантична процедура, яка дозволяє приписати кожному новому значенню лінгвістичної змінної, створеному процедурою G , відповідну семантику шляхом формування відповідної нечіткої множини $\tilde{C}(\alpha)$, тобто відображає нове значення (терм) у нову відповідну нечітку змінну [19].

Визначимо лінгвістичні змінні системи та визначимо область їхніх значень:

- 1) Процесор [2; 24] ядра.
- 2) Оперативна пам'ять [1; 64] GB;
- 3) Об'єм жорсткого диска [128GB; 18TB];

4) Відеокарта [1; 24] GB;

Визначимо терми даних лінгвістичних змінних та побудуємо функції належності:

1. Фактор y_1 – процесор. $T(y_1) = \langle \text{дуже висока; висока; середня; низька; дуже низька} \rangle$. Даний фактор визначено на універсальній множині $U(y_1) = [2; 24]$ ядра. Ці терми формалізуються у вигляді наступних нечітких множин:

- дуже низька = $\{ \langle 1/2 \rangle; \langle 1/3 \rangle; \langle 0.5/4 \rangle; \langle 0.115/6 \rangle \}$;
- низька = $\{ \langle 0.120/3 \rangle; \langle 0.515/4 \rangle; \langle 1/6 \rangle; \langle 0.5/8 \rangle \langle 0.145/10 \rangle \}$;
- середня = $\{ \langle 0.170/6 \rangle; \langle 0.530/8 \rangle; \langle 1/10 \rangle; \langle 1/12 \rangle; \langle 0.5/14 \rangle; \langle 0.215/15 \rangle \}$;
- висока = $\{ \langle 0.185/12 \rangle; \langle 0.5/14 \rangle; \langle 1/15 \rangle; \langle 1/16 \rangle; \langle 0.5/18 \rangle; \langle 0.241/20 \rangle \}$;
- дуже висока = $\{ \langle 0.311/16 \rangle; \langle 0.5/18 \rangle; \langle 1/20 \rangle; \langle 1/24 \rangle \}$.



Рисунок 2.3 – Графічне представлення функції належності лінгвістичної змінної «Процесор»

Визначимо функцію належності для категорії «Дуже низька»:

$$f_{\partial n} = \left\{ \begin{array}{l} 1, \text{ для } x \leq b \\ \frac{d-x}{d-c}, \text{ для } b < x < d \\ 0, \text{ для } x \geq d \end{array} \right\}. \quad (2.5)$$

Підставивши значення отримаємо:

$$f_{\partial n} = \left\{ \begin{array}{l} 1, \text{ для } x \leq 2 \\ \frac{4-x}{4-3}, \text{ для } 2 < x < 4 \\ 0, \text{ для } x \geq 4 \end{array} \right\}. \quad (2.6)$$

Визначимо функцію належності для категорії «Низька» використовуючи формулу приналежності трикутної функції:

$$f_n = \left\{ \begin{array}{l} 0, \text{ для } x \leq 3 \\ \frac{x-3}{4-3}, \text{ для } 3 < x < 4 \\ \frac{5-x}{5-4}, \text{ для } 4 \leq x < 5 \\ 0, \text{ для } x \geq 5 \end{array} \right\}. \quad (2.7)$$

Визначимо функцію належності для категорії «Середня» використовуючи формулу приналежності трапецеїдальної функції:

$$f_c = \left\{ \begin{array}{l} 0, \text{ для } x \leq 5; \\ \frac{x-5}{6-5}, \text{ для } 5 < x < 6; \\ \frac{5-x}{5-4}, \text{ для } 6 \leq x < 7; \\ 0, \text{ для } x \geq 7. \end{array} \right\}. \quad (2.8)$$

Визначимо функцію належності для категорії «Висока» використовуючи формулу приналежності трапецеїдальної функції:

$$f_v = \left\{ \begin{array}{l} 0, \text{ для } x \leq 7; \\ \frac{x-7}{8-7}, \text{ для } 7 < x < 8; \\ \frac{9-x}{9-8}, \text{ для } 8 \leq x < 9; \\ 0, \text{ для } x \geq 9. \end{array} \right\}. \quad (2.9)$$

Визначимо функцію належності для категорії «Дуже висока»:

$$f_{dv} = \left\{ \begin{array}{l} 0, \text{ для } x \leq 9; \\ \frac{x-9}{10-9}, \text{ для } 9 < x < 10; \\ 0, \text{ для } x \geq 10. \end{array} \right\}. \quad (2.10)$$

2. Фактор y_2 – оперативна пам'ять. $T(y_2) = \langle \text{дуже висока; висока; середня; низька; дуже низька} \rangle$. Даний фактор визначено на універсальній множині $U(y_2) = [2; 64]$ GB. Ці терми формалізуються у вигляді наступних нечітких множин:

- дуже низька = $\{ \langle 1/1 \rangle; \langle 1/2 \rangle; \langle 0,5/4 \rangle; \langle 0.112/4 \rangle \}$;
- низька = $\{ \langle 0.185/2 \rangle; \langle 0,5/4 \rangle; \langle 1/6 \rangle; \langle 1/8 \rangle; \langle 0,5/10 \rangle; \langle 0.211/12 \rangle \}$;
- середня = $\{ \langle 0.134/8 \rangle; \langle 0.5/10 \rangle; \langle 1/12 \rangle; \langle 0.5/16 \rangle; \langle 0.224/20 \rangle \}$;
- висока = $\{ \langle 0.107/12 \rangle; \langle 0.5/16 \rangle; \langle 1/20 \rangle; \langle 1/24 \rangle; \langle 0.5/36 \rangle; \langle 0.124/48 \rangle \}$;
- дуже висока = $\{ \langle 0.215/24 \rangle; \langle 0,5/36 \rangle; \langle 1/48 \rangle; \langle 1/64 \rangle \}$.



Рисунок 2.4 – Графічне представлення функції належності лінгвістичної змінної «Оперативна пам'ять»

3. Фактор y_3 – об'єм жорсткого диска. $T(y_3) = \langle \text{дуже високий; високий; середній; низький; дуже низький} \rangle$. Даний фактор визначено на універсальній множині $U(y_3) = [128\text{GB}; 18\text{TB}]$. Ці терми формалізуються у вигляді наступних нечітких множин:

- дуже низький = $\{ \langle 1/128\text{GB} \rangle; \langle 1/256\text{GB} \rangle; \langle 0,5/512\text{GB} \rangle; \langle 0,275/1\text{TB} \rangle \}$;
- низький = $\{ \langle 0,126/256\text{GB} \rangle; \langle 0,5/512\text{GB} \rangle; \langle 1/1\text{TB} \rangle; \langle 0,5/2\text{TB} \rangle; \langle 0,223/4\text{TB} \rangle \}$;
- середній = $\{ \langle 0,163/1\text{TB} \rangle; \langle 0,5/2\text{TB} \rangle; \langle 1/4\text{TB} \rangle; \langle 0,5/6\text{TB} \rangle; \langle 0,147/8\text{TB} \rangle \}$;
- високий = $\{ \langle 0,246/4\text{TB} \rangle; \langle 0,5/6\text{TB} \rangle; \langle 1/8\text{TB} \rangle; \langle 1/10\text{TB} \rangle; \langle 0,5/12\text{TB} \rangle; \langle 0,214/16\text{TB} \rangle \}$;
- дуже високий = $\{ \langle 0,335/10\text{TB} \rangle; \langle 0,5/12\text{TB} \rangle; \langle 1/16\text{TB} \rangle; \langle 1/18\text{TB} \rangle \}$.

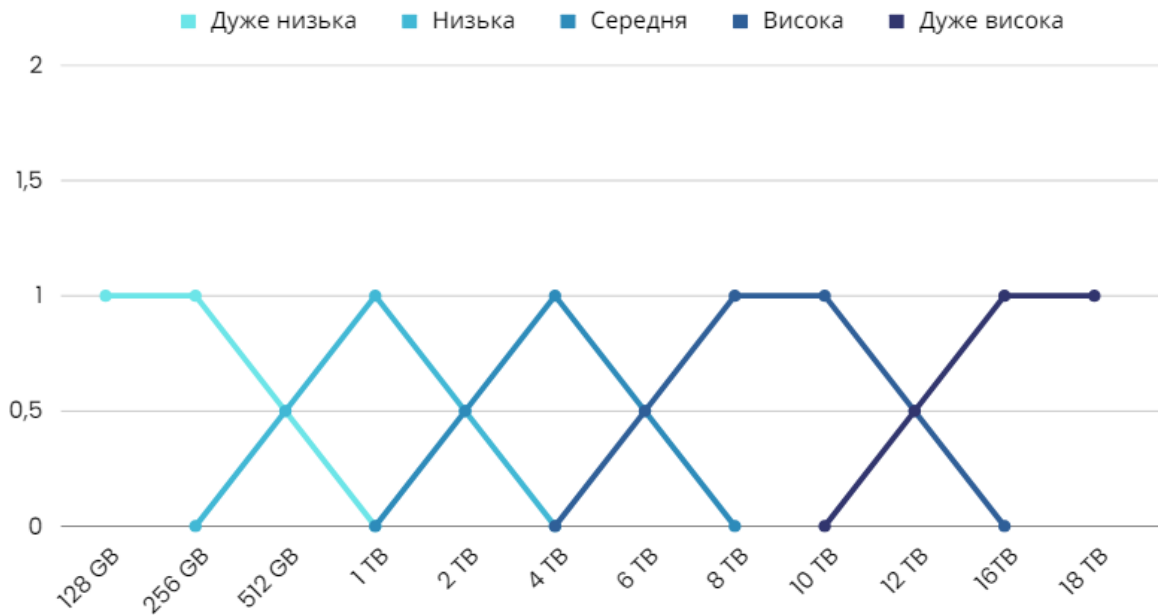


Рисунок 2.5 – Графічне представлення функції належності лінгвістичної змінної «Об'єм жорсткого диска»

4. Фактор y_4 – відеокарта. $T(y_4) = \langle \text{дуже висока; висока; середня; низька; дуже низька} \rangle$. Даний фактор визначено на універсальній множині $U(y_4) = [1; 16]$ GB. Ці терми формалізуються у вигляді наступних нечітких множин:

- дуже низька = $\{ \langle 1/1 \rangle; \langle 0,535/2 \rangle; \langle 0,314/4 \rangle \}$;
- низька = $\{ \langle 0,214/1 \rangle; \langle 0,512/2 \rangle; \langle 1/4 \rangle; \langle 0,535/6 \rangle; \langle 0,144/8 \rangle \}$;
- середня = $\{ \langle 0,232/4 \rangle; \langle 0,5/6 \rangle; \langle 1/8 \rangle; \langle 0,5/10 \rangle; \langle 0,187/12 \rangle \}$;
- висока = $\{ \langle 0,163/8 \rangle; \langle 0,5/10 \rangle; \langle 1/12 \rangle; \langle 0,5/16 \rangle; \langle 0,201/24 \rangle \}$;
- дуже висока = $\{ \langle 0,221/12 \rangle; \langle 0,5/16 \rangle; \langle 1/24 \rangle \}$.

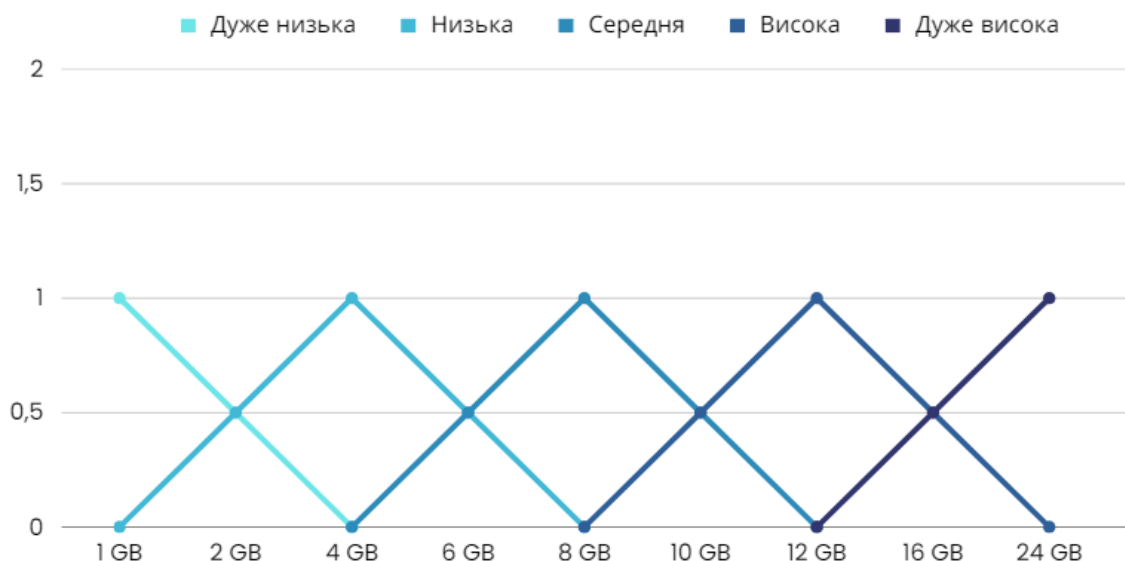


Рисунок 2.6 – Графічне представлення функції належності лінгвістичної змінної «Відеокарта»

Оскільки предметна область магістерського проектування не є добре структурованою, використаємо для подання знань у системі продукційні правила. При цьому нечітка база знань про вплив факторів $X = \{x_1, x_2, \dots, x_n\}$ на значення параметру у називається сукупність логічних висловлювань.

$$\begin{aligned}
 &\text{Якщо } x_1 = a_1^{j1} \text{ та } x_2 = a_2^{j1} \text{ та } x_n = a_n^{j1}, && \text{або} \\
 &x_1 = a_1^{j2} \text{ та } x_2 = a_2^{j2} \text{ та } x_n = a_n^{j2}, && \text{або} \\
 &x_1 = a_1^{jk} \text{ та } x_2 = a_2^{jk} \text{ та } x_n = a_n^{jk}, && (2.11) \\
 &\text{то } y = b_j \quad j = \overline{1, m},
 \end{aligned}$$

де a_i^{jp} – лінгвістичний терм, який оцінює значення фактора x_i у p -й диз'юнкції j -го логічного висловлювання ($j = \overline{1, m}$, $p = \overline{1, k}$, $i = \overline{1, n}$); k_j – число диз'юнкції (або) в j -у логічному висловлюванні.

Правила, що будуть складати причинно-наслідкову ієрархію мають вигляд:

$$\text{ЯКЩО } A = N, \text{ ТО } B = M, \quad (2.12)$$

де A, B – лінгвістичні змінні; N, M – їх терми.

На підставі сформованого набору правил будуємо матрицю знань. Ця матриця знань визначає систему логічних висловлювань типу «Якщо-то, або», яка зв'язує значення параметрів $x_1 \div x_n$ з одним із комплектуючих для ПК d_j ; $j = \overline{1, m}$. За формулою 2.12 складемо правила для нечіткої бази знань, фрагмент із якої представлено в таблиці 2.2. При цьому користуємось умовними позначеннями:

- Дв - дуже висока;
- В – висока;
- С – середня;
- Н – низька;
- Дн - дуже низька.

Таблиця 2.2 – База знань нечіткого методу для вибору комп'ютерних комплектуючих

№	y_1	y_2	y_3	y_4	Y	Z	D (збірка ПК)
1	2	3	4	5	6	7	8
1	Дв	Дв	Дв	Дв	Дв	z_1	Intel Core i9-12900K, Corsair Vengeance LPX 64 GB, Seagate BarraCuda Pro 4TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280
2	Дв	В	В	Дв	Дв	z_1	Intel Core i9-12900K, G.SKILL Trident Z RGB 32 GB, Western Digital WD Black 2TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280
3	В	Дв	Дв	В	Дв	z_1	AMD Ryzen 9 3950X, Corsair Vengeance LPX 64 GB, Seagate BarraCuda Pro 4TB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
4	Дв	С	С	Дв	В	z_2	Intel Core i9-12900K, G.SKILL Trident Z RGB 32 GB, Western Digital WD Black 2TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280

Продовження таблиці 2.2

5	В	С	С	В	В	z ₂	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
6	В	С	В	В	В	z ₂	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
7	В	С	С	С	С	z ₃	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Samsung Portable SSD T7 500GB
8	В	С	С	Н	С	z ₃	AMD Ryzen 9 3950X, G.SKILL Trident Z RGB 32 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Apacer AS340 Panther 480GB
9	Н	Н	Н	Н	Н	z ₄	Intel Core 2 Duo E6300 L2 SL9TA, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce GTX 1050, Apacer AS340 Panther 480GB
10	Дв	Н	Н	Дн	Н	z ₄	Intel Core i9-12900K, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Apacer AS340 Panther 480GB
11	Н	Н	Н	В	Н	z ₄	Intel Core i3-6100, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
12	Дн	Дн	Н	Дн	Дн	z ₅	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Western Digital WD Blue 500GB, NVIDIA GeForce GT 210, Apacer AS340 Panther 480GB
13	Дн	Дн	Дн	Н	Дн	z ₅	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Toshiba 200GB, NVIDIA GeForce GTX 1050, Apacer, Apacer AS340 Panther 480GB
14	Дн	Дн	Дн	Дн	Дн	z ₅	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Toshiba 200GB, NVIDIA GeForce GT 210, Kesu K1 320Gb
...

Формалізація частинних параметрів стану у вигляді лінгвістичних змінних наведена в таблицях 3.2 і 3.3

Таблиця 2.3 – Формалізація частинних параметрів стану лінгвістичними змінними

Назва показника	Позначення	Вхідні параметри
	Y ₁ -(Процесор)	

Продовження таблиці 2.3

Кількість ядер, ядра	x_1	[2; 24]
	Y_2 -(Оперативна пам'ять)	
Обсяг оперативної пам'яті, Гб	x_2	[1; 64]
	Y_3 -(Жорсткий диск)	
Об'єм жорсткого диска, Гб	x_3	[128; 18000]
	Y_4 -(Відеокарта)	
Об'єм відеопам'яті, Гб	x_4	[1; 16]

По базі знань з таблиці 2.2 записуються нечіткі логічні рівняння (2.13) – (2.17) для визначення комплексного показника середньої продуктивності:

$$\mu^{ДВ}(Y) = \mu^{ДВ} \times \mu^{ДВ}(y_2) \times \mu^{ДВ}(y_3) \times \mu^{ДВ}(y_4) \vee \mu^{ДВ}(y_1) \times \mu^B(y_2) \times \mu^B(y_3) \times \mu^{ДВ}(y_4) \vee \mu^B(y_1) \times \mu^{ДВ}(y_2) \times \mu^{ДВ}(y_3) \times \mu^B(y_4); \quad (2.13)$$

$$\mu^B(Y) = \mu^{ДВ}(y_1) \times \mu^C(y_2) \times \mu^C(y_3) \times \mu^{ДВ}(y_4) \vee \mu^B(y_1) \times \mu^C(y_2) \times \mu^C(y_3) \times \mu^B(y_4) \vee \mu^B(y_1) \times \mu^C(y_2) \times \mu^B(y_3) \times \mu^B(y_4); \quad (2.14)$$

$$\mu^C(Y) = \mu^B(y_1) \times \mu^C(y_2) \times \mu^C(y_3) \times \mu^C(y_4) \vee \mu^B(y_1) \times \mu^C(y_2) \times \mu^C(y_3) \times \mu^H(y_4); \quad (2.15)$$

$$\mu^H(Y) = \mu^H(y_1) \times \mu^H(y_2) \times \mu^H(y_3) \times \mu^H(y_4) \vee \mu^{ДВ}(y_1) \times \mu^H(y_2) \times \mu^H(y_3) \times \mu^{ДН}(y_4) \vee \mu^H(y_1) \times \mu^H(y_2) \times \mu^H(y_3) \times \mu^B(y_4); \quad (2.16)$$

$$\mu^{ДН}(Y) = \mu^{ДН}(y_1) \times \mu^{ДН}(y_2) \times \mu^H(y_3) \times \mu^{ДН}(y_4) \vee \mu^{ДН}(y_1) \times \mu^{ДН}(y_2) \times \mu^{ДН}(y_3) \times \mu^H(y_4) \vee \mu^{ДН}(y_1) \times \mu^{ДН}(y_2) \times \mu^{ДН}(y_3) \times \mu^{ДН}(y_4); \quad (2.17)$$

За формулами функцій належності визначаються ступені належності значень параметрів стану нечітким термам та зводяться в таблицю 2.4.

Таблиця 2.4 – Ступені належності частинних параметрів стану

Параметри стану	Ступені належності параметрів стану нечітким термам
x_1	$\mu^{ДВ}(x_1) = 1,0; \mu^B(x_1) = 0,751; \mu^C(x_1) = 0,578; \mu^H(x_1) = 0,389; \mu^{ДН}(x_1) = 0,146;$
x_2	$\mu^{ДВ}(x_2) = 0,655; \mu^B(x_2) = 0,826; \mu^C(x_2) = 1,0; \mu^H(x_2) = 0,363; \mu^{ДН}(x_2) = 0,237;$
x_3	$\mu^{ДВ}(x_3) = 0,421; \mu^B(x_3) = 0,682; \mu^C(x_3) = 0,753; \mu^H(x_3) = 0,957; \mu^{ДН}(x_3) = 0,637;$
x_4	$\mu^{ДВ}(x_4) = 0,852; \mu^B(x_4) = 1,0; \mu^C(x_4) = 0,758; \mu^H(x_4) = 0,415; \mu^{ДН}(x_4) = 0,215;$

$$\mu^{ДВ}(Y) = 1,0 \times 0,655 \times 0,421 \times 0,852 \vee 1,0 \times 0,826 \times 0,682 \times 0,852 \vee 0,751 \times 0,655 \times 0,421 \times 1,0 = 0,682; \quad (2.18)$$

$$\mu^B(Y) = 1,0 \times 1,0 \times 0,753 \times 0,852 \vee 0,751 \times 1,0 \times 0,753 \times 1,0 \vee 0,751 \times 1,0 \times 0,682 \times 1,0 = 0,753; \quad (2.19)$$

$$\mu^C(Y) = 0,751 \times 1,0 \times 0,753 \times 0,758 \vee 0,751 \times 1,0 \times 0,753 \times 0,415 = 0,751; \quad (2.20)$$

$$\mu^H(Y) = 0,389 \times 0,363 \times 0,957 \times 0,415 \vee 1,0 \times 0,363 \times 0,957 \times 0,215 \vee 0,389 \times 0,363 \times 0,957 \times 1,0 = 0,363; \quad (2.21)$$

$$\mu^{ДН}(Y) = 0,146 \times 0,237 \times 0,957 \times 0,215 \vee 0,146 \times 0,237 \times 0,637 \times 0,415 \vee 0,146 \times 0,237 \times 0,637 \times 0,215 = 0,146. \quad (2.22)$$

Таким чином, в результаті логічного висновку отримано таку нечітку множину:

$\tilde{Z} = \left(\frac{0,682}{z_1}; \frac{0,753}{z_2}; \frac{0,751}{z_3}; \frac{0,363}{z_4}; \frac{0,146}{z_5}; \right)$. Найбільший ступінь (0,753) має розв'язок z_2 .

2.3 Розробка алгоритму надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Перед створенням алгоритму роботи програми спершу розглянемо, які кроки мають виконуватись для збору та обробки інформації, а також надання рекомендацій.

1. Збір даних від користувача:
 - Опитування користувача щодо бюджету, призначення системи (ігри, робота, навчання, мультимедіа), особливих вимог та інших факторів, які впливають на вибір комплектуючих.
2. Аналіз вимог та встановлення пріоритетів компонентів на основі призначення системи:
 - Визначення важливості кожного компонента в залежності від призначення системи. Наприклад, для ігрової системи важливі відеокарта та процесор, а для сервера - надійність та енергоефективність.
3. Визначення параметрів нечіткої логіки:
 - Створення нечітких множин для характеристик комплектуючих, таких як продуктивність, ціна, споживана потужність, надійність тощо.
 - Розробка нечітких правил на основі експертних знань або аналізу даних для моделювання взаємозв'язків між характеристиками комплектуючих та користувацькими вимогами.
4. Підбір комплектуючих з урахуванням критеріїв вибору та пріоритетів:
 - Застосування нечітких правил для оцінки альтернативних комплектуючих на основі їх характеристик.
5. Вибір оптимального варіанту комплектації на основі оцінки альтернатив за критеріями вибору:

- Ранжування комплектуючих на основі їх оцінок, враховуючи пріоритети та бюджет користувача.

- Вибір найкращої комплектації, яка максимізує виконання користувацьких вимог та відповідає бюджету.

6. Представлення рекомендацій щодо вибору комплектуючих для користувача:

- Відображення відібраних комплектуючих, їх характеристик та загальної вартості системи.

- Надання додаткової інформації про кожний компонент, такої як відгуки користувачів, порівняння з альтернативними варіантами та переваги/недоліки.

Алгоритм роботи нечіткої інтелектуальної системи вибору комп'ютерних комплектуючих представлено на рисунку 2.7.

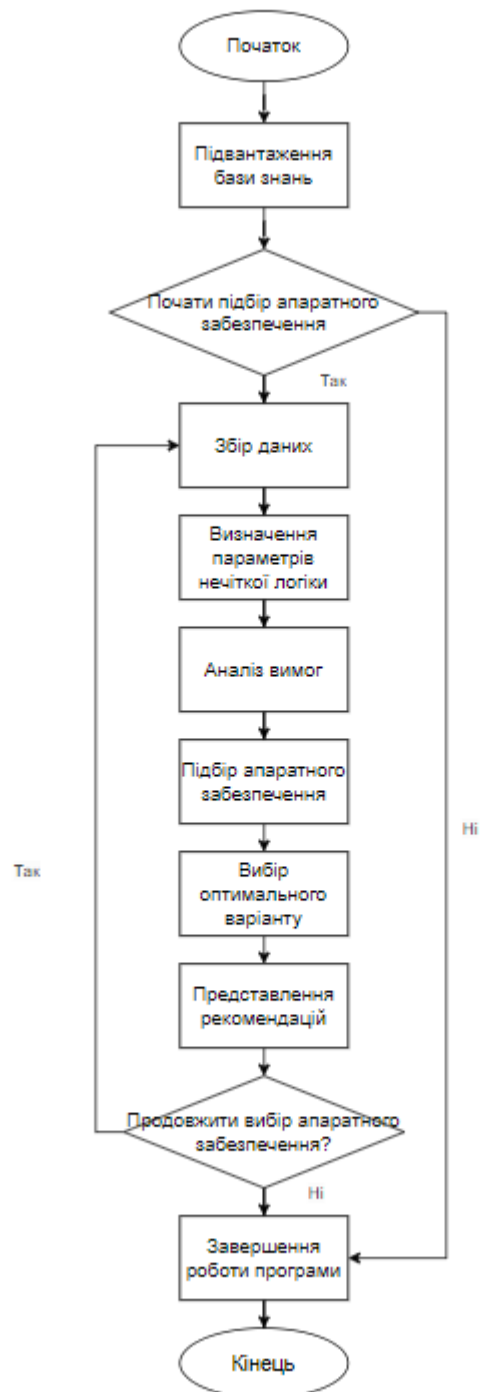


Рисунок 2.7 – Схема алгоритму роботи інформаційної технології надання рекомендацій для підбору апаратного забезпечення

Створивши загальний алгоритм роботи інформаційної системи ми можемо детальніше розглянути алгоритм етапів роботи інформаційної системи.

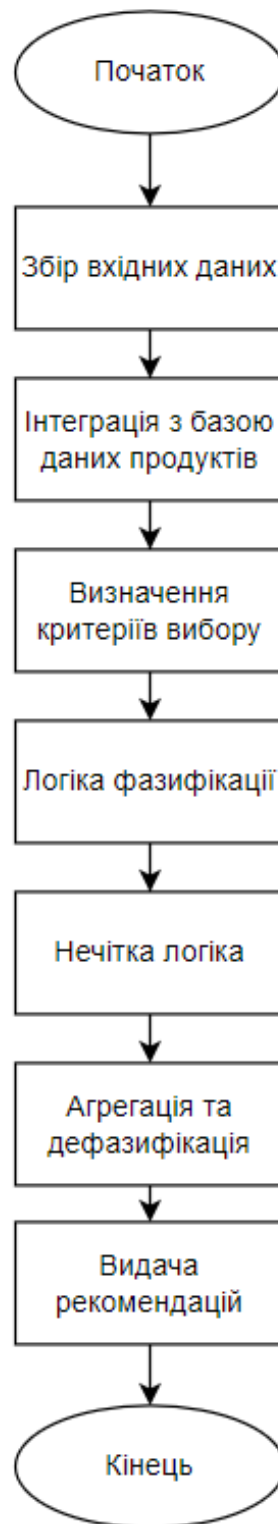


Рисунок 2.8 – Схема алгоритму роботи програмного модуля надання рекомендацій

2.4 Висновок до розділу 2

У розділі було розглянуто проектування інформаційної технології для надання рекомендацій з підбору апаратного забезпечення комп'ютерних систем, акцентуючи увагу на інтегрованому підході до розробки методу формування рекомендацій, експертної системи, та алгоритму надання рекомендацій. Було вироблено стратегічний підхід до аналізу вимог користувачів та динаміки технологічних новинок, втілюючи методологію, яка враховує як експертні знання, так і автоматизовані алгоритми для надання рекомендацій. Розроблений метод оцінювання та валідації апаратного забезпечення здатен адаптуватися до швидкозмінних технічних характеристик та специфікацій. Експертна технологія інтегрує наявний досвід та знання у галузі, надаючи детальні, обґрунтовані рекомендації з урахуванням конкретних вимог користувачів та ринкових трендів. Пропонований алгоритм оптимізує процес вибору апаратного забезпечення, ефективно інтегруючи дані та динаміку змін у технологічних характеристиках та користувацьких вимогах. Загалом, розроблена інформаційна технологія об'єднує ці елементи, маючи на меті надання користувачам координованих, точних та обґрунтованих рекомендацій з підбору апаратного забезпечення, що відповідає їхнім індивідуальним вимогам та актуальним технологічним тенденціям.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ДЛЯ ПІДБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору засобів розробки програмного забезпечення

JavaScript — це мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, що дозволяє створювати інтерактивні сторінки. HTML і CSS є мовами, які надають структуру та стиль веб-сторінкам, JavaScript надає інтерактивні елементи з якими взаємодіє користувач. Підключення JavaScript перетворює звичайну статичну сторінку в інтерактивну.

JavaScript використовується для створення динамічних веб-сайтів, веб- та мобільних додатків, ігор, веб-серверів, внутрішньої інфраструктури тощо [20].

Використання JavaScript має безліч переваг:

- відносно легке вивчення і використоання;
- використання на стороні клієнта і на стороні сервера;
- забезпечення динамічності та інтерактивність веб-сайтів;
- кросплатформеність;
- підтримка всіма основними веб-браузерами;
- велика кількість бібліотек, фреймворків та API для полегшення завдань;
- швидке виконання простих завдань на стороні клієнта;
- добре працює в поєднанні з іншими мовами програмування.

Веб-програмування розвивається дуже швидко. Існує багато бібліотек та фреймворків для JavaScript. Було обрано бібліотеки React та TensorFlow.js для досягнення необхідного функціоналу. React.js - це бібліотека JavaScript з відкритим вихідним кодом, ретельно розроблена компанією Facebook, яка має на меті спростити складний процес створення інтерактивних користувацьких

інтерфейсів [21].

У React ви розробляєте свої додатки, створюючи багаторазові компоненти. Ці компоненти є окремими частинами кінцевого інтерфейсу, які, будучи зібраними, формують весь користувацький інтерфейс додатку.

Основна роль React у додатку полягає в тому, щоб обробляти рівень представлення цього додатку, подібно до V у патерні модель-вид-контролер (MVC), забезпечуючи найкраще та найефективніше виконання рендерингу. Замість того, щоб розглядати весь інтерфейс користувача як єдине ціле, React.js заохочує розробників розділяти ці складні інтерфейси на окремі багаторазові компоненти, які формують будівельні блоки всього інтерфейсу. Таким чином, фреймворк ReactJS поєднує швидкість та ефективність JavaScript з більш ефективним методом маніпулювання DOM для швидшого рендерингу веб-сторінок та створення високодинамічних та адаптивних веб-додатків.

Розглянемо основні переваги використання React [22]:

1. Декларативність. У реакті DOM є декларативним. Ми можемо створювати інтерактивні інтерфейси, змінюючи стан компонента, і React піклується про оновлення DOM відповідно до цього. Це означає, що ми ніколи не взаємодіємо з DOM. Отже, це полегшує розробку інтерфейсу користувача та його налагодження. Ми можемо просто змінити стан програми і подивитися, як буде виглядати інтерфейс в цей момент. Це робить наш код більш передбачуваним і легшим для налагодження.

2. Пиши один раз, використовуй будь-де: Ми можемо розробляти нові функції в React без переписування існуючого коду. Він також може рендерити на сервері за допомогою Node і працювати в мобільних додатках за допомогою React Native. Таким чином, ми можемо створювати IOS, Android та веб-додатки одночасно. На завершення, React підтримує широке повторне використання коду.

3. SEO-дружній: SEO полягає в тому, щоб полегшити розробникам пошук потрібного контенту для користувача. Коли користувач здійснює пошук, пошукові системи, такі як Google, Yahoo, Bing або Baidu, намагаються знайти

сторінку, яка найбільше відповідає конкретному запиту. React впливає на SEO, надаючи вам SPA (односторінковий додаток), який вимагає Javascript для відображення вмісту сторінки, який потім може бути відрендерений та проіндексований.

4. Швидкий, ефективний і простий у вивченні: Він містить заздалегідь створені шаблони і функції, які можна вибирати і комбінувати, як будівельні блоки, щоб створювати швидкі, привабливі і масштабовані проекти за менший час у порівнянні з розробкою всього додатка по черзі.

5. Гарантує стабільний код: ReactJS використовує виключно низхідний потік даних, щоб гарантувати, що навіть незначні зміни в дочірніх структурах не матимуть впливу на їх батьків. Розробники змінюють стан об'єкта лише тоді, коли змінюють його; тоді оновлюються лише конкретні компоненти. Стабільність коду та стабільна робота програми гарантується цим механізмом. Розглянемо бібліотеку TensorFlow.js. TensorFlow.js - це бібліотека JavaScript для навчання та розгортання моделей машинного навчання у веб-браузері та Node.js [23].

Tensorflow.js - це фреймворк JavaScript, розроблений Google, який дозволяє розробникам навчати та створювати моделі машинного навчання (ML) для використання в браузері [24]. Розробники також мають можливість використовувати попередньо навчені моделі замість того, щоб навчати свої власні.

Tensorflow.js швидкий, оскільки використовує WebGL (графічний API JavaScript), тому має чудову продуктивність. Існує Node.js версія TensorFlow, tfjs-node, яка пропонує ще більш ефективну роботу порівняно з браузерною версією.

Tensorflow.js дозволяє використовувати попередньо навчені моделі, а це означає, що розробники, які не знайомі з машинним навчанням, але хочуть інтегрувати його у свої проекти, можуть одразу ж почати створювати додатки - наприклад, додаток для розпізнавання аудіо - без необхідності витратити час на навчання власних моделей. Це робить машинне навчання більш доступним, ніж

раніше. TensorFlow.js також дозволяє завантажувати моделі, навчені за допомогою Python-версії TensorFlow. Це дозволяє розробникам створювати та навчати модель за допомогою Python, а потім завантажувати її у свій JavaScript-додаток. Перевага цієї техніки полягає в тому, що вона може значно підвищити продуктивність, оскільки вам не потрібно тренувати модель всередині браузера.

Можливо використовувати TensorFlow.js для перенавчання та доопрацювання вже існуючих моделей машинного навчання з даними, доступними лише на стороні клієнта в браузері, наприклад, зображеннями або відео з веб-камери.

Node.js — це однопоточне кросплатформове середовище виконання з відкритим вихідним кодом і бібліотека, яка використовується для запуску вебдодатків, написаних на JavaScript, поза браузером клієнта. Node.js — це програмне середовище, яке дозволяє запускати програми, написані мовою Javascript, поза браузером.

Visual Studio Code (VS Code) представляє собою безкоштовний текстовий редактор з відкритим кодом від компанії Microsoft, доступний для операційних систем Windows, Linux та macOS [25]. Незважаючи на свою легкість, редактор оснащений могутніми функціями, що зробили його однією з найпопулярніших розробницьких інструментів в останній час.

3.2 Програмна розробка інформаційної технології

Для початку необхідно створити головну сторінку для взаємодії користувача із системою. Код, що описує структуру головної сторінки:

```
function App() {  
  return (  
    <div className="app">  
      <header className="header">  
        <nav className="nav-menu"></nav>  
      </header>  
      <main className="app-main">  
        <section className="hero">
```

```

    <div className="hero-content">
      <h1>Підбір необхідного апаратного забезпечення</h1>
      <p>Використайте нашу поліку для підбору комплектуючих, що
задовольнять ваші потреби</p>
      <button className="hero-button">Почати підбір →</button>
    </div>
  </section>
</main>
</div>
);
}

```

Далі потрібно створити структуру модуль для вибору комп'ютерних комплектуючих, які потрібні користувачу або їх варіацій:

```

function Component() {
  return (
    <div className="component">
      <header className="component-header">
        Комп'ютерні комплектуючі
      </header>
      <ul className="component-list">
        <li>Відеокарта <span className="toggle-switch"></span></li>
        <li>Процесор <span className="toggle-switch"></span></li>
        <li>Жорсткий диск <span className="toggle-switch"></span></li>
        <li>Оперативна пам'ять <span className="toggle-switch"></span></li>
        <li>Материнська плата <span className="toggle-switch"></span></li>
      </ul>
      <button className="component-button">Почати підбір</button>
    </div>
  );
}

```

Створюємо модуль, що відповідатиме за потреби користувача та дозволить вказувати нечіткі параметри за якими буде здійснюватись підбір

```

function ComponentSelectionList() {
  const [selections, setSelections] = useState({
    videoCard: false,
    processor: false,
    hardDrive: false,
    ram: false,
    motherboard: false,
  });
}

```

```

const toggleSelection = (component) => {
  setSelections({
    ...selections,
    [component]: !selections[component],
  });
};

console.log(selections);
};

return (
  <div className="component-selection-list-container">
    {Object.keys(selections).map((component) => (
      <div key={component} className="component-item">
        <span>{component}</span>
        <span className={`toggle-switch ${selections[component] ? 'active' : ''}`
          onClick={() => toggleSelection(component)} />
      </div>
    ))}
    <button onClick={submitSelections}>Почати підбір</button>
  </div>
);
}

function GraphicsCardSelector() {
  const renderDropdown = () => {
    return (
      <select className="selector-dropdown">
        <option value="veryLow">Дуже низька</option>
        <option value="low">Низька</option>
        <option value="medium">Середня</option>
        <option value="high">Висока</option>
        <option value="veryHigh">Дуже висока</option>
      </select>
    );
  };
};

return (

```

```

<div className="graphics-card-selector">
  <div className="selector-header">Відеокарта</div>
  <div className="selector-body">
    <div className="selector-row">
      <span>Графіка</span>
      {renderDropdown()}
    </div>
    <div className="selector-row">
      <span>Ігри</span>
      {renderDropdown()}
    </div>
    <div className="selector-row">
      <span>Варість</span>
      {renderDropdown()}
    </div>
  </div>
  <button className="selector-button">Підібрати відеокарту</button>
</div>
);
}

```

Для виконання нечіткого підбору апаратного забезпечення з метою аналізу та якнайточнішого підбору використовуємо бібліотеку `brain.js`.

```

const brain = require('brain.js');
const net = new brain.NeuralNetwork();

const prepareData = (data) => {
  const inputs = data.map(card => [card.games_performance,
card.graphics_work_performance, card.price]);
  const labels = data.map(card => {

```

```

const label = card.type;
if (label === 'high_end') return [1, 0, 0];
if (label === 'mid_range') return [0, 1, 0];
if (label === 'entry_level') return [0, 0, 1];
});
return {
  inputs: tf.tensor2d(inputs),
  labels: tf.tensor2d(labels)
};
};
const { inputs, labels } = prepareData(videoCardsData);
const model = tf.sequential();
model.add(tf.layers.dense({inputShape: [3], units: 16, activation: 'relu'}));
model.add(tf.layers.dense({units: 3, activation: 'softmax'}));
model.compile({
  optimizer: tf.train.adam(),
  loss: 'categoricalCrossentropy',
  metrics: ['accuracy'],
});
const train = async () => {
  await model.fit(inputs, labels, {
    epochs: 100,
    callbacks: {
      onEpochEnd: (epoch, log) => console.log(`Epoch ${epoch}: loss = ${log.loss}`)
    }
  });
};
};

```


3.3 Тестування інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

На рисунку 3.1 зображено початкову сторінку інформаційної технології, яка виконана у вигляді веб-додатку.

Відкриємо головну сторінку для вибору апаратного забезпечення. Перейдемо до підбору комплектуючих натиснувши кнопку «Почати підбір».

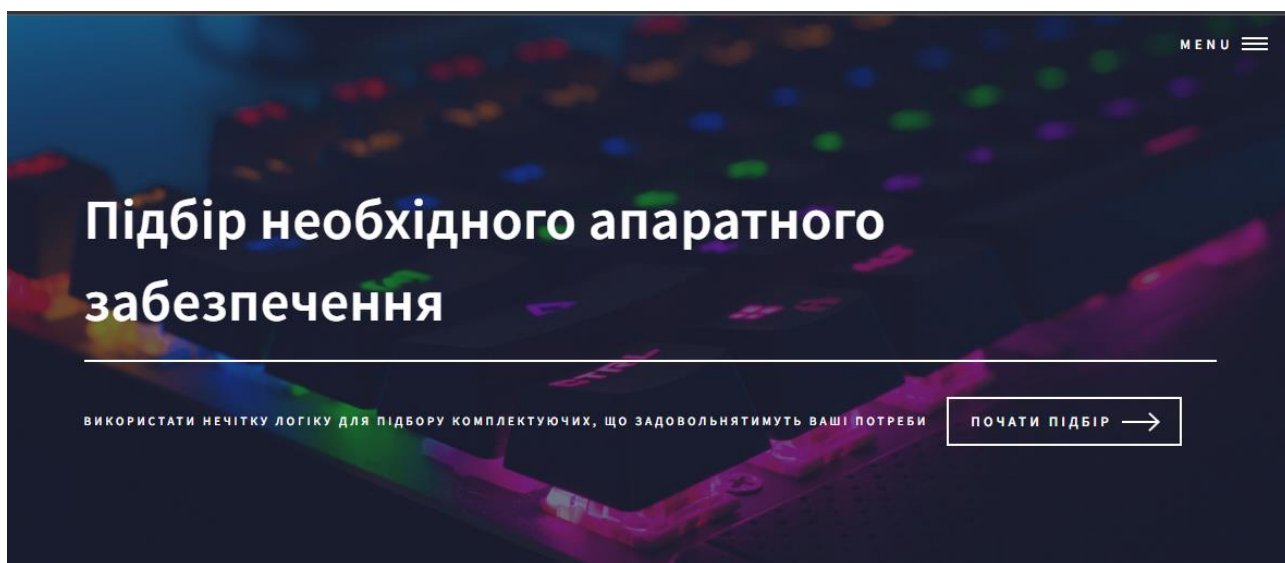


Рисунок 3.1 – Загальний вигляд інтерфейсного вікна сервісу для підбору апаратного забезпечення

На рисунку 3.2 нам буде відображено форму з комплектуючими. У залежності від потреб користувач вказує, які комплектуючі або їхні варіації він потребує.

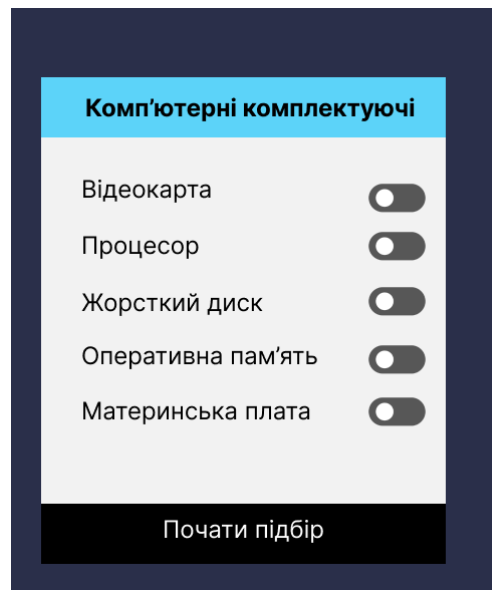


Рисунок 3.2 – Загальний вигляд інтерфейсного вікна доступних для вибору комплектуючих

На рисунку 3.3 вкажемо відеокарту та почнемо підбір.

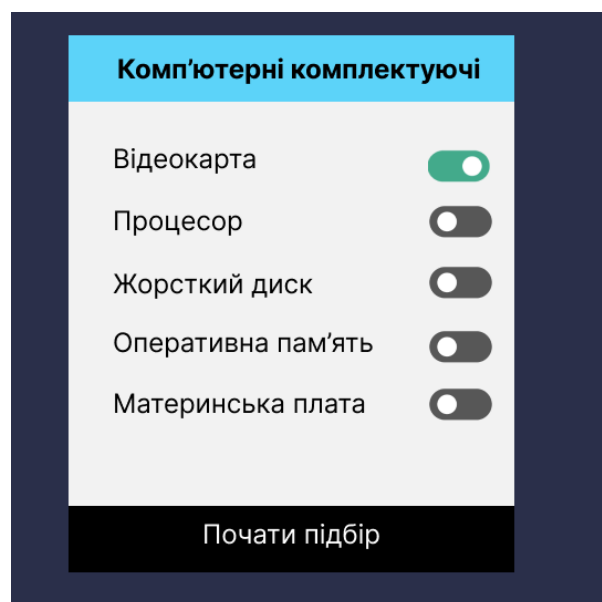


Рисунок 3.3 – Загальний вигляд інтерфейсного вікна вибору відеокарти серед комплектуючих

На рисунку 3.4 представлено форму, що відкриється за обраним варіантом і з потребами користувача, де можна задати ступінь тієї чи іншої вимоги.

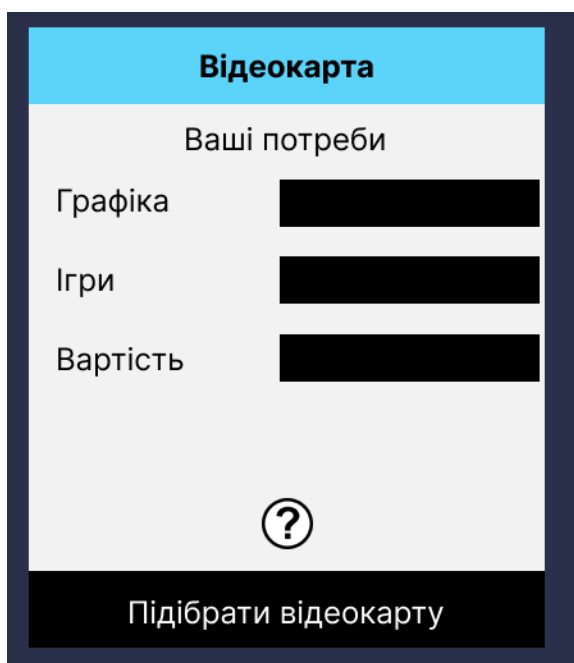


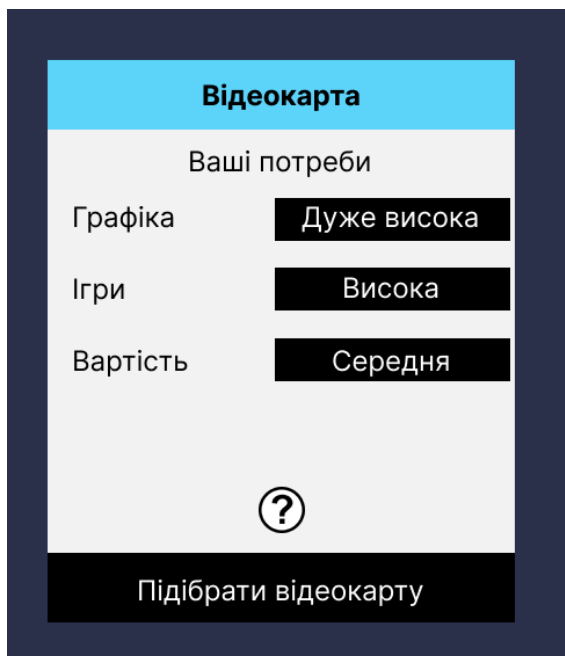
Рисунок 3.4 – Загальний вигляд інтерфейсного вікна критеріїв вибору відеокарти

Натиснувши іконку зі знаком питання просунутий користувач може переглянути інформацію з чисельними діапазонами значень, що відповідають нечітким значенням, які можна вказати в модулі підбору відеокарти. На рисунку 3.5 зображено форму, що відображає відповідні значення.

Графічні характеристики	Комп'ютерні ігри	Вартість
<p>1. Дуже Низькі:</p> <ul style="list-style-type: none"> • VRAM: ≤ 2 GB • Частота GPU: ≤ 1 GHz • Спеціалізовані функції: Мінімальні <p>2. Низькі:</p> <ul style="list-style-type: none"> • VRAM: 2-4 GB • Частота GPU: 1-1.4 GHz • Спеціалізовані функції: Підтримка базових графічних додатків <p>3. Середні:</p> <ul style="list-style-type: none"> • VRAM: 4-8 GB • Частота GPU: 1.4-1.8 GHz • Спеціалізовані функції: Підтримка 3D-модельовання та важких графічних додатків <p>4. Високі:</p> <ul style="list-style-type: none"> • VRAM: 8-16 GB • Частота GPU: 1.8-2 GHz • Спеціалізовані функції: Оптимізовані для високої продуктивності у професійних графічних програмах <p>5. Дуже Високі:</p> <ul style="list-style-type: none"> • VRAM: > 16 GB • Частота GPU: > 2 GHz • Спеціалізовані функції: Висока продуктивність для складних графічних проєктів і відеопродакшену 	<p>1. Дуже Низькі:</p> <ul style="list-style-type: none"> • VRAM: ≤ 2 GB • Частота GPU: ≤ 1 GHz • Підтримка DirectX: до DirectX 11 <p>2. Низькі:</p> <ul style="list-style-type: none"> • VRAM: 2-4 GB • Частота GPU: 1-1.4 GHz • Підтримка DirectX: DirectX 11 <p>3. Середні:</p> <ul style="list-style-type: none"> • VRAM: 4-8 GB • Частота GPU: 1.4-1.7 GHz • Підтримка DirectX: DirectX 12 <p>4. Високі:</p> <ul style="list-style-type: none"> • VRAM: 6-8 GB • Частота GPU: 1.7-2 GHz • Підтримка DirectX: DirectX 12, підтримка ray tracing для деяких моделей <p>5. Дуже Високі:</p> <ul style="list-style-type: none"> • VRAM: > 8 GB • Частота GPU: > 2 GHz • Підтримка DirectX: DirectX 12, підтримка ray tracing 	<p>1. Дуже Низькі:</p> <ul style="list-style-type: none"> • Ціна: до \$100 • Типові відеокарти: Основні моделі або старіші моделі. <p>2. Низькі:</p> <ul style="list-style-type: none"> • Ціна: \$100 - \$200 • Відеокарти для базових ігор та непрофесійної роботи з графікою. <p>3. Середні:</p> <ul style="list-style-type: none"> • Ціна: \$200 - \$500 • Відеокарти для загальних ігор та середнього рівня графічних завдань. <p>4. Високі:</p> <ul style="list-style-type: none"> • Ціна: \$500 - \$1000 • Відеокарти для вимогливих ігор та професійної роботи з графікою. <p>5. Дуже Високі:</p> <ul style="list-style-type: none"> • Ціна: \$1000 та вище • Відеокарти для найсучасніших ігор, складних графічних проєктів, та VR.

Рисунок 3.5 – Загальний вигляд інтерфейсного вікна діапазон значень критеріїв вибору

Відповідно до потреб в даному випадку користувач задає ступінь вимог до графіки, ігор та вартості. На рисунку 3.6 продемонстровано задання потреб користувача.



Відеокарта	
Ваші потреби	
Графіка	Дуже висока
Ігри	Висока
Вартість	Середня
?	
Підібрати відеокарту	

Рисунок 3.6 – Загальний вигляд інтерфейсного вікна задання критеріїв згідно вимог користувача

Задавши потреби користувачу буде виведено кілька результатів, що найкраще задовольняють його вимогам. На рисунку 3.7 зображено результат надання рекомендацій користувачу відповідно до заданих потреб.

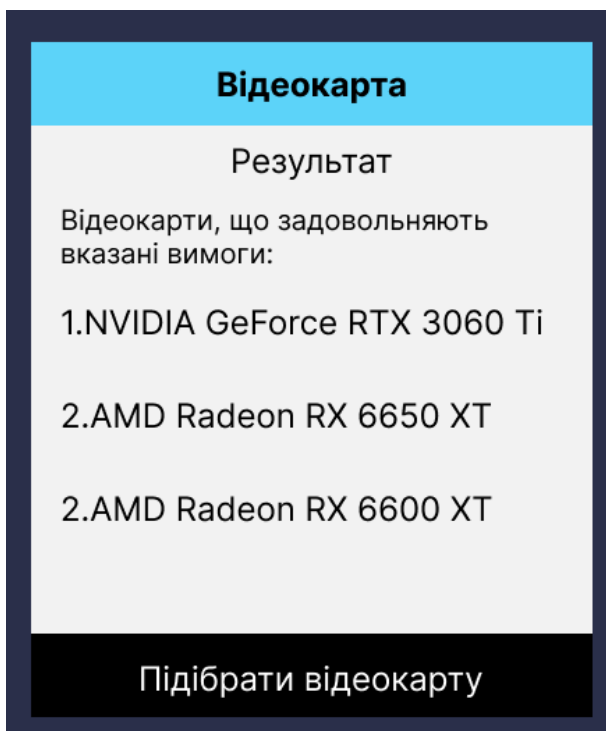


Рисунок 3.7 – Загальний вигляд інтерфейсного вікна надання рекомендацій, що задовольняють вимогам

3.4 Аналіз результатів тестування програми

Результати тестування інформаційної технології, яка надає рекомендації для підбору апаратного забезпечення, були проведені з використанням бази даних про актуальні комплектуючі та їх характеристики. У процесі аналізу бралися до уваги такі параметри, як продуктивність у іграх та в роботі з графікою, а також вартість.

Для тестування було виконано 100 повторів підбору конфігурації.

Протестувавши програму, ми порівняли її з результатами попередніх досліджень, які використовували стандартні підходи до вибору комплектуючих.

За результатами тестування, представленими в таблиці 3.1, використання розробленої системи рекомендацій показало поліпшення точності підбору комплектуючих на 7.81% у порівнянні зі стандартними методами, що свідчить про ефективність запропонованого підходу.

Таблиця 3.1 – Тестування та аналіз результатів роботи інформаційної технології

Номер конфігурації	1	2	3	4	5	6	7
Загальна вартість компонентів, \$	1050	950	1200	1100	900	850	1000
Комп'ютерні ігри, бали	4500	4200	5000	4700	4000	3800	4600
Графічні характеристики, бали	2700	2750	3500	3000	2500	2450	2500
Співвідношення ціна/продуктивність	0.8748	0.8202	0.8472	0.8574	0.831	0.816	0.8448
Відсоток покращення надання рекомендацій	8.7%	7.4%	8.1%	8.3%	7.7%	7.1%	7.4%

3.5 Висновок до розділу 3

1. Проаналізовано мови програмування та обрано JavaScript та її бібліотеки для розробки інформаційної технології.

2. Подано структуру програмного забезпечення, яке дозволяє користувачам вказувати потрібну комплектуючу або різні варіації комплектуючих. Також створено модуль для підбору апаратного забезпечення, на основі нечітких значень потреб користувача.

3. Роботу інформаційної технології протестовано на підборі відеокарти і отримано результат, що задовольняє вказані потреби користувача.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Критерії	Експерти		
	Озеранський В.С.	Сілагін О.В.	Арсенюк І.Р
	Бали, виставлені експертами		
Технічна здійсненність концепції	2	3	3
Ринкові переваги (наявність аналогів)	3	2	3
Ринкові переваги (ціна продукту)	3	4	3
Ринкові переваги (технічні властивості)	2	3	4
Ринкові переваги (експлуатаційні витрати)	3	3	3
Ринкові перспективи (розмір ринку)	4	3	4

Продовження таблиці 4.1

Ринкові перспективи (конкуренція)	3	3	4
Практична здійсненність (наявність фахівців)	4	4	4
Практична здійсненність (наявність фінансів)	3	3	3
Практична здійсненність (необхідність нових матеріалів)	3	2	3
Практична здійсненність (термін реалізації)	3	3	3
Практична здійсненність (розробка документів)	3	2	3
Сума балів	36	35	40
Середньоарифметична сума балів, СБ	37		

За результатами розрахунків, наведених в таблиці 4.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем – вище середнього.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату

дослідників (Z_o) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	21500	977,3	35	34205
Розробник	16700	759,1	74	56173
Всього:				90378

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C і можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2023 році $M_m=6700$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Зроблені розрахунки зводимо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Огляд проблемної області підбору апаратного забезпечення комп'ютерних систем	10	2	41,5	1,09	415
Аналіз сучасних методів підбору	10	3	44,9	1,18	449
Аналіз методів надання рекомендацій	20	3	44,9	1,18	898
Розробка методу формування рекомендацій	40	4	48,4	1,27	1936
Розробка експертної системи надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	80	5	51,8	1,36	4144
Розробка алгоритму надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем	30	3	44,9	1,18	1347

Продовження таблиці 4.3

Розробка програмного забезпечення	360	6	55,2	1,45	19872
Тестування реалізованої інформаційної технології	40	5	51,8	1,36	2072
Всього					30733

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (90378 + 30733) = 12111,1 \text{ грн.} \quad (4.4)$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned} H_{зп} &= \beta \cdot (Z_o + Z_p + Z_d) = \\ &= 0,22 \cdot (90378 + 30733 + 12111,1) = 29309 \text{ грн.} \end{aligned} \quad (4.5)$$

де Z_o – основна заробітна плата розробників, грн.; Z_p – основна заробітна плата робітників, грн.; Z_d – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Розрахунок витрат на матеріали. Витрати на матеріали M , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.6)$$

де H_i – кількість матеріалів i -го виду, шт.; C_i – ціна матеріалів i -го виду, грн.; K_i – коефіцієнт транспортних витрат, $K_i = (1,1\dots1,15)$; n – кількість видів матеріалів.

Зроблені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Комплектуючі, що використані на розробку

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	270	1	270
Упаковка паперу	140	1	140
Всього, з врахуванням коефіцієнта транспортних витрат			451

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i, \quad (4.7)$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення i -го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1,1\dots1,12)$; k – кількість видів програмного забезпечення.

Зроблені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Microsoft Windows	1150	1	1150
Microsoft Office	1820	1	1820

Продовження таблиці 4.5

Всього, з врахуванням коефіцієнта інсталяції та налагодження	3267
--------------------------------------------------------------	------

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_б}{T_в} \cdot \frac{t}{12}, \quad (4.8)$$

де $Ц_б$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного використання основного фонду, роки.

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 4.6 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук Acer Aspire 5 a517-51g	30000	5	2,5	
Всього				1250

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію B_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Зроблені розрахунки зводимо до таблиці 4.7.

Таблиця 4.7 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук Acer Aspire 5 a517-51g	0,065	590
Освітлення	0,022	590

$$B_e = \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{vni}}{\text{ККД}} = \frac{0,065 \cdot 590 \cdot 7,5 \cdot 0,85}{0,98} + \frac{0,022 \cdot 590 \cdot 7,5 \cdot 0,85}{0,98} = 334 \text{ грн.}, \quad (4.9)$$

де W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; K_{vni} – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_B = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%} = (90378 + 30733) \cdot \frac{75}{100} = 90833 \text{ грн.}, \quad (4.10)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та

ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нз}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%} = (90378 + 30733) \cdot \frac{140}{100} = 169555 \text{ грн.}, \quad (4.11)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} V_{\text{заг}} &= Z_o + Z_p + Z_{\text{дод}} + Z_{\text{н}} + M + V_{\text{прг}} + A_{\text{обл}} + V_e + \quad (4.12) \\ &\quad + I_v + V_{\text{нзв}} = \\ &= 90378 + 30733 + 12111,1 + 29309 + 451 + 3267 + 1250 + 334 + 90833 \\ &\quad + 169555 = 428221 \text{ грн.} \end{aligned}$$

Загальні витрати. Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{V_{\text{заг}}}{\eta} = \frac{428221}{0,9} = 475801 \text{ грн.}, \quad (4.13)$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

У ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

У даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; C_0 – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta C_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.14)$$

де $\pm\Delta C$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta C_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні

розробки у році до впровадження результатів нової науково-технічної розробки; C_o – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; C_b – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda=0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho =0,2\dots0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta =18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\Pi = ((75000 - 45000) \cdot 300 - (370 - 300) \cdot 45000) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 1286485 \text{ грн.} \quad (4.15)$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t} = \frac{1286485}{(1+0,1)^1} = 1169532 \text{ грн.}, \quad (4.16)$$

де $\Delta\Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо

$T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B = 2 \cdot 475801 = 951602 \text{ грн.}, \quad (4.17)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2 \dots 5$, але може бути і більшим; $3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 1169532 - 951602 = 217930 \text{ грн.}, \quad (4.18)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_e або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну

дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} = \sqrt[1]{1 + \frac{217930}{951602}} = 1,1, \quad (4.19)$$

де $T_{ж}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{мін}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f = 0,9 + 0,05 = 0,95, \quad (4.20)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,9...0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05...0,5$, але може бути і значно вищою.

Оскільки $E_B = 1,1 > \tau_{мін} = 0,95$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{1,1} = 0,9 \text{ року.} \quad (4.21)$$

Оскільки $T_o=1,1 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновок до розділу 4

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 475801 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,9 роки.

ВИСНОВКИ

У процесі виконання магістерської роботи було розроблено інформаційну технології для надання рекомендацій по підбору апаратного забезпечення комп'ютерних систем. Основний акцент дослідження було зосереджено на аналізі існуючих технологічних рішень, методів та алгоритмів, які застосовуються для автоматизованого підбору апаратного забезпечення.

Було проведено аналіз предметної області, розглянуто ключові задачі та вимоги, що висуваються до сучасних систем підбору апаратного забезпечення. Також було проаналізовано сервіси, які надають рекомендації по підбору апаратного забезпечення, та визначено їхні сильні і слабкі сторони, а також можливості для інновацій і покращення.

На основі аналізу та дослідження, було обрано та обґрунтовано методологічний підхід до розробки нової інформаційної технології, яка буде орієнтована на високу точність та персоналізацію рекомендацій, а також на адаптивність до спектру потреб користувачів. Розроблений метод оцінювання апаратного забезпечення здатен адаптуватися до швидкозмінних технічних характеристик та специфікацій. Пропонований алгоритм оптимізує процес вибору апаратного забезпечення, ефективно інтегруючи дані та динаміку змін у технологічних характеристиках та користувацьких вимогах. Розроблено систему надання рекомендацій підбору апаратного забезпечення для комп'ютерних систем на основі методу нечітких множин.

Розроблено програмне забезпечення, що надає рекомендації при виборі апаратного забезпечення комп'ютерних систем, засобами мови програмування JavaScript із застосуванням бібліотек React.js, TensorFlow.js та Node.js. Також створено модуль для підбору апаратного забезпечення, на основі нечітких значень потреб користувача. Було здійснено тестування розробленого веб-додатку, при наданні рекомендацій було досягнуто покращення точності на 7.81%.

Було виконано розрахунок витрат на розробку нового програмного

продукту, сума яких складає 475801 гривень. Спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. У результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,9 роки.

Таким чином, передбачається, що реалізація проекту сприятиме значному покращенню процесу підбору апаратного забезпечення, забезпечуючи користувачам не тільки технічні рекомендації, але і зорієтовуючи їх на продукцію, яка найбільше відповідає їх індивідуальним потребам та бюджету.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сапіташ В. С., Іванчук Я. В. «Алгоритм надання рекомендацій для підбору комплектуючих апаратного забезпечення комп'ютерних систем,» в Матеріали конференції «LII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2023)», Вінниця, 2023. С. 491-493. [Електронний ресурс]. Режим доступу: <https://press.vntu.edu.ua/index.php/vntu/catalog/view/788/1373/2632-1>. Дата звернення: Черв. 2023.

2. Сапіташ В.С., Іванчук Я.В. ВЕБ-додаток для онлайн продажу авіаквитків. Матеріали конференції «LI Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2022)», м. Вінниця, Україна. 2022. – Режим доступу: https://conferences.vntu.edu.ua/public/files/1/vntu_2022_netpub.pdf (дата звернення: 23.05.2022).

3. Програмне та апаратне забезпечення: поняття, призначення, рівні, характеристики та налаштування – [Електронний ресурс]. – Режим доступу: <https://what.com.ua/programne-ta-aparatne-zabezpec/>.

4. Яку конфігурацію вибрати для ігрового комп'ютера? Скільки коштує зібрати ігровий ПК – [Електронний ресурс]. – Режим доступу: <https://allmaster.com.ua/ua/stati/yaku-konfiguraciyu-vibrati-dlya-igrovogo-pk-karakteristiki-igrovogo-kompyutera>.

5. Як підібрати комплектуючі для ПК: важливі нюанси - [Електронний ресурс]. – Режим доступу: <https://pc.com.ua/ua/articles/kak-podobrat-komplektujushhie-dlja-pk-vazhnye-njuansy>.

6. How to build a PC – [Електронний ресурс]. – Режим доступу: <https://www.techradar.com/how-to/how-to-build-a-pc>.

7. Intel – [Електронний ресурс]. – Режим доступу: <https://www.intel.com/content/www/us/en/homepage.html>.

8. Nvidia – [Електронний ресурс]. – Режим доступу:

<https://www.nvidia.com>.

9. AMD – [Електронний ресурс]. – Режим доступу: <https://www.amd.com/en.html>.

10. Amazon.com. Spend less. Smile more – [Електронний ресурс]. – Режим доступу: <https://www.amazon.com/>.

11. Rozetka – [Електронний ресурс]. – Режим доступу: <https://rozetka.com.ua/ua/>.

12. Reddit - Dive into anything - [Електронний ресурс]. – Режим доступу: <https://www.reddit.com/>.

13. TechRadar | the technology experts – [Електронний ресурс]. – Режим доступу: <https://www.techradar.com/>.

14. PCMag: The Latest Technology Product Reviews, News, Tips – [Електронний ресурс]. – Режим доступу: <https://www.pcmag.com/>.

15. PCPartPicker: Pick parts. Build your PC. Compare and share. – [Електронний ресурс]. – Режим доступу: <https://pcpartpicker.com/>.

16. Яка пам'ять сумісна з моїм ноутбуком, ПК або сервером? – [Електронний ресурс]. – Режим доступу: <https://www.kingston.com/ua/memory>.

17. ORIGIN PC | Gaming PCs, Gaming Laptops, Custom Computers – [Електронний ресурс]. – Режим доступу: <https://www.originpc.com/>.

18. Подання та обробка нечітких знань – [Електронний ресурс]. – Режим доступу: <http://um.co.ua/3/3-13/3-13119.html>.

19. Месюра В.І. Експертні системи. Частина 1. Навчальний посібник /В.І. Месюра, А.А. Яровий, І.Р. Арсенюк— Вінниця: ВНТУ, 2006.— 114 с.

20. What Is JavaScript & What Is It Used For? A Basic Guide to JS <https://www.semrush.com/blog/javascript/>.

21. What is React.js? Uses, Examples, & More – [Електронний ресурс]. – Режим доступу: <https://www.pcmag.com/>.

22. What is React.js? Uses, Examples, & More – [Електронний ресурс]. – Режим доступу: <https://blog.hubspot.com/website/react-js> .

23. Що таке Node JS простими словами – [Електронний ресурс]. – Режим

доступу: <https://dan-it.com.ua/uk/blog/chtu-jeto-takoe-node-js-prostymi-slovami/>.

24. Get started with TensorFlow.js – [Электронный ресурс]. – Режим доступа: <https://www.tensorflow.org/js/tutorials>.

25. Знайомство з Visual Studio Code – [Электронный ресурс]. – Режим доступа: <https://romul.name/blog/znayomstvo-z-visual-studio-code/>.

ДОДАТКИ

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія надання рекомендацій для підбору апаратного забезпечення комп'ютерних систем

Тип роботи: магістерська кваліфікаційна робота
(БДР, МРР)

Підрозділ кафедра комп'ютерних наук, ФШТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 88,9% Схожість 11,1%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи

Сапіташ В.С.

Керівник роботи

Іванчук Я.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку

Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>

```

```

import React, { useState, useEffect } from 'react';
import * as tf from '@tensorflow/tfjs';
function App() {
  import React, { useState, useEffect } from 'react';
  import * as tf from '@tensorflow/tfjs';

```

```

const PredictionComponent = () => {
  const [prediction, setPrediction] = useState("");
  const buildAndTrainModel = async () => {
  };
  const makePrediction = async (inputData) => {
  };
  useEffect(() => {
    buildAndTrainModel();
  }, []);

```

```

const handlePredictClick = () => {
  const inputData = [0.8, 0.2, 0.5];
  makePrediction(inputData);
};

```

```

return (
  <div>
    <h1>Вибір апаратних компонентів</h1>
    <button onClick={handlePredictClick}>Робити прогноз</button>
    <p>Рекомендація: {prediction}</p>
  </div>
);
};

export default PredictionComponent;
return (
  <div className="app">
    <header className="header">
      <nav className="nav-menu"></nav>
    </header>
    <main className="app-main">
      <section className="hero">
        <div className="hero-content">
          <h1>Підбір необхідного апаратного забезпечення</h1>
          <p>Використайте нашу поліку для підбору комплектуючих, що
задовольняють ваші потреби</p>
          <button className="hero-button">Почати підбір →</button>
        </div>
      </section>
    </main>
  </div>
);
}
.header {
  background-color: #1a1a1a;
  color: white;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.header h1 {
  font-size: 24px;
  margin: 0;
}

.nav-menu {
  display: flex;

```

```

align-items: center;
gap: 20px;
}

```

```

.nav-menu a {
  color: white;
  text-decoration: none;
  font-size: 18px;
  padding: 10px 20px;
  border: 1px solid transparent;
  transition: border-color 0.3s ease;
}

```

```

.nav-menu a:hover,
.nav-menu a:focus {
  border-color: #00adee;
}

```

```

@media (max-width: 768px) {
  .header h1 {
    font-size: 20px;
  }
}

```

```

.nav-menu a {
  font-size: 16px;
  padding: 8px 16px;
}
}

```

```

function Component() {
  return (
    <div className="component">
      <header className="component-header">
        Комп'ютерні комплектуючі
      </header>
      <ul className="component-list">
        <li>Відеокарта <span className="toggle-switch"></span></li>
        <li>Процесор <span className="toggle-switch"></span></li>
        <li>Жорсткий диск <span className="toggle-switch"></span></li>
        <li>Оперативна пам'ять <span className="toggle-switch"></span></li>
        <li>Материнська плата <span className="toggle-switch"></span></li>
      </ul>
      <button className="component-button">Почати підбір</button>
    </div>
  );
}

```

```
);  
}  
  
.hardware-selection {  
  background-color: #f0f0f0;  
  border-radius: 8px;  
  padding: 20px;  
  margin: 20px;  
  box-shadow: 0 2px 5px rgba(0,0,0,0.2);  
}  
  
.hardware-selection h2 {  
  color: #333;  
  font-size: 20px;  
  margin-bottom: 16px;  
}  
  
.hardware-option {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  margin-bottom: 10px;  
}  
  
.hardware-option label {  
  color: #333;  
  font-size: 18px;  
  margin-right: 10px;  
}  
  
.hardware-option input[type='checkbox'] {  
  accent-color: #007bff;  
  width: 24px;  
  height: 24px;  
  cursor: pointer;  
}  
  
@media (max-width: 768px) {  
  .hardware-selection h2 {  
    font-size: 18px;  
  }  
  
  .hardware-option label {  
    font-size: 16px;  
  }  
}
```

```

    }
import React from 'react';
import './HardwareSelection.css'; // Import the CSS file

const HardwareSelection = ({ onSelectionChange }) => {
  const hardwareOptions = ['Відеокарта', 'Процесор', 'Жорсткий диск', 'Оперативна
пам'ять', 'Материнська плата'];

  return (
    <div className="hardware-selection">
      <h2>Комп'ютерні комплектуючі</h2>
      {hardwareOptions.map((option, index) => (
        <div key={index} className="hardware-option">
          <label>{option}</label>
          <input type="checkbox" onChange={() => onSelectionChange(option)} />
        </div>
      ))}
    </div>
  );
};

export default HardwareSelection;
}
function ComponentSelectionList() {
  const [selections, setSelections] = useState({
    videoCard: false,
    processor: false,
    hardDrive: false,
    ram: false,
    motherboard: false,
  });

  const toggleSelection = (component) => {
    setSelections({
      ...selections,
      [component]: !selections[component],
    });
  };

  console.log(selections);
};

return (
  <div className="component-selection-list-container">

```



```

    {Object.keys(selections).map((component) => (
      <div key={component} className="component-item">
        <span>{component}</span>
        <span className={`toggle-switch ${selections[component] ? 'active' : ''}`
          onClick={() => toggleSelection(component)} />
      </div>
    ))}
    <button onClick={submitSelections}>Почати підбір</button>
  </div>
);
}

```

```

function GraphicsCardSelector() {
  // Function to render a dropdown for options
  const renderDropdown = () => {
    return (
      <select className="selector-dropdown">
        <option value="veryLow">Дуже низька</option>
        <option value="low">Низька</option>
        <option value="medium">Середня</option>
        <option value="high">Висока</option>
        <option value="veryHigh">Дуже висока</option>
      </select>
    );
  };

  return (
    <div className="graphics-card-selector">
      <div className="selector-header">Відеокарта</div>
      <div className="selector-body">
        <div className="selector-row">
          <span>Графіка</span>
          {renderDropdown()}
        </div>
        <div className="selector-row">
          <span>Ігри</span>
          {renderDropdown()}
        </div>
        <div className="selector-row">
          <span>Варість</span>
          {renderDropdown()}
        </div>
      </div>
      <button className="selector-button">Підібрати відеокарту</button>
    </div>
  );
}

```

```

</div>
);
}[
{
  "model": "NVIDIA GeForce RTX 3090",
  "brand": "NVIDIA",
  "series": "RTX 3000",
  "vram": "24GB",
  "base_clock": "1.40 GHz",
  "boost_clock": "1.70 GHz",
  "cuda_cores": 10496,
  "memory_type": "GDDR6X",
  "price": "Around $1499"
},
{
  "model": "NVIDIA GeForce RTX 3080",
  "brand": "NVIDIA",
  "series": "RTX 3000",
  "vram": "10GB",
  "base_clock": "1.44 GHz",
  "boost_clock": "1.71 GHz",
  "cuda_cores": 8704,
  "memory_type": "GDDR6X",
  "price": "Around $699"
},
{
  "model": "NVIDIA GeForce RTX 3070",
  "brand": "NVIDIA",
  "series": "RTX 3000",
  "vram": "8GB",
  "base_clock": "1.50 GHz",
  "boost_clock": "1.73 GHz",
  "cuda_cores": 5888,
  "memory_type": "GDDR6",
  "price": "Around $499"
},
{
  "model": "NVIDIA GeForce RTX 3060 Ti",
  "brand": "NVIDIA",
  "series": "RTX 3000",
  "vram": "8GB",
  "base_clock": "1.41 GHz",
  "boost_clock": "1.67 GHz",
  "cuda_cores": 4864,

```

```
"memory_type": "GDDR6",
"price": "Around $399"
},
{
  "model": "AMD Radeon RX 6900 XT",
  "brand": "AMD",
  "series": "RX 6000",
  "vram": "16GB",
  "base_clock": "1.82 GHz",
  "boost_clock": "2.25 GHz",
  "stream_processors": 5120,
  "memory_type": "GDDR6",
  "price": "Around $999"
},
{
  "model": "AMD Radeon RX 6800 XT",
  "brand": "AMD",
  "series": "RX 6000",
  "vram": "16GB",
  "base_clock": "1.82 GHz",
  "boost_clock": "2.25 GHz",
  "stream_processors": 4608,
  "memory_type": "GDDR6",
  "price": "Around $649"
},
{
  "model": "AMD Radeon RX 6800",
  "brand": "AMD",
  "series": "RX 6000",
  "vram": "16GB",
  "base_clock": "1.70 GHz",
  "boost_clock": "2.10 GHz",
  "stream_processors": 3840,
  "memory_type": "GDDR6",
  "price": "Around $579"
},
{
  "model": "AMD Radeon RX 6700 XT",
  "brand": "AMD",
  "series": "RX 6000",
  "vram": "12GB",
  "base_clock": "2.32 GHz",
  "boost_clock": "2.58 GHz",
  "stream_processors": 2560,
```

```

    "memory_type": "GDDR6",
    "price": "Around $479"
  },
  {
    "model": "AMD Radeon RX 6600 XT",
    "brand": "AMD",
    "series": "RX 6000",
    "vram": "8GB",
    "base_clock": "2.36 GHz",
    "boost_clock": "2.58 GHz",

import * as tf from '@tensorflow/tfjs';
const videoCardsData = [
];
[
  {
    "model": "AMD Ryzen 9 5950X",
    "cores": 16,
    "threads": 32,
    "base_clock": 3.4,
    "boost_clock": 4.9,
    "tdp": 105,
    "price": 799
  },
  {
    "model": "AMD Ryzen 9 5900X",
    "cores": 12,
    "threads": 24,
    "base_clock": 3.7,
    "boost_clock": 4.8,
    "tdp": 105,
    "price": 549
  },
  {
    "model": "AMD Ryzen 7 5800X",
    "cores": 8,
    "threads": 16,
    "base_clock": 3.8,
    "boost_clock": 4.7,
    "tdp": 105,
    "price": 449
  },
  {
    "model": "AMD Ryzen 5 5600X",

```

```
"cores": 6,  
"threads": 12,  
"base_clock": 3.7,  
"boost_clock": 4.6,  
"tdp": 65,  
"price": 299  
},  
{  
  "model": "Intel Core i9-11900K",  
  "cores": 8,  
  "threads": 16,  
  "base_clock": 3.5,  
  "boost_clock": 5.3,  
  "tdp": 125,  
  "price": 539  
},  
{  
  "model": "Intel Core i7-11700K",  
  "cores": 8,  
  "threads": 16,  
  "base_clock": 3.6,  
  "boost_clock": 5.0,  
  "tdp": 125,  
  "price": 399  
},  
{  
  "model": "Intel Core i5-11600K",  
  "cores": 6,  
  "threads": 12,  
  "base_clock": 3.9,  
  "boost_clock": 4.9,  
  "tdp": 125,  
  "price": 262  
},  
{  
  "model": "Intel Core i5-10600K",  
  "cores": 6,  
  "threads": 12,  
  "base_clock": 4.1,  
  "boost_clock": 4.8,  
  "tdp": 125,  
  "price": 237  
},  
{
```

```

    "model": "Intel Core i3-10100",
    "cores": 4,
    "threads": 8,
    "base_clock": 3.6,
    "boost_clock": 4.3,
    "tdp": 65,
    "price": 122
  },
  {
    "model": "Intel Core i3-9350KF",
    "cores": 4,
    "threads": 4,
    "base_clock": 4.0,
    "boost_clock": 4.6,
    "tdp": 91,
    "price": 148
  }
]
const prepareData = (data) => {
  const inputs = data.map(card => [card.games_performance,
card.graphics_work_performance, card.price]);
  const labels = data.map(card => {

    const label = card.type;
    if (label === 'high_end') return [1, 0, 0];
    if (label === 'mid_range') return [0, 1, 0];
    if (label === 'entry_level') return [0, 0, 1];
  });
  return {
    inputs: tf.tensor2d(inputs),
    labels: tf.tensor2d(labels)
  };
};
const { inputs, labels } = prepareData(videoCardsData);
const model = tf.sequential();
model.add(tf.layers.dense({inputShape: [3], units: 16, activation: 'relu'}));
model.add(tf.layers.dense({units: 3, activation: 'softmax'}));
model.compile({
  optimizer: tf.train.adam(),
  loss: 'categoricalCrossentropy',
  metrics: ['accuracy'],
});
const train = async () => {
  await model.fit(inputs, labels, {

```

```

    epochs: 100,
    callbacks: {
      onEpochEnd: (epoch, log) => console.log(`Epoch ${epoch}: loss = ${log.loss}`)
    }
  });
};
const predict = (gamesPerformance, graphicsWorkPerformance, price) => {
  const inputTensor = tf.tensor2d([[gamesPerformance, graphicsWorkPerformance,
price]]);
  const prediction = model.predict(inputTensor);
  return prediction.arraySync();
};
train().then(() => {
  const userPreferences = { gamesPerformance: 0.8, graphicsWorkPerformance: 0.2,
price: 0.5 };
  const [highEndProb, midRangeProb, entryLevelProb] =
predict(userPreferences.gamesPerformance,
userPreferences.graphicsWorkPerformance, userPreferences.price);

  console.log(`Прогноз: High End - ${highEndProb}, Mid Range -
${midRangeProb}, Entry Level - ${entryLevelProb}`);
}); console.log(`Epoch ${epoch}: loss = ${logs.loss}`);
}
}
});
console.log("Тренування завершено!");
return response;
}
async function runModel(inputData) {
  const inputTensor = tf.tensor2d([inputData], [1, 3]);
  const prediction = model.predict(inputTensor);

  const index = prediction.argmax(1).dataSync()[0]; ймовірності
  const categories = ['high_end', 'mid_range', 'professional', 'entry_level'];
  console.log(`Рекомендація: ${categories[index]}`);
}
trainModel().then(() => {
  runModel([0.8, 0.2, 0.5]); // Потреби користувача
});
const processorsData = [];
const prepareData = (data) => {
  const inputs = data.map(proc => [proc.cores, proc.base_clock, proc.boost_clock,
proc.tdp, proc.price]);
  const labels = data.map(proc => {

```

```

    const label = proc.performance;
    if (label === 'high') return [1, 0, 0];
    if (label === 'medium') return [0, 1, 0];
    if (label === 'low') return [0, 0, 1];
  });

  return {
    inputs: tf.tensor2d(inputs),
    labels: tf.tensor2d(labels)
  };
};

const { inputs, labels } = prepareData(processorsData);
const model = tf.sequential();
model.add(tf.layers.dense({inputShape: [5], units: 16, activation: 'relu'}));
model.add(tf.layers.dense({units: 3, activation: 'softmax'}));
model.compile({
  optimizer: tf.train.adam(),
  loss: 'categoricalCrossentropy',
  metrics: ['accuracy'],
});

const train = async () => {
  await model.fit(inputs, labels, {
    epochs: 100,
    callbacks: {
      onEpochEnd: (epoch, log) => console.log(`Epoch ${epoch}: loss = ${log.loss}`)
    }
  });
};

const predict = (cores, baseClock, boostClock, tdp, price) => {
  const inputTensor = tf.tensor2d([[cores, baseClock, boostClock, tdp, price]]);
  const prediction = model.predict(inputTensor);
  return prediction.arraySync();
};

train().then(() => {
  const userPreferences = { cores: 8, baseClock: 3.6, boostClock: 4.9, tdp: 95, price:
  300 };
  const [highPerfProb, medPerfProb, lowPerfProb] = predict(userPreferences.cores,
  userPreferences.baseClock, userPreferences.boostClock, userPreferences.tdp,
  userPreferences.price);

  console.log(`Прогноз: High Performance - ${highPerfProb}, Medium Performance
  - ${medPerfProb}, Low Performance - ${lowPerfProb}`);
})

```


Додаток В (Обов'язковий)

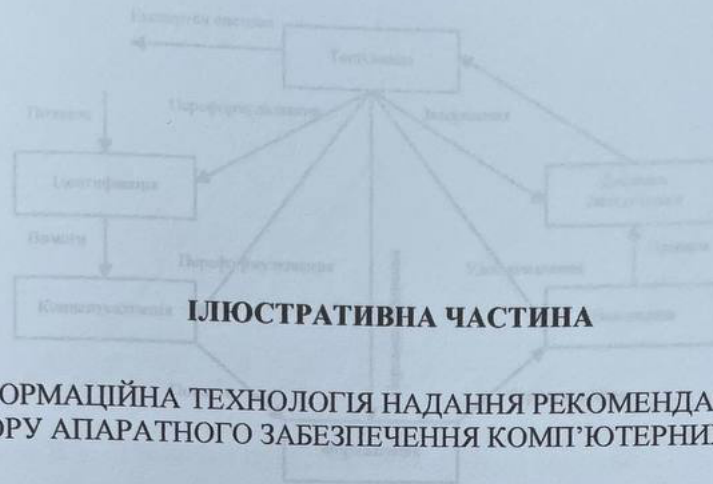


Рисунок В.1 – Схема етапів розробки експертних систем

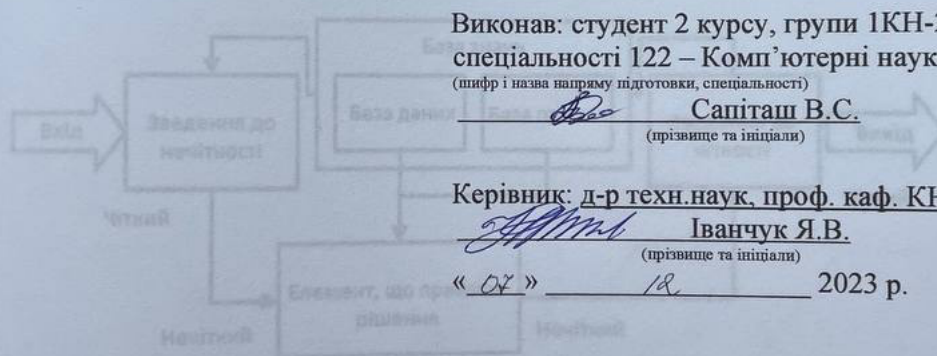


Рисунок В.2 – Схема функціонування системи нечіткого висновку

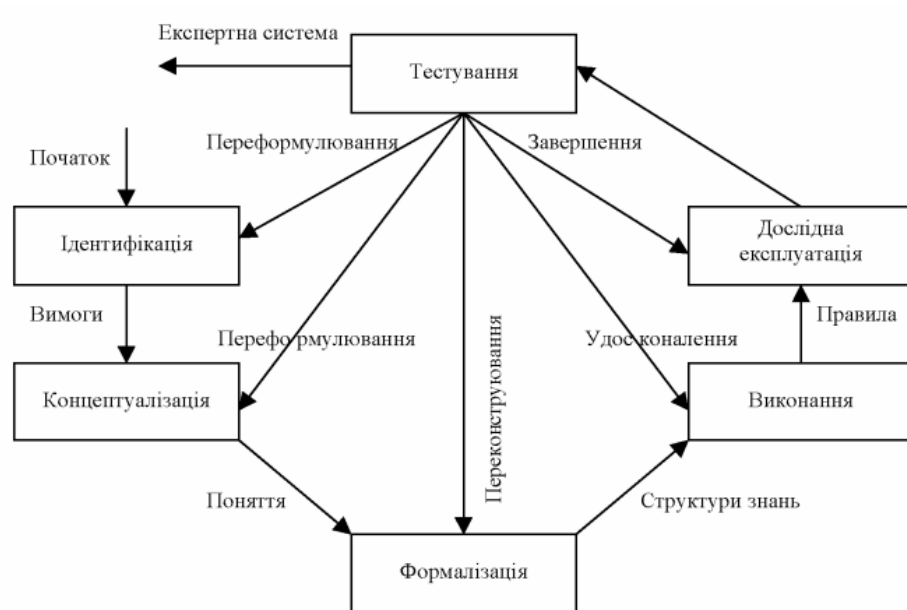


Рисунок В.1 – Схема етапів розробки експертних систем

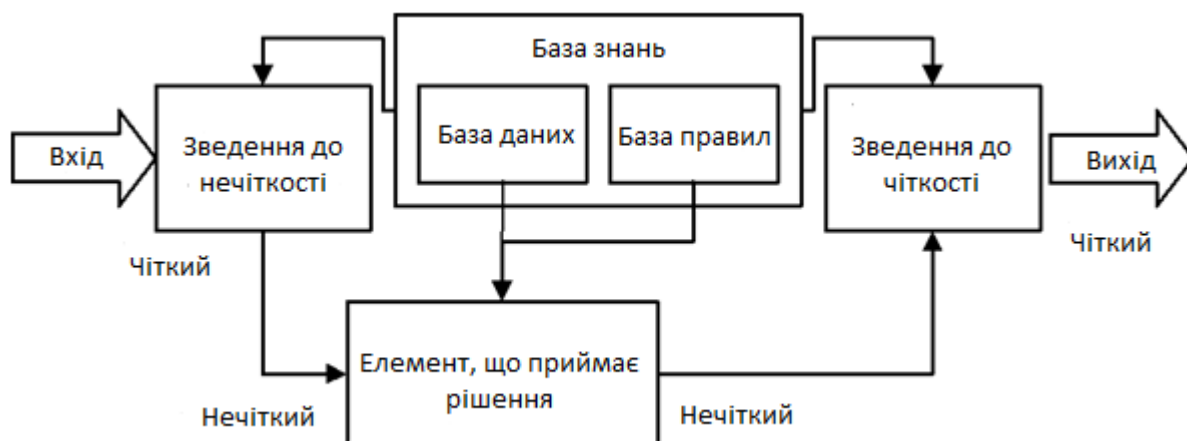


Рисунок В.2 – Схема функціонування системи нечіткого висновку



Рисунок В.3 – Графічне представлення функції належності лінгвістичної змінної «Процесор»

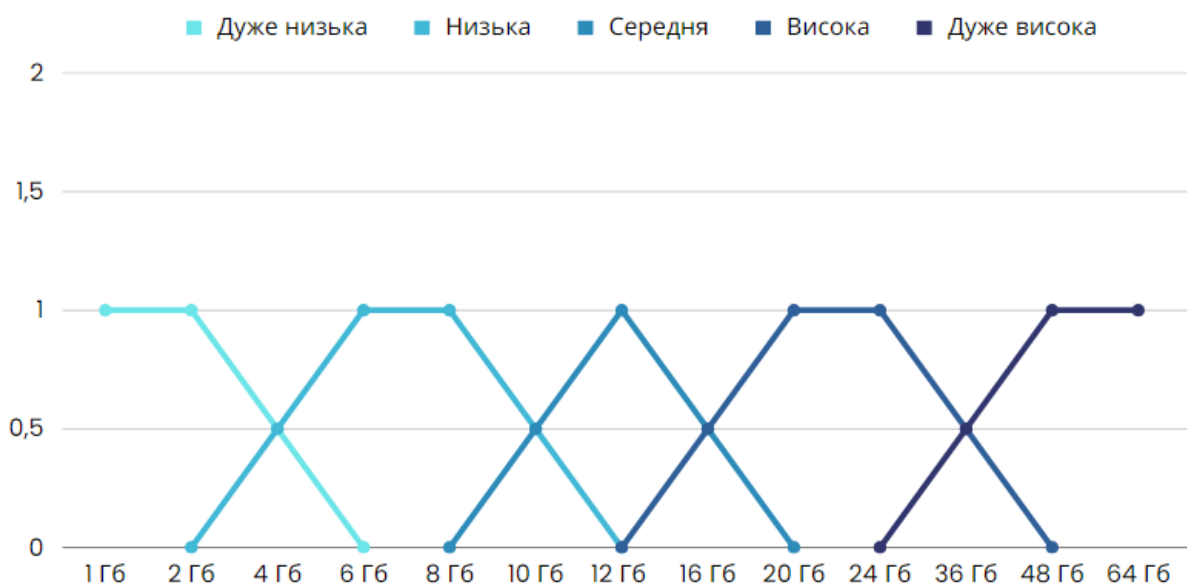


Рисунок В.4 – Графічне представлення функції належності лінгвістичної змінної «Оперативна пам'ять»

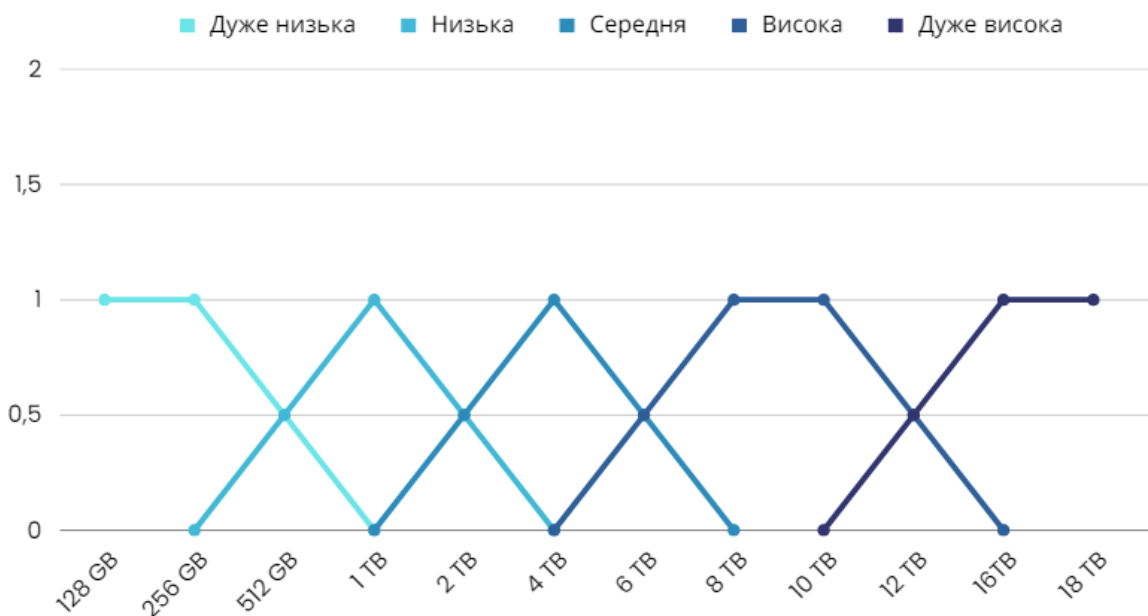


Рисунок В.5 – Графічне представлення функції належності лінгвістичної змінної «Об'єм жорсткого диска»

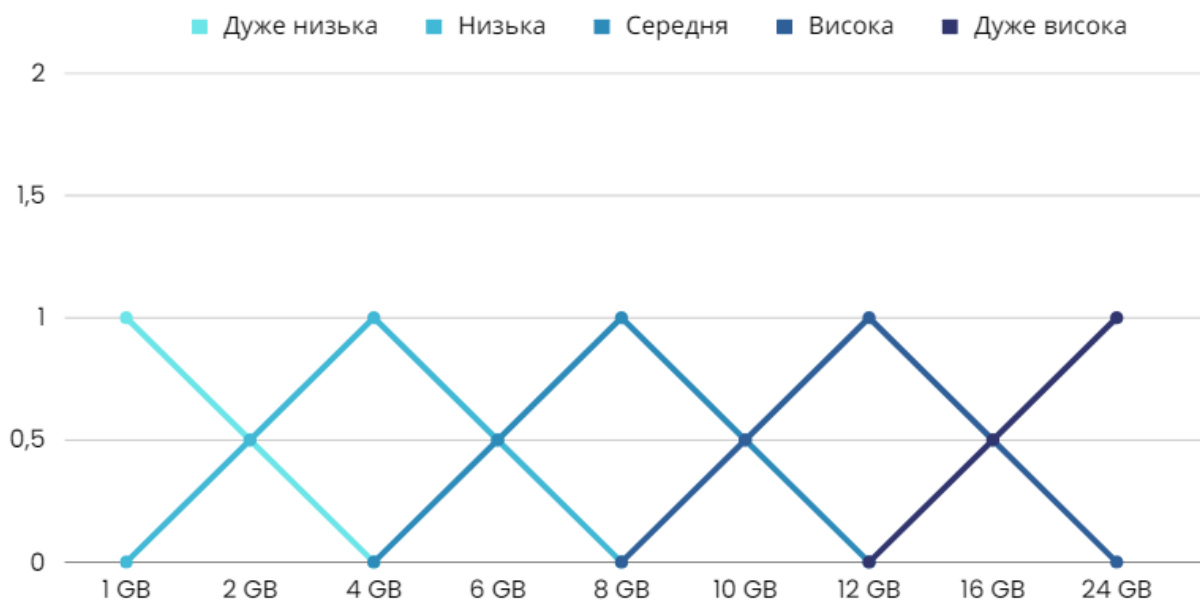


Рисунок В.6 – Графічне представлення функції належності лінгвістичної змінної «Відеокарта»

Таблиця В.1 – База знань нечіткого методу для вибору комп'ютерних комплектуючих

№	у ₁	у ₂	у ₃	у ₄	У	Z	D (збірка ПК)
1	2	3	4	5	6	7	8
1	ДВ	ДВ	ДВ	ДВ	ДВ	z ₁	Intel Core i9-12900K, Corsair Vengeance LPX 64 GB, Seagate BarraCuda Pro 4TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280
2	ДВ	В	В	ДВ	ДВ	z ₁	Intel Core i9-12900K, G.SKILL Trident Z RGB 32 GB, Western Digital WD Black 2TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280
3	В	ДВ	ДВ	В	ДВ	z ₁	AMD Ryzen 9 3950X, Corsair Vengeance LPX 64 GB, Seagate BarraCuda Pro 4TB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
4	ДВ	С	С	ДВ	В	z ₂	Intel Core i9-12900K, G.SKILL Trident Z RGB 32 GB, Western Digital WD Black 2TB, NVIDIA GeForce RTX 3090, Kingston FURY Renegade SSD 1TB M.2 2280
5	В	С	С	В	В	z ₂	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
6	В	Н	В	В	В	z ₂	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280
7	В	С	С	В	С	z ₃	AMD Ryzen 9 3950X, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Samsung 970 PRO NVMe M.2 1TB
8	В	С	С	ДВ	С	z ₃	AMD Ryzen 9 3950X, G.SKILL Trident Z RGB 32 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Samsung Portable SSD T7 500GB
9	Н	Н	Н	Н	Н	z ₄	Intel Core 2 Duo E6300 L2 SL9TA, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce GTX 1050, Apacer AS340 Panther 480GB
10	ДВ	Н	Н	ДВ	Н	z ₄	Intel Core i9-12900K, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 3090, Apacer AS340 Panther 480GB
11	Н	Н	Н	В	Н	z ₄	Intel Core i3-6100, Crucial Ballistix Sport LT 8 GB, Western Digital WD Blue 500GB, NVIDIA GeForce RTX 2080, Crucial P5 Plus NVMe 500GB M.2 2280

Продовження таблиці В.1

12	Дн	Дн	Н	Дн	Дн	z_5	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Western Digital WD Blue 500GB, NVIDIA GeForce GT 210, Apacer AS340 Panther 480GB
13	Дн	Дн	Дн	Н	Дн	z_5	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Toshiba 200GB, NVIDIA GeForce GTX 1050, Apacer, Apacer AS340 Panther 480GB
14	Дн	Дн	Дн	Дн	Дн	z_5	Intel Core i3-3220, Goodram SODIMM DDR3L-1600 4096MB, Toshiba 200GB, NVIDIA GeForce GT 210, Kesu K1 320Gb
...



Рисунок В.7 – Графічне представлення функції належності лінгвістичної змінної «Відеокарта»

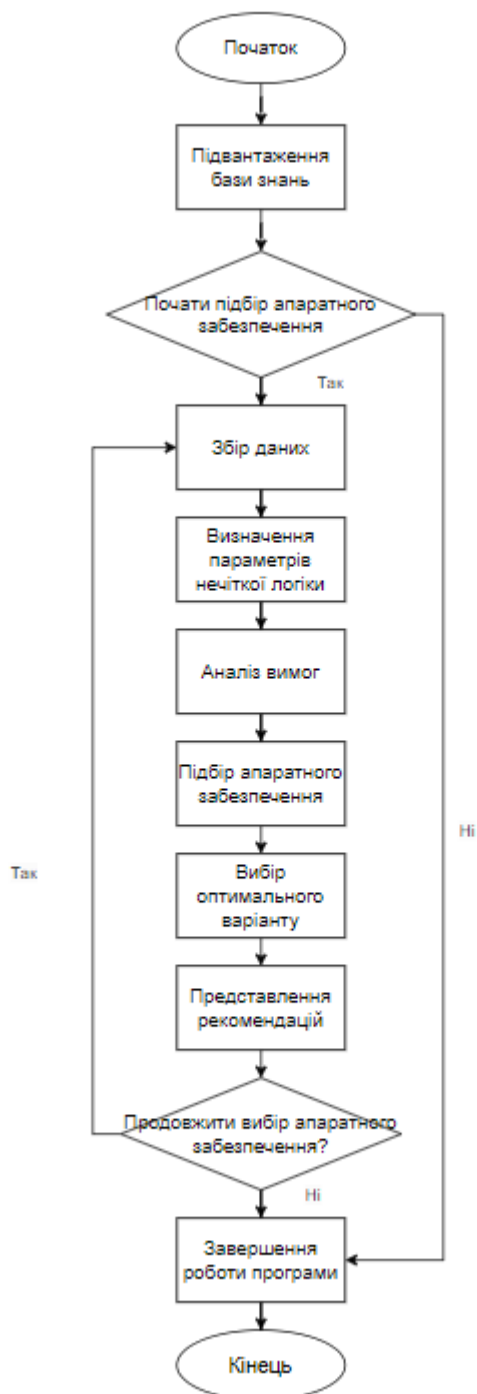


Рисунок В.8 – Схема алгоритму роботи інформаційної технології надання рекомендацій для підбору апаратного забезпечення

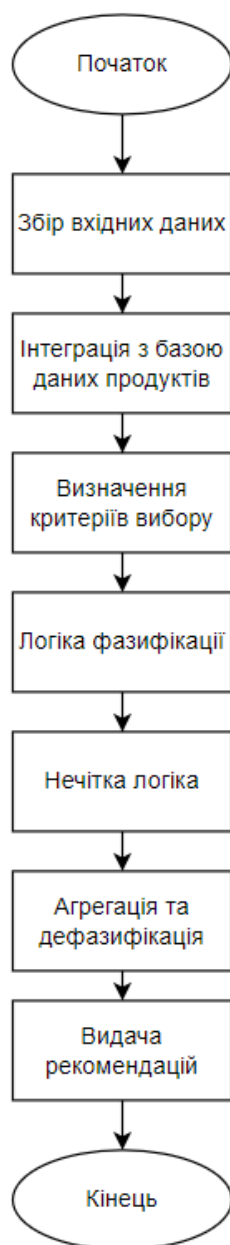


Рисунок В.9 – Схема алгоритму етапів роботи інформаційної системи

Додаток Г (довідниковий)

Інструкція користувача

Після запуску програми відкривається інтерфейсне вікно, в якому необхідно перейти до початку підбору апаратного забезпечення.

На рисунку Г.1 зображено початкову сторінку інформаційної технології, яка виконана у вигляді веб-додатку.

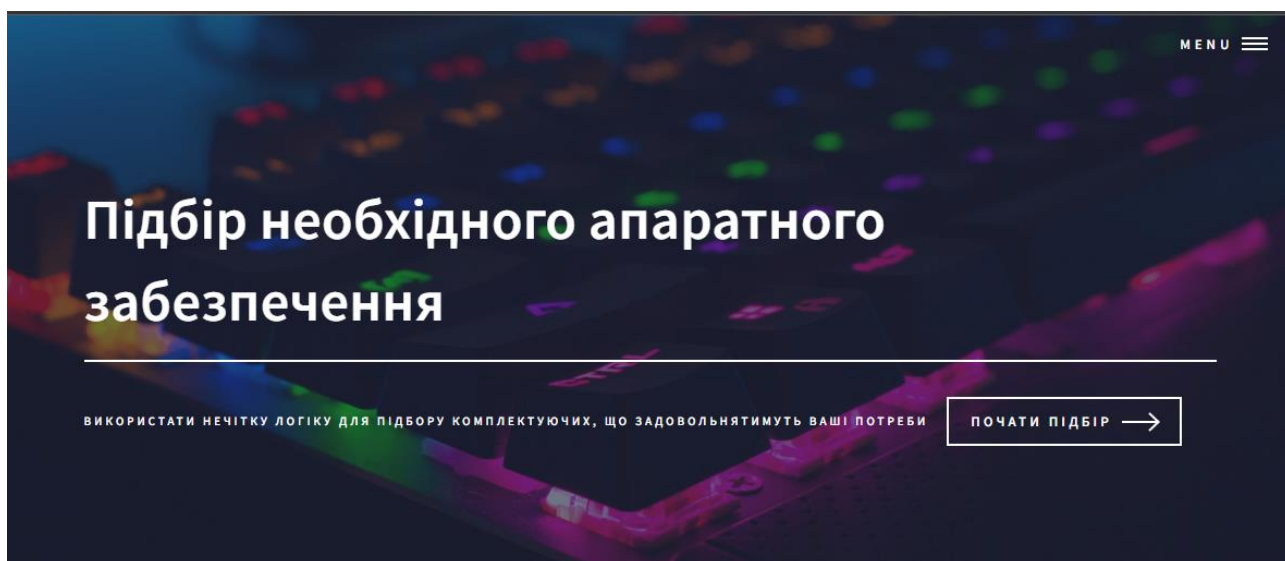


Рисунок Г.1 – Загальний вигляд інтерфейсного вікна сервісу для підбору апаратного забезпечення

На рисунку Г.2 нам буде відображено форму з комплектуючими. У залежності від потреб користувач вказує, які комплектуючі або їхні варіації він потребує.

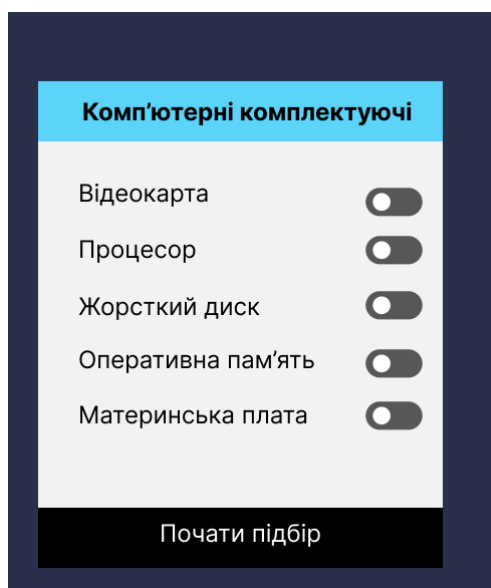


Рисунок Г.2 – Загальний вигляд інтерфейсного вікна доступних для вибору комплектуючих

На рисунку Г.3 вкажемо відеокарту та почнемо підбір.

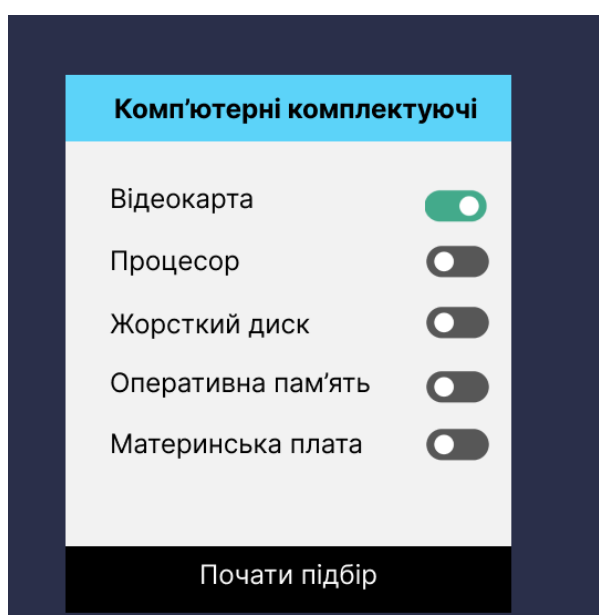


Рисунок Г.3 – Загальний вигляд інтерфейсного вікна вибору відеокарти серед комплектуючих

На рисунку 3.4 представлено форму, що відкривається за обраним варіантом і з потребами користувача, де можна задати ступінь тієї чи іншої вимоги.

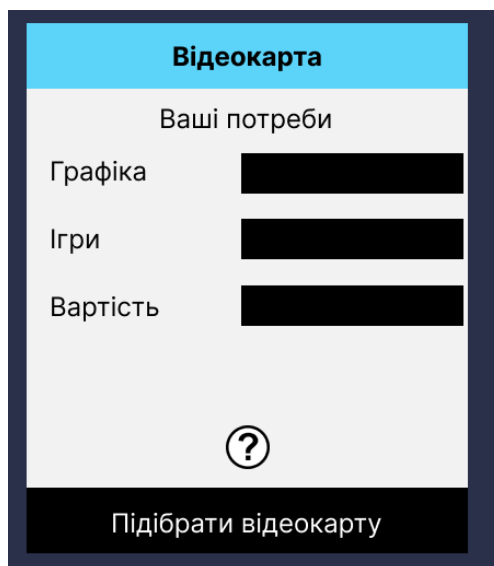


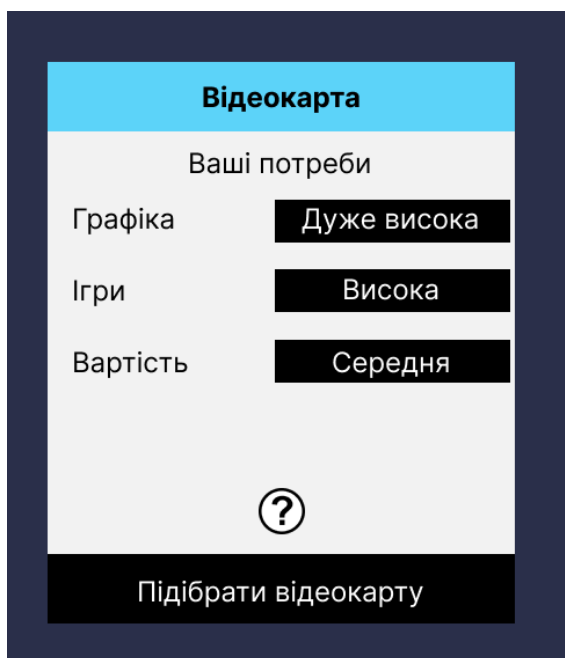
Рисунок Г.4 – Загальний вигляд інтерфейсного вікна критеріїв вибору відеокарти

Натиснувши іконку зі знаком питання просунутий користувач може переглянути інформацію з чисельними діапазонами значень, що відповідають нечітким значенням, які можна вказати в модулі підбору відеокарти. На рисунку Г.5 зображено форму, що відображає відповідні значення.

Графічні характеристики	Комп'ютерні ігри	Вартість
<ol style="list-style-type: none"> Дуже Низькі: <ul style="list-style-type: none"> VRAM: ≤ 2 GB Частота GPU: ≤ 1 GHz Спеціалізовані функції: Мінімальні Низькі: <ul style="list-style-type: none"> VRAM: 2-4 GB Частота GPU: 1-1.4 GHz Спеціалізовані функції: Підтримка базових графічних додатків Середні: <ul style="list-style-type: none"> VRAM: 4-8 GB Частота GPU: 1.4-1.8 GHz Спеціалізовані функції: Підтримка 3D-модельовання та важких графічних додатків Високі: <ul style="list-style-type: none"> VRAM: 8-16 GB Частота GPU: 1.8-2 GHz Спеціалізовані функції: Оптимізовані для високої продуктивності у професійних графічних програмах Дуже Високі: <ul style="list-style-type: none"> VRAM: > 16 GB Частота GPU: > 2 GHz Спеціалізовані функції: Висока продуктивність для складних графічних проєктів і відеопродакшену 	<ol style="list-style-type: none"> Дуже Низькі: <ul style="list-style-type: none"> VRAM: ≤ 2 GB Частота GPU: ≤ 1 GHz Підтримка DirectX: до DirectX 11 Низькі: <ul style="list-style-type: none"> VRAM: 2-4 GB Частота GPU: 1-1.4 GHz Підтримка DirectX: DirectX 11 Середні: <ul style="list-style-type: none"> VRAM: 4-6 GB Частота GPU: 1.4-1.7 GHz Підтримка DirectX: DirectX 12 Високі: <ul style="list-style-type: none"> VRAM: 6-8 GB Частота GPU: 1.7-2 GHz Підтримка DirectX: DirectX 12, підтримка ray tracing для деяких моделей Дуже Високі: <ul style="list-style-type: none"> VRAM: > 8 GB Частота GPU: > 2 GHz Підтримка DirectX: DirectX 12, підтримка ray tracing 	<ol style="list-style-type: none"> Дуже Низькі: <ul style="list-style-type: none"> Ціна: до \$100 Типові відеокарти: Основні моделі або старіші моделі. Низькі: <ul style="list-style-type: none"> Ціна: \$100 - \$200 Відеокарти для базових ігор та непрофесійної роботи з графікою. Середні: <ul style="list-style-type: none"> Ціна: \$200 - \$500 Відеокарти для загальних ігор та середнього рівня графічних завдань. Високі: <ul style="list-style-type: none"> Ціна: \$500 - \$1000 Відеокарти для вимогливих ігор та професійної роботи з графікою. Дуже Високі: <ul style="list-style-type: none"> Ціна: \$1000 та вище Відеокарти для найсучасніших ігор, складних графічних проєктів, та VR.

Рисунок Г.5 – Загальний вигляд інтерфейсного вікна діапазон значень критеріїв вибору

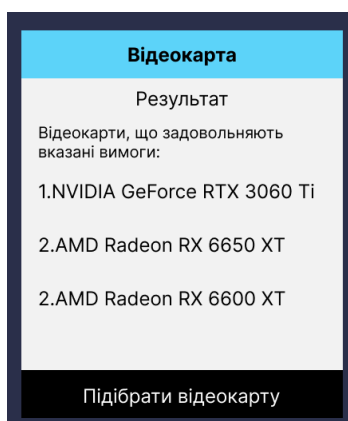
Відповідно до потреб в даному випадку користувач задає ступінь вимог до графіки, ігор та вартості. На рисунку Г.6 продемонстровано задання потреб користувача.



Відеокарта	
Ваші потреби	
Графіка	Дуже висока
Ігри	Висока
Вартість	Середня
?	
Підібрати відеокарту	

Рисунок Г.6 – Загальний вигляд інтерфейсного вікна задання критеріїв згідно вимог користувача

Задавши потреби користувачу відображається кілька результатів, що найточніше задовольняють його вимогам. На рисунку Г.7 зображено результат надання рекомендацій користувачу відповідно до заданих потреб.



Відеокарта	
Результат	
Відеокарти, що задовольняють вказані вимоги:	
1.NVIDIA GeForce RTX 3060 Ti	
2.AMD Radeon RX 6650 XT	
2.AMD Radeon RX 6600 XT	
Підібрати відеокарту	

Рисунок Г.7 – Загальний вигляд інтерфейсного вікна надання рекомендацій, що задовольняють вимогам