

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ

Зав. каф. "Комп'ютерні науки"

доц. _____ **Сергій ШАРОВ**
"17" лютого 2025 р.

Пояснювальна записка
до кваліфікаційної роботи здобувача СВО Магістр
(ступінь вищої освіти)

на тему: Інтелектуальна система для визначення конфігурації
обчислювальної системи в режимі консультування

52/4КНД.11260602.000000ПЗ

Виконав: здобувач вищої освіти 2 курсу, групи 21МБКН
спеціальності 122 Комп'ютерні науки
за ОПП Комп'ютерні науки
(шифр і назва спеціальності та ОПП)

_____ **Сергій СКРИПЧЕНКО**
(підпис) _____ (прізвище й ініціали)

Керівник, доц. _____ **Микола МІРОШНИЧЕНКО**
(підпис) _____ (прізвище й ініціали)

Нормоконтроль, ст.викл. _____ **Ольга ЗІНОВ'ЄВА**
(підпис) _____ (прізвище й ініціали)

Рецензент, доц. _____ **Степан СКРУПСЬКИЙ**
(підпис) _____ (прізвище й ініціали)

Запоріжжя – 2025 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет імені Дмитра
Моторного

Факультет: Енергетики і комп'ютерних технологій

Кафедра: Комп'ютерні науки

Ступінь вищої освіти: Магістр

Спеціальність: 122 Комп'ютерні науки

ОПП: Комп'ютерні науки _____

ЗАТВЕРДЖУЮ

Завідувач кафедри_КН

к.пед.н., доц. _____ Сергій ШАРОВ

“04” жовтня 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

_____Скрипченку Сергію Сергійовичу_____

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Інтелектуальна система для визначення конфігурації обчислювальної системи в режимі консультування»

керівник проєкту Мірошниченко Микола Юрійович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “23” вересня 2024 року № 442-С

2. Строк подання здобувачем вищої освіти роботи 14 лютого 2025

3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проєктування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Аналіз предметної області та існуючих рішень.

2. Специфікація вимог до інтелектуальної системи

3. Обґрунтування мови та середовища програмування.

4. Дослідна експлуатація інтелектуальної системи

5. Тестування інтелектуальної системи

5. Перелік графічного матеріалу: Діаграма використання, діаграми IDEF0, розкадровка інтерфейсу, структурна схема, візуальні компоненти інтерфейсу, алгоритм роботи, інтерфейс програми

Презентація – 10 слайдів

1. Титульний слайд

2. Предмет, об'єкт, мета, завдання дослідження

3. Порівняльна характеристика аналогів програмного продукту

4. Діаграма варіантів використанн

- 5. Інструментальні засоби
- 6. Інтерфейс і робота інтелектуальної системи
- 7. Інтерфейс і робота інтелектуальної системи
- 8. Тестування
- 9. Висновок
- 10. Дякую за увагу

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 01 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	
1	Аналіз предметної області	12.09.2024	
2	Специфікація вимог до комп’ютерної системи	06.10.2024	
3	Проектування інтелектуальної системи	16.10.2024	
4	Обґрунтування інструментальних засобів	05.11.2024	
5	Розробка інтелектуальної системи	10.12.2024	
6	Тестування та налагодження програмного засобу	20.01.2025	
7	Підпис керівником роботи	14.02.2025	
8	Підпис завідувачем кафедри	17.02.2025	

Здобувач вищої освіти

Сергій СКРИПЧЕНКО
(підпис) (власне ім’я та прізвище)

Керівник
кваліфікаційної
роботи

Микола МІРОШНИЧЕНКО
(підпис) (власне ім’я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота: 91 с., 27 рис., 9 табл., 29 посилань.

Актуальність: Інтелектуальні системи для визначення оптимальної конфігурації обчислювальної системи стають все більш важливими в умовах сучасного розвитку технологій. З кожним роком складність апаратного забезпечення збільшується, що призводить до розширення можливих варіантів його конфігурації. Це ускладнює процес вибору необхідних компонентів для досягнення оптимальної продуктивності, енергоефективності та відповідності бюджету. Особливо актуальними такі системи стають у зв'язку зі зростанням попиту на індивідуальні рішення, що враховують специфічні потреби користувачів, будь то геймери, професіонали у сфері графіки, або ж корпоративні клієнти, які прагнуть оптимізувати свої бізнес-процеси.

Інтелектуальна система допомагає автоматизувати процес підбору компонентів, використовуючи алгоритми машинного навчання та штучного інтелекту. Це дозволяє мінімізувати вплив людського фактора, що знижує ймовірність помилок при виборі компонентів. Завдяки такому підходу користувачі можуть швидко та ефективно отримати рекомендації щодо збірки комп'ютера, що максимально відповідає їхнім потребам та фінансовим можливостям.

Об'єкт дослідження: обчислювальні системи та їх конфігурація.

Предмет дослідження: інтелектуальна система для визначення конфігурації обчислювальної системи в режимі консультування.

Мета роботи: розробити інтелектуальну систему для визначення конфігурації обчислювальної системи.

Основні задачі:

1. Розглянути особливості та можливості інтелектуальних систем.
2. Проаналізувати наявні програмні продукти для визначення конфігурації обчислювальної системи.
3. Розглянути інструментальні засоби для розробки інтелектуальної системи.

4. Описати проектні рішення інтелектуальної системи для визначення конфігурації обчислювальної системи.
5. Створити технічне завдання на розробку інтелектуальної системи для визначення конфігурації обчислювальної системи.
6. На основі складеного технічного завдання розробити інтелектуальну систему та скласти інструкцію користувача.
7. Провести тестування розробленої інтелектуальної системи

Практичне значення дослідження: автоматизувати, оптимізувати та прискорювати процес вибору апаратного забезпечення відповідно до потреб користувача або бізнесу.

Ключові слова: ІНТЕЛЕКТУАЛЬНА СИСТЕМА, КОНФІГУРАЦІЯ ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ, КОМП'ЮТЕРНІ КОМПЛЕКТУЮЧІ, АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, ВИБІР ОБЛАДНАННЯ, РИНОК ТЕХНОЛОГІЙ.

SUMMARY

Qualification work: 91 p., 27 figures, 7 tables, 29 references.

Relevance: Intelligent systems for determining the optimal configuration of a computer system are becoming increasingly important in the context of modern technology development. Every year the complexity of hardware increases, which leads to the expansion of possible options for its configuration. This complicates the process of selecting the necessary components to achieve optimal performance, energy efficiency, and budget compliance. Such systems are becoming especially relevant due to the growing demand for customized solutions that take into account the specific needs of users, whether they are gamers, graphics professionals, or corporate clients seeking to optimize their business processes.

The intelligent system helps automate the component selection process using machine learning and artificial intelligence algorithms. This minimizes the influence of the human factor, which reduces the likelihood of errors when selecting components. Thanks to this approach, users can quickly and efficiently get recommendations on how to build a computer that best suits their needs and financial capabilities.

The object of research: the configuration of the computer system

Subject of research: Intelligent system for determining the configuration of a computer system

Purpose of work: development of an intelligent system for determining the configuration of a computer system

Main tasks:

1. To consider the features and capabilities of intelligent systems.
2. Analyze existing software products for determining the configuration of a computer system.
3. Consider tools for developing an intelligent system.
4. Describe the design decisions of an intelligent system to determine the configuration of a computer system.

5. Create a technical task for the development of an intelligent system to determine the configuration of a computer system.
6. Based on the terms of reference, develop an intelligent system and create a user manual.
7. Test the developed intelligent system.

Practical significance of the study: to automate, optimize, and speed up the process of choosing hardware according to the needs of the user or business.

Keywords: INTELLIGENT SYSTEM, COMPUTER SYSTEM CONFIGURATION, COMPUTER COMPONENTS, HARDWARE, PERSONALIZED RECOMMENDATIONS, HARDWARE SELECTION, TECHNOLOGY MARKET.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	13
1.1 Особливості та напрямки використання інтелектуальних систем	13
1.2 Аналіз існуючих рішень для підбору конфігурації обчислювальної системи.....	17
1.3. Технічне завдання на розробку програмного забезпечення для підбору конфігурації обчислювальної системи	24
Висновки до розділу 1	31
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ ..	33
2.1 Концептуальна модель використання інформаційної системи.....	33
2.2 Функціональна модель інформаційної системи.....	36
2.3 Специфікація функціональних та нефункціональних вимог	39
Висновки до розділу 2	41
РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ.....	42
ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	42
3.1 Загальний огляд технологічного стеку	42
3.2 Обґрунтування вибору мов програмування	46
3.3 Інструменти для розробки.....	48
Висновки до розділу 3	52
РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	53
4.1 Особливості програмної реалізації інтелектуальної системи для визначення конфігурації обчислювальної системи.....	53
4.2 Функціональні можливості та інструкція користувача інформаційної системи	59

4.2.1 Покрокова інструкція користування.....	62
4.2.2 Адміністрування	63
4.3 Тестування інтелектуальної системи	65
4.3.1 Методика тестування.....	66
4.3.2 Результати тестування.....	67
Висновки до розділу 4	75
ВИСНОВКИ.....	77
СПИСОК ЛІТЕРАТУРИ.....	78
ДОДАТОК А.....	81
ДОДАТОК Б	85
ДОДАТОК В.....	88
ДОДАТОК Г	91

ВСТУП

Актуальність. На сьогоднішній день ІС активно використовуються в різних областях діяльності людини, підтверджуючи свою універсальність і значний потенціал. Наприклад, вони допомагають вирішити завдання моніторингу, прогнозування та управління, наприклад, транспортними потоками, забезпечити навчання здобувачів освіти та дорослих з урахуванням їхніх особливостей тощо. Серед основних напрямів застосування інтелектуальних систем можна виділити такі: обробка природної мови (використання чат-ботів, голосових помічників, автоматизованих перекладачів, що використовуються для автоматизації спілкування з клієнтами, в адаптивних навчальних платформах, для аналізу великих текстових масивів); підтримка прийняття рішень (аналіз даних, оцінка можливих сценаріїв розвитку подій, генерація оптимальних рішень); аналіз візуальної інформації (розпізнавання та аналіз зображень, що застосовується у сферах безпеки, медицини, автоматизації виробництва тощо); аналіз великих обсягів даних та формування на цій основі різноманітних прогнозів; робототехніка та автоматизація різних процесів на виробництві, побуті та ін.

У зв'язку з необхідністю автоматизації діяльності консультування щодо конфігурації обчислювальної системи, тему дослідження було сформульовано як **«Розробка інтелектуальної системи для визначення конфігурації обчислювальної системи в режимі консультування»**.

Об'єкт: використання прикладного програмного забезпечення з елементами штучного інтелекту

Предмет: інтелектуальна системи для визначення конфігурації обчислювальної системи.

Мета: розробити інтелектуальну систему для визначення конфігурації обчислювальної системи.

Сформована мета, об'єкт і предмет дослідження дозволили поставити такі **завдання:**

1. Розглянути особливості та можливості інтелектуальних систем.
2. Проаналізувати наявні програмні продукти для визначення конфігурації обчислювальної системи.
3. Розглянути інструментальні засоби для розробки інтелектуальної системи.
4. Описати проектні рішення інтелектуальної системи для визначення конфігурації обчислювальної системи.
5. Створити технічне завдання на розробку інтелектуальної системи для визначення конфігурації обчислювальної системи.
6. На основі складеного технічного завдання розробити інтелектуальну систему та скласти інструкцію користувача.
7. Провести тестування розробленої інтелектуальної системи

При реалізації поставлених задач було використано наступні **методи**: теоретичні: аналіз та синтез наукової та спеціальної літератури з теорії роботи зі штучним інтелектом, програмування; емпіричні: спостереження та бесіда з метою визначення специфіки роботи консультанта; практичні: програмна реалізація інтелектуальної системи.

Практичне значення дослідження полягає в розробці інформаційної системи, яка дозволить автоматизувати окремі функції консультанта, а саме надання рекомендацій щодо конфігурації обчислювальної системи.

Апробація результатів здійснювалася шляхом опублікування статті «Особливості та напрямки використання інтелектуальних систем» у збірнику матеріалів IV Всеукраїнської науково-практичної інтернет-конференції «Сучасні комп'ютерні та інформаційні системи і технології» (м. Запоріжжя, 2024).

Структура і обсяг роботи. Дипломна робота складається з вступу, чотирьох розділів, загальних висновків, списку використаних джерел (29 найменувань). Перший розділ присвячений аналізу специфіки роботи з інтелектуальними системами, аналізу аналогічного програмного забезпечення та постановці технічного завдання. Другий розділ присвячений створенню концептуальної та функціональної моделі а також складанню специфікації

функціональних та нефункціональних вимог. Третій розділ присвячено, аналізу інструментальних засобів розробки інтелектуальної системи, обґрунтуванню вибору мов програмування та інструментів розробки. Четвертий розділ присвячено особливостям програмної реалізації інтелектуальної системи, функціональним можливостям, інструкції користувача та тестуванню інтелектуальної системи. Загальний обсяг роботи складає 91 сторінки.

РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Особливості та напрямки використання інтелектуальних систем

Глибинні процеси інформатизації суспільства призвели до широкого використання різноманітного програмного забезпечення для реалізації безлічі прикладних задач. Одним з перспективних типів програмного забезпечення є інтелектуальні системи (ІС), в основу яких покладено функціонування елементів штучного інтелекту (ШІ). ІС набули значної популярності, оскільки забезпечують вирішення складних неформалізованих задач, які раніше традиційним програмним забезпеченням вирішити було неможливо.

Зростання продуктивності обчислювальних пристроїв та розробка сучасних алгоритмів обробки інформації дало змогу вирішувати широкий спектр завдань, пов'язаних із цифровою обробкою даних. Однак, попри високий рівень технологічного прогресу, на сьогодні існує багато задач, для яких створення ефективних алгоритмів залишається проблематичним [11, с. 145] та які складно вирішити за допомогою стандартного програмного забезпечення.

У цьому контексті інтелектуальні системи (ІС) мають важливе практичне значення, адже вони розроблені для роботи із слабоформалізованими знаннями та складними обчислювальними процесами. Інтелектуальні системи належать до перспективних напрямів розвитку сучасних інформаційних технологій. Вони ефективно використовуються у вирішенні завдань, що потребують аналізу великих обсягів даних, ухвалення обґрунтованих рішень на основі різних параметрів, дозволяють генерувати зображення, звук, інші види інформації.

Інтелектуальну систему можна визначити як програмну систему, що здатна розв'язувати завдання у конкретній предметній галузі, що традиційно

пов'язані з творчістю. Такі системи здатні імітувати людські здібності за рахунок обробки інформації та використання знань для досягнення поставлених цілей [9, с. 150]. Функціонування ІС засновано на обробці бази знань або моделі предметної області, яка описується мовою високого рівня, максимально наближеною до природної мови людини [6, с. 47].

Коротко окреслимо основні характеристик та переваги інтелектуальних систем. Однією з важливих особливостей ІС є їх здатність до генерації нових знань шляхом навчання і самонавчання, здатність до зміни не лише параметрів функціонування, але й способів своєї поведінки залежно від поточного стану інформаційних входів. Далі, важливою характеристикою інтелектуальних систем є їхня адаптивність, тобто здатність зберігати свою функціональність у разі непередбачених змін властивостей об'єкта управління, навколишнього середовища [13, с. 85] та інших параметрів. Це забезпечує високу адаптивність системи до нових умов і задач, а також дозволяє автоматизувати процеси ухвалення управлінських рішень на якісно новому рівні [19, с. 137]. Інтелектуальні системи здатні до обробки інформації, яка не завжди має формалізований вигляд. Прикладом може слугувати звичайна мова людини, яку складно представити у вигляді формальних алгоритмів. Інтелектуальні системи більше спрямовані на обробку змістової складової даних, ніж на їхньому механічному обчисленні. Завдяки здатності поєднанню значних обчислювальних можливостей комп'ютерної техніки з когнітивними механізмами ІС, такі системи можуть вирішувати проблеми, що потребують як точного аналізу, так і творчого підходу.

Існує декілька видів інтелектуальних систем. Наприклад, за функціональними особливостями виділяють такі ІС:

- інтелектуальні інформаційні системи призначені для збору, зберігання, аналізу та інтеграції даних, найчастіше використовуються для прийняття обґрунтованих рішень на основі великих масивів інформації;

- експертні системи – ІС, що здатні автоматизувати процеси ухвалення складних рішень на основі знань, отриманих від експертів, тому мають практичне значення у сфері консультування та діагностики;
- розрахунково-логічні системи широко застосовуються у технічних і наукових дослідженнях за рахунок використання точних обчислень, математичного моделювання та аналітичного аналізу;
- гібридні інтелектуальні системи поєднують функціональні можливості декількох видів систем, що робить їх більш універсальними з точки зору виконання широкого спектру завдань;
- рефлекторні інтелектуальні системи здатні моделювати поведінкові реакції людини або інших живих організмів за рахунок здатності реагувати на зовнішні стимули в режимі реального часу [9, с. 150].

На сьогоднішній день ІС активно використовуються в різних областях діяльності людини, підтверджуючи свою універсальність і значний потенціал. Наприклад, вони допомагають вирішити завдання моніторингу, прогнозування та управління, наприклад, транспортними потоками [14, с. 7], забезпечити навчання здобувачів освіти та дорослих з урахуванням їхніх особливостей тощо [17, с. 244]. Серед основних напрямів застосування інтелектуальних систем можна виділити такі: обробка природної мови (використання чат-ботів, голосових помічників, автоматизованих перекладачів, що використовуються для автоматизації спілкування з клієнтами, в адаптивних навчальних платформах, для аналізу великих текстових масивів); підтримка прийняття рішень (аналіз даних, оцінка можливих сценаріїв розвитку подій, генерація оптимальних рішень); аналіз візуальної інформації (розпізнавання та аналіз зображень, що застосовується у сферах безпеки, медицини, автоматизації виробництва тощо); аналіз великих обсягів даних та формування на цій основі різноманітних прогнозів; робототехніка та автоматизація різних процесів на виробництві, побуті та ін.

З іншого боку, штучний інтелект, який лежить в основі інтелектуальних систем, має певні обмеження, що пояснюється їх сутністю [3, с. 198].

Наприклад, має обмежені здатності до створення нових ідей. ІС переважно функціонують на основі попередньо закладених алгоритмів і моделей, можуть відтворювати та комбінувати відомі знання, але не створювати абсолютно нові. Але вони не здатні генерувати інноваційні концепції без втручання людини. Це робить їх ефективними лише в межах заданих вихідних даних у певній наочній області. Також до слабких сторін ІС та штучного інтелекту загалом слід віднести відсутність інтуїції, що іноді є важливим для ухвалення нестандартних рішень. Крім того, досвід використання генеративних системи ШІ показав, що для отримання якісного результату потрібно чітко сформулювати проблему. В протилежному випадку людина не отримує ту відповідь, на яку розраховувала.

Стосовно перспектив використання ІС хочемо зазначити, що зараз спостерігається прискорення темпів впровадження інтелектуальних систем у вирішення транспортних і виробничих проблем і завдань [2, с. 12], юридичну справу, медицину, безпеку тощо. Вдосконалення ІС дозволить забезпечити більшу точності у прийнятті рішень, розширити сфери їх застосування, звільнити людину від виконання рутинних видів робіт та спрямує її активність на творчу та стратегічну діяльність.

Отже, інтелектуальні системи є вагомим інструментом людської діяльності, що використовують передові технології з обробки інформації та дозволяють вирішувати задачі, які традиційно потребують значних інтелектуальних зусиль з боку людини. Завдяки своїй здатності до самоорганізації і навчання, інтелектуальні системи забезпечують високий рівень автоматизації і підтримки прийняття рішень в реальному часі. Не зважаючи на слабкі сторони, ІС дозволять сформувати нові підходи у бізнесі, управлінні та науці, досягти позитивних змін в різних аспектах діяльності людини.

1.2 Аналіз існуючих рішень для підбору конфігурації обчислювальної системи

У сучасному світі інформаційних технологій комп'ютери є невід'ємною частиною життя як у професійній, так і в побутовій сферах. Зростаючі вимоги до продуктивності та специфікацій комп'ютерних систем обумовлюють необхідність ретельного підбору комплектуючих для ПК. Вибір правильних компонентів залежить не лише від технічних параметрів, але й від індивідуальних потреб користувача, таких як ігровий досвід, професійна робота з графікою чи програмування, або ж звичайне домашнє використання. Однак процес вибору комплектуючих ускладнюється через різноманіття доступних рішень на ринку. Неправильний підбір компонентів може призвести до несумісності, неефективного використання бюджету або навіть до зниження продуктивності системи. Саме тому аналіз ринку, включно з вивченням доступних інструментів та сервісів для підбору комплектуючих, є критично важливим.

Проаналізуємо ринок інструментів для підбору комплектуючих для створення ПК, а також дослідимо наявні сервіси та їхні особливості, що можуть допомогти в цьому процесі.

Онлайн-сервіс PCPartPicker – це один із найбільш відомих інструментів для складання конфігурацій ПК. Його популярність обумовлена великою базою даних комплектуючих, яка охоплює широкий спектр брендів і моделей від бюджетного до преміум-класу. Сервіс забезпечує автоматичну перевірку сумісності компонентів, що дозволяє уникнути типових помилок, наприклад, несумісності материнської плати з процесором або оперативною пам'яттю (рис. 1.1).

Окрім цього, PCPartPicker дозволяє користувачам оцінювати вартість створеної конфігурації та розраховувати енергоспоживання системи, що є важливим для правильного вибору блоку живлення. Інтерфейс платформи інтуїтивно зрозумілий, що робить її доступною як для новачків, так і для

досвідчених користувачів.

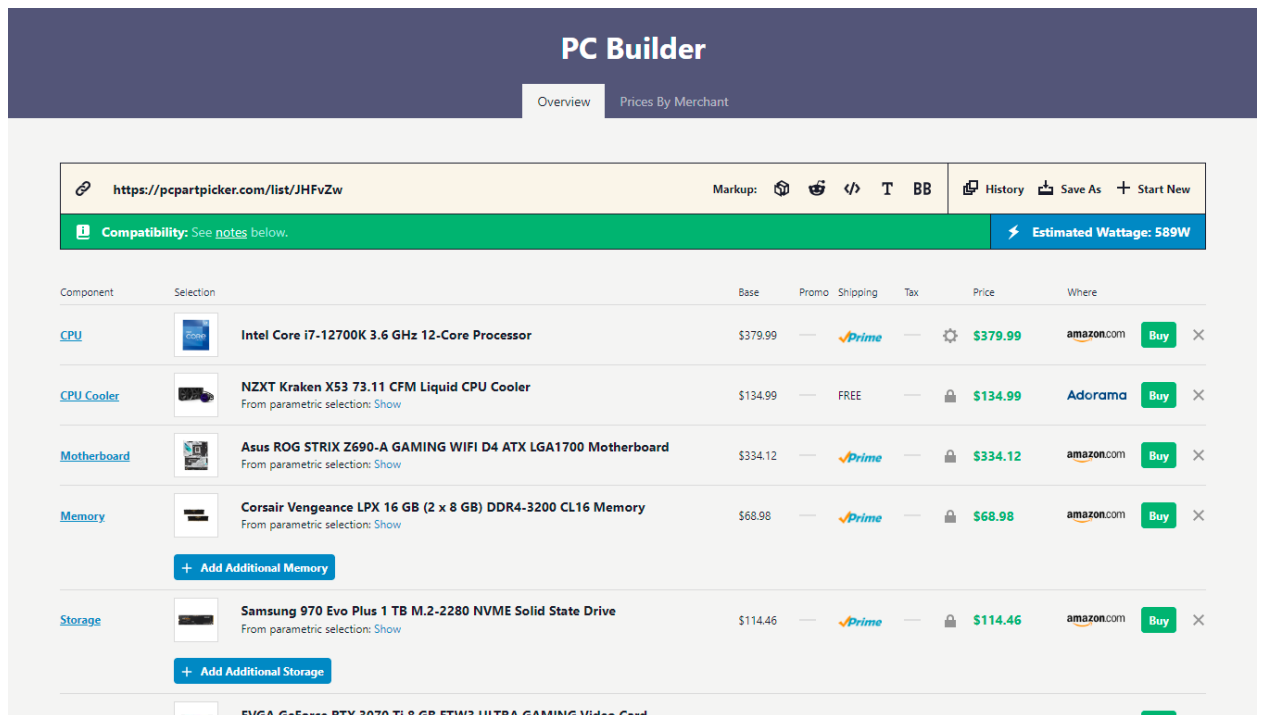


Рисунок 1.1 – Вікно сервісу PCPartPicker

Проте сервіс має свої обмеження. Він не підтримує адаптації під конкретні сценарії використання, наприклад, оптимізації для ігор, офісної роботи чи професійної обробки відео. Штучний інтелект у платформі не використовується, тому рекомендації базуються виключно на технічних параметрах і сумісності, без врахування індивідуальних потреб користувачів. Також платформа не має підтримки української мови, що може створити труднощі для україномовних користувачів, які не володіють англійською мовою [28].

Веб-додаток BuildMyPC – це інструмент, розроблений для користувачів, які шукають простий спосіб підібрати комплектуючі для ПК у межах заданого бюджету. Його головна функція полягає в тому, щоб знайти оптимальні компоненти, які відповідають фінансовим можливостям користувача. Сервіс також перевіряє сумісність обраних комплектуючих, що забезпечує їх коректну роботу в складі майбутньої системи (рис. 1.2).

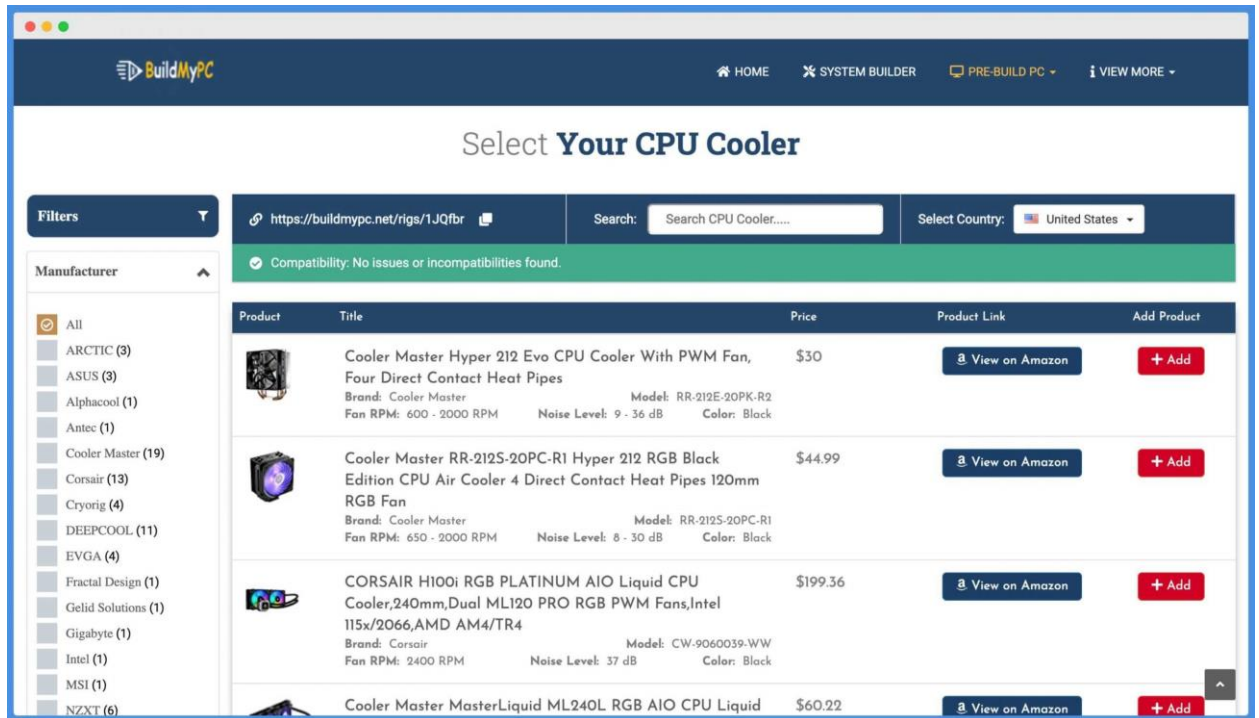


Рисунок 1.2 – Вікно сервісу BuildMyPC

Важливою перевагою BuildMyPC є його простота у використанні. Завдяки мінімалістичному дизайну інтерфейсу навіть користувачі без технічного досвіду можуть швидко скласти конфігурацію. Сервіс добре підходить для тих, хто має обмежений бюджет і шукає максимально вигідні варіанти.

Недоліком платформи є обмежена база даних компонентів, через що користувачі можуть не знайти деякі специфічні моделі чи новинки ринку. Крім того, сервіс не адаптований для підбору компонентів під конкретні завдання, такі як відеомонтаж або ігри. Платформа не використовує штучний інтелект, тому її функціонал обмежується лише базовими рекомендаціями. Українська мова також відсутня, що звужує її доступність для україномовної аудиторії [23].

Logical Increments – це унікальний сервіс, який пропонує користувачам чітко структуровану систему підбору комплектуючих на основі рівнів продуктивності. Він не здійснює автоматичної перевірки сумісності, але надає

рекомендації щодо компонентів залежно від доступного бюджету. Сервіс регулярно оновлює свою базу, щоб відповідати поточним тенденціям на ринку (рис. 1.3).

	Graphics	CPU	CPU Cooler	Motherboard	RAM	HDD	SSD	Power Supply	Case	Total
Destitute	UHD 610 \$0	G5905 \$40	Stock \$0	ASRock B560M-HDV \$104	4GB DDR4 \$14	None \$0	256GB \$26	Tt 500W (W) \$40	Focus G \$55	\$279
		G5900 \$48	212 Black \$30	ASUS H510M-E \$136	8GB DDR4 \$21	2TB \$52	512GB \$48	SeaSonic 500W (B) \$65	MX330X \$64	
expand to see details										
Poor	Vega 8 \$0	R3 3200G \$80	Stock \$0	ASUS B450M-A \$70	4GB DDR4 \$14	None \$0	256GB \$26	Tt 500W (W) \$40	Focus G \$55	\$285
			AK400 \$35	ASRock B450M-HDV \$73	8GB DDR4 \$21	2TB \$52	512GB \$48	SeaSonic 500W (B) \$65	MX330X \$64	
expand to see details										
Minimum	Vega 7 \$0	R5 5600G \$137	Stock \$0	ASUS B450M-A \$70	8GB DDR4 \$21	2TB \$65	512GB \$48	Tt 500W (W) \$40	Focus G \$55	\$436
			212 Black \$30	ASRock B450M-HDV \$73			1TB \$70	SeaSonic 500W (B) \$65	MX330X \$64	
expand to see details										
Entry	▲ RX 550 \$83	i3 12100F \$85	Stock \$0	ASUS H610M-A \$110	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	Focus G \$55	\$532
			Pure Rock 2 \$38				1TB \$70	SeaSonic 550W (B) \$70	Archon 2 \$60	
expand to see details										
Modest	RX 6400 \$120	i3 12100F \$85	Stock \$0	ASUS H610M-A \$110	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	Focus G \$55	\$569
	RX 6500XT \$140		Pure Rock 2 \$38		16GB DDR4 \$31		1TB \$70	SeaSonic 550W (B) \$70	Archon 2 \$60	
expand to see details										
Fair	Arc A580 \$180	i3 12100F \$85	Pure Rock 2 \$38	ASUS H610M-A \$110	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	P7 Neo \$75	\$687
	RX 6600 \$200		Peerless \$40	ASRock B660 Steel \$160	16GB DDR4 \$31		1TB \$70	SeaSonic 550W (B) \$70	Focus 2 \$80	
expand to see details										
Good	Arc A750 \$190	▲ i5 12400F \$111	Pure Rock 2 \$38	ASRock Z690 Phantom \$145	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	Pure Base 500 \$80	\$763
	RX 6600 \$200		Peerless \$40	ASRock B660 Steel \$160	16GB DDR4 \$31	4TB \$85	1TB \$70	SeaSonic 550W (B) \$70	NR600 \$90	
expand to see details										
Very Good	RX 6650XT \$235	i5 12400F \$160	Pure Rock 2 \$38	ASRock B760M Riptide \$120	8GB DDR5 \$23	2TB \$65	512GB \$48	SeaSonic 650W (B) \$70	Pure Base 500 \$80	\$839
	RX 7600 \$265	i5 12600K \$169	Peerless \$40	Gigabyte B760M DS3H \$140	16GB DDR5 \$47	4TB \$85	1TB \$70	EVGA 650W (G) \$87	DF600 Flux \$80	

Рисунок 1.3 – Вікно сервісу Logical Increments

Однією з головних переваг платформи є її простота. Завдяки системі рівнів продуктивності користувач може легко знайти конфігурацію, що відповідає його фінансовим можливостям і вимогам. Сервіс підходить для новачків, які не знайомі з технічними деталями, але хочуть отримати базові рекомендації. Недоліком є відсутність функції автоматичної перевірки сумісності, що може призвести до помилок при складанні системи. Також сервіс не використовує штучний інтелект, тому рекомендації залишаються загальними і не враховують індивідуальні потреби користувачів. Платформа недоступна українською мовою, що може ускладнювати її використання для україномовних користувачів [26].

Newegg PC Builder – це сервіс, інтегрований із платформою великого

онлайн-магазину Newegg, що дозволяє користувачам підбирати комплектуючі з перевіркою сумісності та замовляти їх безпосередньо через сайт. Окрім підбору, платформа пропонує готові шаблони конфігурацій для різних сценаріїв використання, таких як ігри, офісна робота або мультимедійні завдання (рис. 1.4).

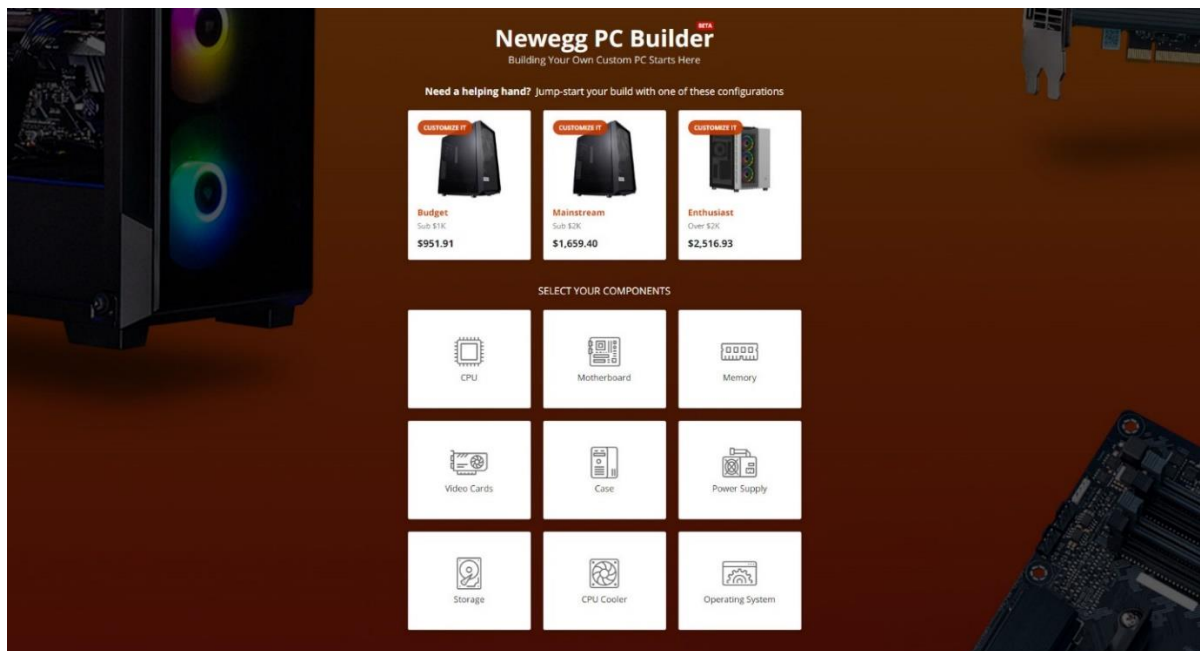


Рисунок 1.4 – Вікно сервісу Newegg PC Builder

Перевагою сервісу є його інтеграція з магазином, що дозволяє легко перейти від створення конфігурації до купівлі комплектуючих. Це зручно для користувачів, які хочуть швидко отримати готовий до зборки комплект. Сервіс також підтримує базові алгоритми рекомендацій, що частково використовують принципи штучного інтелекту.

До недоліків можна віднести обмеженість вибору лише тими товарами, які доступні в магазині Newegg, а також базовість алгоритмів рекомендацій. Платформа орієнтована переважно на американський ринок, а українська мова в ній відсутня, що ускладнює використання сервісу для україномовних користувачів [24].

Micro Center Custom PC Builder – це інструмент для складання ПК,

орієнтований на локальні магазини Micro Center. Він надає користувачам можливість створювати персоналізовані конфігурації з перевіркою сумісності комплектуючих. Інтеграція з фізичними магазинами дозволяє користувачам не лише отримати інформацію про наявність і ціни компонентів у реальному часі, а й отримати професійну консультацію на місці (рис. 1.5).

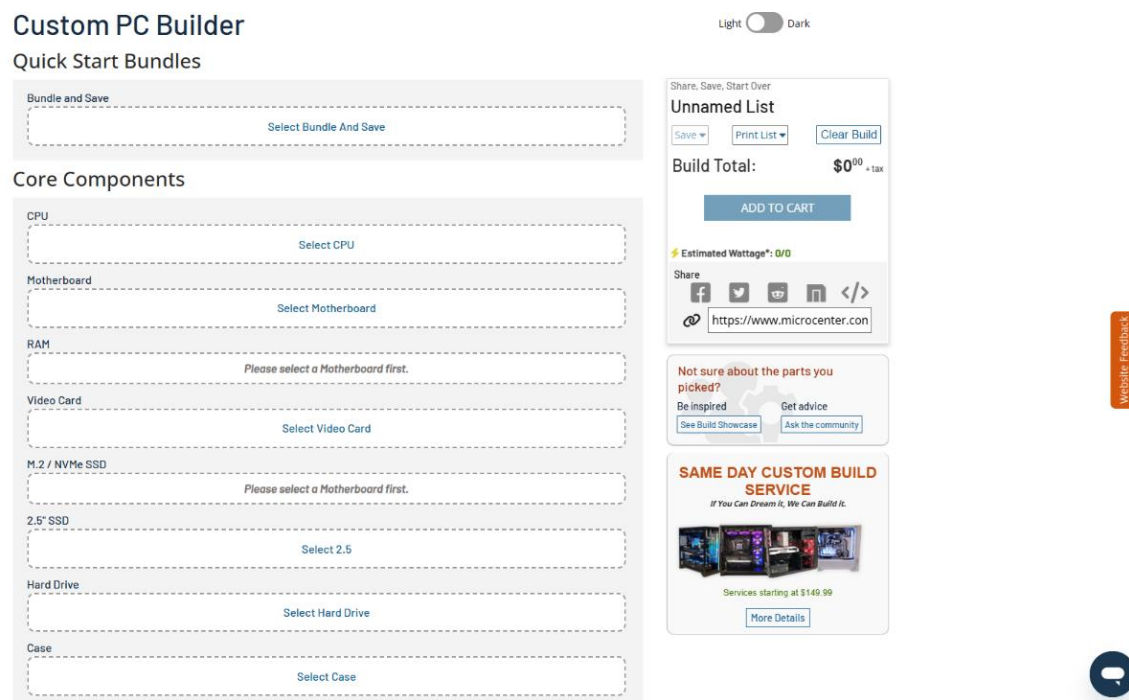


Рисунок 1.5 – Вікно сервісу Micro Center Custom PC Builder

Однією з ключових переваг платформи є її орієнтація на різні сценарії використання, такі як ігрові системи, офісні ПК чи конфігурації для роботи з графікою. Це робить сервіс гнучким і зручним для широкої аудиторії. Інструмент також забезпечує високу актуальність цін і наявності товарів.

Головним обмеженням є географічна локалізація, оскільки сервіс доступний лише в регіонах, де є магазини Micro Center. Крім того, платформа не використовує штучний інтелект для автоматичного аналізу потреб користувача. Як і інші розглянуті сервіси, Micro Center Custom PC Builder не підтримує української мови, що знижує його доступність для україномовних користувачів [27].

Таблиця 1.1 – Порівняння сервісів

Назва сервісу	Назва критерію						
	Функціональні можливості	База даних	Використання ІІ	Орієнтація	Адаптація до задач	Зручність використання	Українська локалізація
PCPartPicker	Перевірка сумісності та ціни	Вел.	Ні	Загал.	Част.	Вис.	Ні
BuildMyPC	Підбір по бюджету	Сер.	Ні	Локал.	Ні	Вис.	Ні
Logical Increments	Підбір за рівнем продуктивності	Сер.	Ні	Загал.	Част.	Сер.	Ні
Newegg PC Builder	Перевірка сумісності, замовлення	Сер.	Част.	Локал.	Част.	Вис.	Ні
Micro Center Custom PC Builder	Персональні конфігурації	Вел.	Ні	Локал.	Вис.	Вис.	Ні

Отже, аналіз існуючих рішень для підбору комплектуючих для ПК свідчить про різноманітність підходів та інструментів, які допомагають користувачам створювати індивідуальні конфігурації (табл 1.1). Кожен із розглянутих сервісів має свої сильні сторони та обмеження, що робить їх придатними для різних категорій користувачів. Таким чином, аналіз показує, що сучасні сервіси є ефективними для базових сценаріїв вибору компонентів, але мають простір для вдосконалення, особливо у напрямках персоналізації та використання штучного інтелекту. Це створює можливості для розробки

інноваційних рішень, які забезпечать більш гнучкий і точний підбір комплектуючих для різноманітних потреб користувачів.

Цей аналіз показує, що кожен із сервісів має як сильні, так і слабкі сторони, але всі вони обмежені у підтримці української мови, що може бути вирішальним фактором для деяких користувачів, особливо у наш час.

1.3. Технічне завдання на розробку програмного забезпечення для підбору конфігурації обчислювальної системи

Будь-яка програмна розробка (довідково-інформаційна система, програмно-педагогічний засіб, електронний ресурс тощо), на початковому етапі повинна передбачати створення технічного завдання (ТЗ). Цей документ регламентує функціональність програмного засобу та визначає вимоги до процесу впровадження та звітної документації. При цьому виконавець робіт повинен зрозуміти сутність завдання, показати замовнику приблизний інтерфейс майбутнього програмного продукту, а замовнику зрозуміти, підтвердити та розписатися в тому, що саме йому потрібно у цьому програмному засобі.

При написанні ТЗ ми орієнтувалися на ГОСТ 34.602-89. Цей стандарт розповсюджується на автоматизовані системи для автоматизації різних видів діяльності (управління, проектування, дослідження тощо), включаючи їх поєднання, і встановлює склад, зміст, правила оформлення документа «Технічне завдання на створення (розвиток або модернізацію) системи».

Зміст

1. Вступ

1.1 Загальні визначення

1.2 Призначення документу

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

- 2.2 Характеристика об'єктів автоматизації
- 3. Вимоги до програмного засобу
 - 3.1 Вимоги до оформлення інтерфейсу
 - 3.2 Вимоги до обробки даних і мов програмування
 - 3.3 Вимоги до програмного та технічного забезпечення
- 4. Стадії і етапи розробки програмного засобу
 - 4.1 Склад і зміст робіт із розробки програмного засобу
- 5. Порядок контролю і приймання
 - 5.1 Перевірка працездатності роботи програми
 - 5.2. Вимоги до введення програмного засобу в дію

1. Вступ

1.1 Загальні визначення

Повна назва програмного продукту: програмний засіб для автоматизації роботи головного менеджера супермаркету.

Замовник – організація, яке замовляє створення програмного засобу.

Виконавець – програміст, який розробляє програмний засіб та встановлює його на комп'ютер Замовника, в даному випадку це студентка 4 курсу очного відділення Скрипченко С.С.

C#, Microsoft Visual Studio – мова програмування та середовище візуальної розробки додатків, що дозволяють розробити програмний засіб за невеликий період часу.

Python, IDLE - це потужна і проста у вивченні мова програмування та її вбудована IDE.

Vertex AI – це платформа машинного навчання (ML) від Google Cloud, яка об'єднує багато попередніх сервісів Google Cloud ML в єдине, інтегроване середовище. Вона створена для того, щоб допомогти командам з обробки даних та ML-інженерам створювати, розгортати та масштабувати моделі машинного навчання швидше та ефективніше.

1.2 Призначення документу

У цьому документі приводиться повний перелік вимог до програмної

реалізації та впровадження програмного засобу. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

1. Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.
2. Замовник згоден зі всіма положеннями цього Технічного Завдання.
3. Замовник не має права вимагати від Виконавця в рамках поточного Договору виконання робіт або надання послуг, прямо не описаних у справжньому Технічному Завданні.
4. Виконавець зобов'язується виконати роботи в об'ємі, вказаному в справжньому Технічному Завданні.
5. Замовник не має права вимагати від Виконавця дотримання яких-небудь форматів і стандартів, якщо це не вказано в справжньому Технічному Завданні.
6. Все неоднозначності, виявлені в справжньому Технічному завданні після його підписання, підлягають двосторонньому узгодженню між Сторонами. В процесі узгодження можуть бути розроблені додаткові вимоги, які оформляються додатковою угодою до Договору і відповідним чином оцінюються.

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

Предметом розробки є програмний засіб для автоматизації роботи менеджера у сфері обслуговування покупців комп'ютерної техніки.

Програмний засіб призначений вибору комп'ютера, допомагаючи користувачам визначитися з оптимальною конфігурацією, враховуючи їхні потреби та бюджет. Програмний засіб має бути корисним для користувачів, які не мають глибоких знань у галузі комп'ютерної техніки.

2.2 Характеристика об'єктів автоматизації

Автоматизація процесу підбору комп'ютерної техніки в магазинах, що займаються продажем ПК та комплектуючих, є важливим кроком для підвищення якості обслуговування клієнтів і оптимізації роботи персоналу. В

умовах високої конкуренції на ринку та стрімкого розвитку технологій автоматизовані системи стають ключовими інструментами для вирішення основних викликів, пов'язаних із вибором оптимальних конфігурацій комп'ютерів.

Першим і головним об'єктом автоматизації є процес консультування клієнтів. Багато покупців стикаються з проблемами через недостатню обізнаність у характеристиках комплектуючих та складності їхньої сумісності. Автоматизована система виконує функцію консультанта, аналізуючи потреби клієнтів, такі як призначення ПК (ігровий, офісний, для графічного дизайну тощо), бажаний рівень продуктивності, бюджет та інші специфічні параметри. Це дозволяє надавати індивідуалізовані рекомендації, зводячи до мінімуму ймовірність помилок.

Другим об'єктом автоматизації є база даних комплектуючих. Вона містить інформацію про доступні на ринку компоненти, їх характеристики, ціни, наявність і сумісність. Постійне оновлення цієї бази через інтеграцію з платформами постачальників забезпечує актуальність інформації та дозволяє системі формувати рекомендації на основі найсвіжіших даних.

Наступним важливим об'єктом автоматизації є механізм перевірки сумісності комплектуючих. Це включає аналіз технічних параметрів, таких як відповідність сокета процесора материнській платі, підтримка обсягу оперативної пам'яті чи розрахунок потужності блоку живлення. Така функціональність гарантує коректну роботу запропонованих конфігурацій і виключає можливі помилки, які можуть виникнути через недосвідченість користувачів.

Інтерфейс користувача також є важливим об'єктом автоматизації. Він забезпечує зручність і простоту взаємодії з системою, дозволяючи користувачам швидко отримати необхідні рекомендації.

Особливу увагу слід приділити локалізації системи. Підтримка української мови як одного з основних інструментів взаємодії робить систему доступною для ширшої аудиторії, забезпечуючи зручність використання для

україномовних клієнтів.

Таким чином, об'єкти автоматизації охоплюють всі етапи процесу підбору ПК: від аналізу потреб користувачів до формування рекомендацій, перевірки сумісності та забезпечення інтеграції з ринком. Автоматизація цих процесів значно підвищує якість обслуговування клієнтів, знижує витрати часу й ресурсів, мінімізує вплив людського фактора та сприяє зростанню ефективності роботи магазинів комп'ютерної техніки.

3. Вимоги до програмного засобу

3.1 Вимоги до оформлення інтерфейсу

Графічний інтерфейс програмного засобу повинен бути легким (розрахованим на користувача з початковим рівнем інформаційної культури та навичок роботи за комп'ютером), але в той же час функціональним. Блоки керуючих кнопок мінімальну кількість елементів управління, перелік критеріїв має бути написаний зрозумілими словами будь-якому користувачу (рис. 2.1).



Рисунок 2.1. – Інтерфейс головного вікна

Вікно для редагування критеріїв для підбору комплектуючих ПК (рис.

2.2).

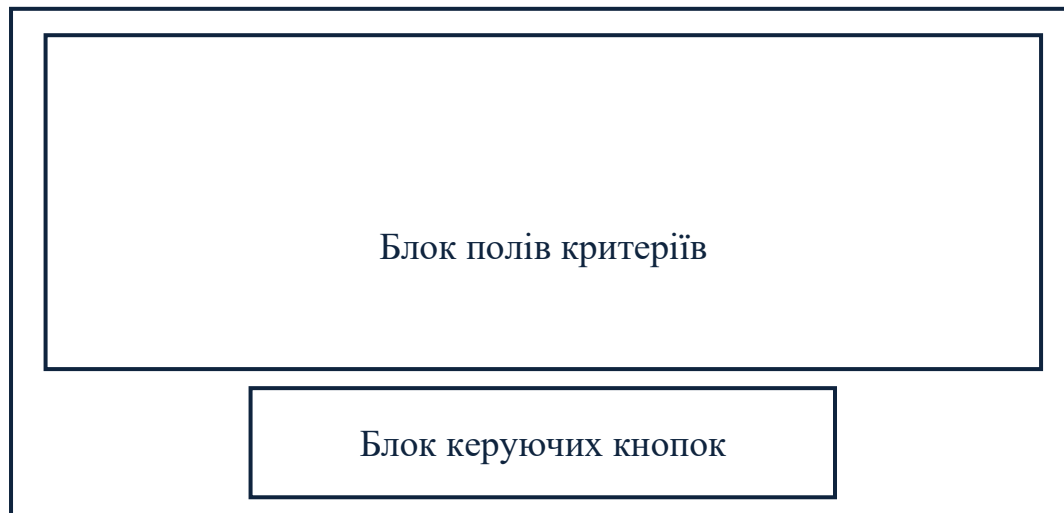


Рисунок 2.2. – Інтерфейс вікна введення інформації

3.2 Вимоги до обробки даних і мов програмування

Усі запити користувачів повинні оброблятися за допомогою Vertex AI. Для створення програмного комплексу повинне використовуватися середовище Microsoft Visual Studio та мова програмування C#. Для формування запитів до Vertex AI використовувати мову програмування Python та відповідні модулі. *3.3 Вимоги до програмного та технічного забезпечення*

Для функціонування програмного засобу необхідне наступне програмне та мінімальне апаратне забезпечення:

- Операційна система – Windows 11 та більш пізні версії;
- Framework 4.5;
- Доступ до інтернету;
- Python 3.12
- Бібліотека Vertex AI
- процесор – 4 потоки обробки або більше;
- оперативна пам'ять – 4 Гб RAM або більше;
- жорсткий диск – 64 Гб або більше.

4. Стадії і етапи розробки програмного засобу

4.1 Склад і зміст робіт із розробки програмного засобу

Розробка програмного засобу для автоматизації роботи головного менеджера супермаркету припускає виконання наступних етапів:

1. Аналіз наочної області, яка буде реалізована в програмному засобі.
2. Вибір ШІ, який буде обробляти запити для роботи програмного засобу.
3. Вибір інструментального середовища, за допомогою якого буде створений програмний продукт.
4. Створення модулів програмного засобу.
5. Тестування і перевірка працездатності програмного продукту в реальних умовах.
6. Виправлення помилок роботи програми і проведення завершального показу Замовнику.

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи програми

Приймання програмного засобу:

1. Перевірка зміни форм-фактору.
2. Перевірка коректності рекомендацій наданих програмним забезпеченням.
3. Перевірка збереження рекомендацій наданих програмним забезпеченням.
4. Перевірка додавання критеріїв підбору комплектуючих.

5.2. Вимоги до введення програмного засобу в дію

Підготовка програмного засобу до експлуатації припускає:

1. Отримання Замовником програмного засобу на диску у вигляді теки із здійснимим файлом програми та всіма необхідними для роботи файлами. Додатково Виконавець може створити інсталятор програмного засобу.
2. Розташування ярлика програмного засобу на Робочому столі в комп'ютері головного менеджера (повинна бути встановлена ОС Windows).
3. Показ відповідальному представнику Замовника (головному менеджеру) всіх програмного засобу.

Висновки до розділу 1

У першому розділі досліджено предметну область інтелектуальних систем (ІС) та проаналізовано існуючі рішення для підбору конфігурацій обчислювальних систем. Головним об'єктом дослідження виступили ІС як інструменти, засновані на штучному інтелекті (ШІ), здатні вирішувати складні, слабоформалізовані задачі, які неможливо автоматизувати традиційними методами. Виявлено, що ключові переваги ІС полягають у їхній адаптивності, здатності до навчання, обробці великих обсягів даних та генерації персоналізованих рішень. Проте вони мають обмеження: відсутність інтуїції, залежність від якості вхідних даних та нездатність до створення принципово нових ідей без участі людини.

Аналіз існуючих сервісів для підбору комплектуючих ПК (PCPartPicker, BuildMyPC, Logical Increments тощо) виявив їхні сильні сторони, такі як автоматична перевірка сумісності компонентів та інтеграція з онлайн-магазинами. Однак більшість цих платформ не використовують потенціал ШІ для персоналізації рекомендацій, обмежені англomовним інтерфейсом і орієнтовані на глобальні або локальні ринки, що ускладнює їх використання українськими користувачами. Це підкреслює потребу в розробці нової системи, яка поєднувала б технологічні інновації з локалізацією та гнучкістю.

Технічне завдання, представлене в розділі, формує основу для створення програмного забезпечення, що враховує виявлені прогалини. Воно передбачає використання Vertex AI для обробки запитів, інтеграцію мов Python і C# для гнучкої реалізації алгоритмів, а також підтримку української мови для покращення доступності. Система орієнтована на автоматизацію підбору конфігурацій з урахуванням індивідуальних потреб користувачів, що включає аналіз бюджету, призначення ПК та технічних параметрів.

Таким чином, розділ 1 обґрунтовує актуальність розробки інтелектуальної системи, яка поєднає переваги сучасних технологій ШІ з урахуванням специфіки українського ринку. Подальша реалізація проекту,

заснована на вимогах технічного завдання, має забезпечити ефективний інструмент для оптимізації процесу вибору обчислювальних систем, усуваючи недоліки існуючих аналогів.

РОЗДІЛ 2. СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Концептуальна модель використання інформаційної системи

Перелік основних скорочень та використовуваних термінів до нашої предметної області представлений у таблиці 2.1.

Таблиця 2.1 Глосарій

Термін	Опис терміну
1	2
1 Основні поняття та категорії проекту	
Програмне забезпечення	сукупність програм системи оброблення інформації та програмних документів, необхідних для експлуатації цих програм
форм-фактор	стандарт, що задає габаритні розміри технічного виробу, а також описує додаткові сукупності його технічних параметрів
Алгоритм	послідовність дій для вирішення певної задачі.
Автоматизація	є одним з напрямів науково-технічного прогресу, який спрямовано на застосування саморегульованих технічних засобів, економіко-математичних методів і систем керування, що звільняють людину від участі у процесах
База даних	програмний або програмно-апаратний комплекс, розроблений для зберігання великого обсягу різноманітної інформації.
Штучний інтелект (ШІ)	галузь інформатики, що займається розробкою алгоритмів та систем, здатних виконувати завдання, які зазвичай вимагають людського інтелекту, такі як розпізнавання образів, прийняття рішень та інші.

Продовження таблиці 2.1

1	2
Інтегрована середовище розробки	це комплекс програмного забезпечення, яке містить все необхідне для розробки, компіляції, лінкінга і налагодження коду.
Командний рядок	різновид текстового інтерфейсу між людиною і комп'ютером, в якому інструкції комп'ютера даються в основному шляхом введення з клавіатури текстових рядків.
Масив	це сукупний тип даних, який дозволяє отримати доступ до всіх змінним одного і того ж типу даних через використання одного ідентифікатора.
Область видимості	визначає, де можна використовувати змінну.
ОС	Операційна Система - комплекс взаємопов'язаних програм, призначених для управління ресурсами комп'ютера та організації взаємодії з користувачем.
Програма	набір інструкцій, які вказують комп'ютеру, що йому потрібно робити.
Інтелектуальна система	програмно-апаратний комплекс, що використовує методи штучного інтелекту для автоматизації процесів аналізу, прогнозування або прийняття рішень.

Як відомо, розробці потужного програмного засобу за допомогою конкретного інструментального середовища повинен передувати процес проектування його структури та функціональності. При цьому повинен забезпечуватися високий рівень і докладний опис логіки додатку, що

задовольняє системним вимогам та можливим обмеженням. Крім того, у процесі створення та функціонування інформаційної системи потреби користувачів можуть змінюватися або уточнюватися, що призводить до більшого ускладнення розробки і супроводу таких систем [20, с. 10].

Досить відомою методологією візуального моделювання вважається UML (Unified Modelling Language). Це уніфікована мова моделювання, призначена для візуалізації та документування систем, у тому числі програмного забезпечення [19, с. 222].

Візуальне моделювання в UML можна розглядати як процес поступового переходу від загальної, абстрактної концептуальної моделі системи до логічної, а згодом – до фізичної моделі програмного забезпечення. Для цього спочатку створюють модель у вигляді діаграми варіантів використання (use case diagram), яка визначає функціональне призначення системи, тобто описує, що саме система виконуватиме під час своєї роботи. Ця діаграма слугує початковим концептуальним представленням або базовою моделлю системи на етапах її розробки та впровадження.



Рисунок 2.1 – Діаграма варіантів використання

Розробка діаграми варіантів використання спрямована на досягнення кількох основних цілей. По-перше, вона допомагає визначити межі та контекст модельованої предметної області на початкових етапах проектування системи. По-друге, дає змогу сформулювати загальні вимоги до функціональної поведінки проєктованої системи. Також діаграма слугує базовою концептуальною моделлю, яка згодом деталізується у вигляді логічних і фізичних моделей. Окрім цього, вона забезпечує створення початкової документації, необхідної для взаємодії між розробниками, замовниками та користувачами.

Основна ідея цієї діаграми полягає в представленні системи у вигляді сукупності сутностей або акторів, які взаємодіють із системою через так звані варіанти використання. Актор (actor) – це будь-яка зовнішня сутність, яка впливає на систему. Це може бути людина, пристрій, програма або інша система, що взаємодіє з моделлю, як це визначає розробник. Варіант використання (use case) описує сервіси, які система надає актору, тобто це набір дій, які система виконує під час взаємодії з актором. При цьому не деталізується, як саме реалізується ця взаємодія.

У загальному вигляді діаграма варіантів використання є спеціальним графом, що графічно відображає конкретні варіанти використання, акторів, можливі інтерфейси та зв'язки між цими елементами. Окремі компоненти діаграми можуть бути об'єднані у прямокутник, який позначає проєктовану систему загалом.

2.2 Функціональна модель інформаційної системи

IDEF0 (Integration Definition for Function Modeling) – це структурований метод моделювання, що використовується для опису функцій, процесів або систем із фокусом на їхній взаємодії та потоках інформації, матеріалів чи ресурсів. IDEF0 належить до сімейства стандартів IDEF (Integration Definition)

і створений для графічного представлення складних систем із метою їхнього аналізу, розуміння, проєктування чи вдосконалення.

Метод має на меті виявлення, організацію та документування ключових функцій системи або процесу, а також зв'язків між ними. Він підходить як для моделювання нових, так і для аналізу існуючих систем. Основою IDEF0 є графічне представлення, де функції зображені у вигляді прямокутників, а потоки інформації, матеріалів чи управління – у вигляді стрілок, які входять у блок або виходять із нього. Стрілки поділяються на кілька типів: вхідні дані (ресурси чи інформація, що трансформуються в результат функції), управління (правила, обмеження чи вимоги, що впливають на виконання функції), механізми (засоби або ресурси, за допомогою яких функція виконується) і вихідні дані (результати виконання функції).

Модель будується за ієрархічною структурою, де спочатку створюється високорівнева абстрактна модель (контекстна діаграма), яка поступово деталізується на нижчих рівнях. Для забезпечення чіткості та уніфікації моделі використовується стандартизований підхід, описаний у документах, таких як стандарт IDEF0, що базується на діаграмах SADT (Structured Analysis and Design Technique).

Метод має низку переваг. Він є універсальним і застосовується в різних галузях – від виробничих процесів до інформаційних систем. Графічний формат спрощує комунікацію між різними зацікавленими сторонами, а також допомагає аналізувати та оптимізувати процеси, ідентифікувати проблеми, дублювання функцій чи неефективність.

Дана програма написана за допомогою мов програмування Python та C#. Вона надає рекомендації користувачу щодо комплектуючих персонального комп'ютера та можливості зберегти їх. А також містить інструменти для редагування критеріїв адміністратором (рис. 2.2).

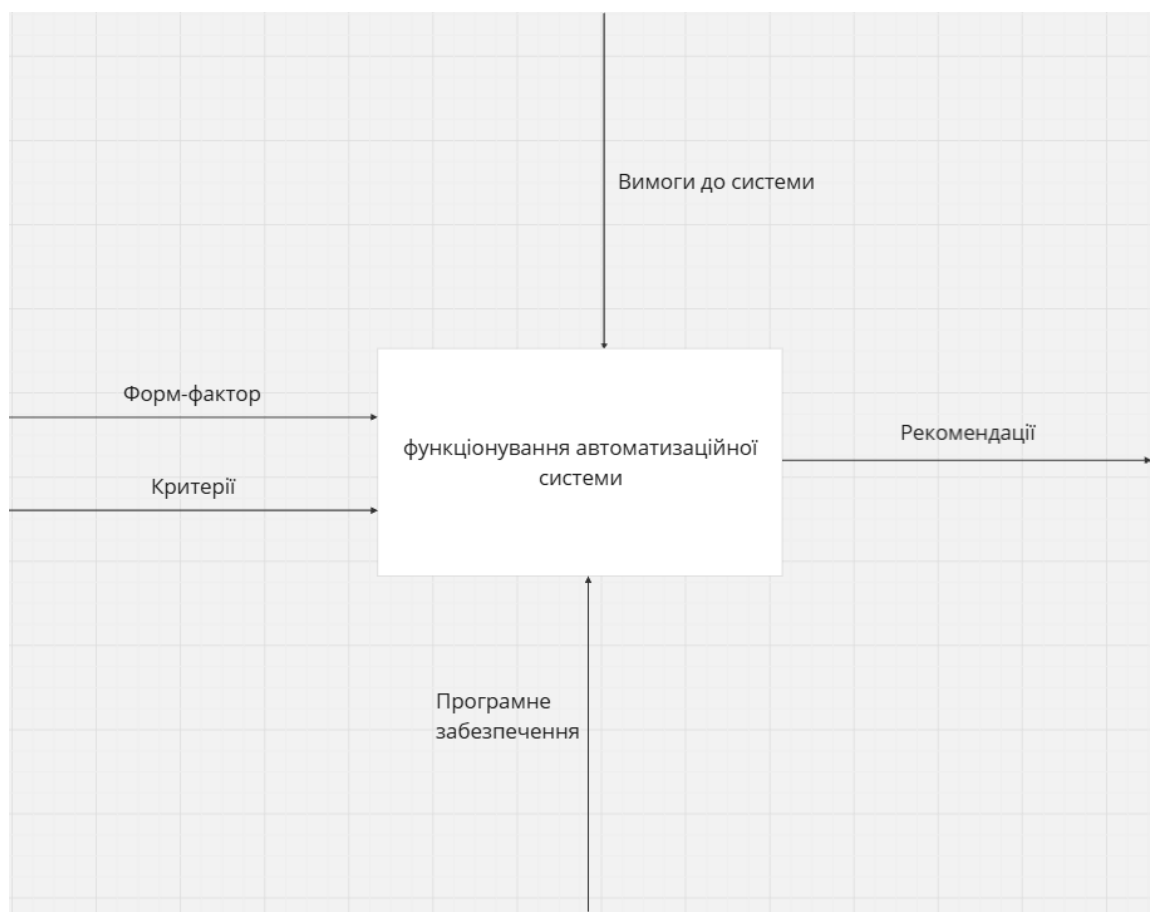


Рисунок 2.2 – Контекстна діаграма IDEF0

Вхідними даними для даної автоматизованої системи є критерії та форм-фактор бажаного персонального комп'ютера, які системі надає користувач. Після чого система опрацьовує їх та надає рекомендовані комплектуючі ПК, які відповідають критеріям.

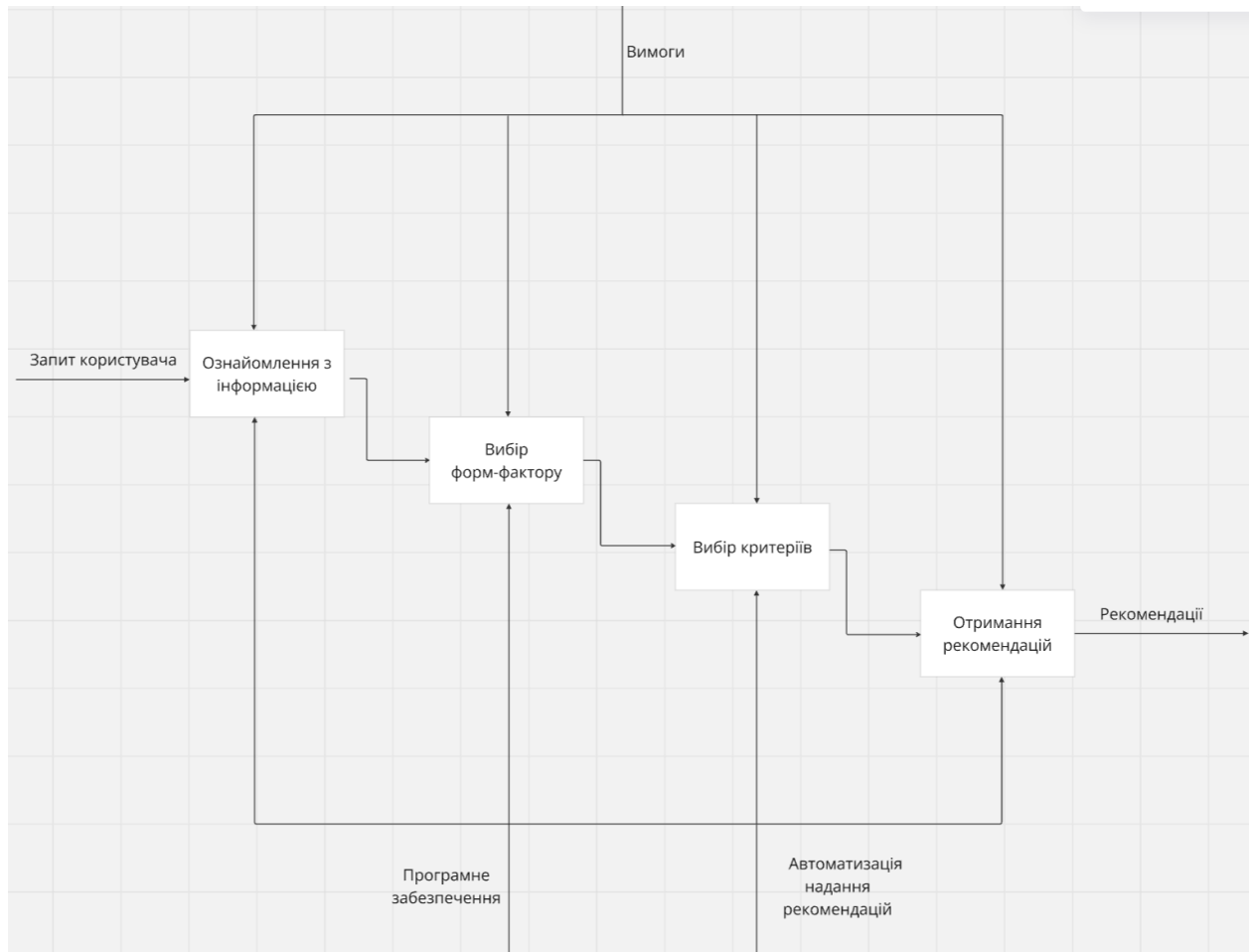


Рисунок 2.3 – Декомпована діаграма IDEF0

Вихідними даними є рекомендація, яка містить в собі такі рекомендовані назви комплектуючих: процесор, відеокарта, оперативна пам'ять, блок живлення, материнська плата, корпус тощо. Ці рекомендації можуть бути збережені в окремому файлі та використані для створення персонального комп'ютера (рис. 2.3).

2.3 Специфікація функціональних та нефункціональних вимог

Мова IDEF0 стандартизована з точки зору синтаксису та семантики, тому моделі є чітко визначеними, добре структурованими, легкими для розуміння, простими у модифікації та використанні, а також можуть бути розширені до будь-якої глибини деталізації. Моделі IDEF0 також є гнучкими, масштабованими та адаптованими до різних ситуацій та умов. Моделі IDEF0

можна вважати тривимірними, оскільки будь-яка двовимірна діаграма моделі функції IDEF0 може бути розширена дочірніми функціями, представленими на різних рівнях моделі. Кількість шарів дочірніх функцій також не обмежена.

Моделі IDEF0 найчастіше розробляються і використовуються системними інженерами для моделювання та аналізу складних систем, вивчення функцій і взаємозв'язків компонентів системи, а також для моделювання діяльності підприємств [Kim, с. 125].

Список та опис усіх функціональних вимог розроблюваної системи и атрибутів вимог наведено у таблиці 2.2.

Таблиця 2.2 - Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
01	Вибір форм фактору	Обов.	Середня	Розробник
02	Редагування критеріїв	Обов.	Висока	Розробник
03	Збереження рекомендацій	Необов.	Висока	Розробник
04	Управління інформацією бота	Необов.	Висока	Розробник

До нефункціональних вимог відносяться всі вимоги, які не стосуються функціональності проекту. Детальний опис нефункціональних вимог наведено у таблиці 2.3.

Таблиця 2.3 – Специфікація нефункціональних вимог

№	Назва вимоги	Характеристика
1	2	3
1 Застосовність		
1.1	Час для навчання користувачів	Від 5 хв. (при наявності інструкції)
1.2	Вимірюваний час відгуку для типових кнопок та меню	До 1с.

Продовження таблиці 2.3

2 Надійність		
2.1	Середній час безвідмовної роботи	Безтерміново.
2.2	Середнє напрацювання до ремонту ПК	Безтерміново.
3 Робочі характеристики		
3.1	Швидкодія для транзакції	Від 0.5 до 2х с.
3.2	Продуктивність (в кількості одночасних користувачів)	Від 1го до 1000 чоловік.
4 Проектні обмеження		
4.1	Мови програмування	Python, C#
5	Вимоги до документації, призначеної для користувача, і до системи допомоги	Вимоги вказані в ТЗ
6 Інтерфейси		
6.1	Інтерфейси користувача	Згідно з ТЗ.
6.2	Програмні інтерфейси	Вказуються замовником.
6.3	Комунікаційні інтерфейси	Вказуються замовником.
7	Вимоги до ліцензування	Згідно з ТЗ.
8	Вживані стандарти	Згідно з нормативними документами та ГОСТами

Висновки до розділу 2

У даному розділі проведено детальний аналіз специфікації вимог до розроблюваної інформаційної системи, яка забезпечує автоматизацію вибору комп'ютерних комплектуючих на основі заданих критеріїв. У процесі аналізу визначено концептуальну модель використання системи, основні терміни й поняття, функціональні та нефункціональні вимоги, а також характеристики взаємодії системи з користувачем.

Таким чином, у цьому розділі закладено основу для реалізації ефективної інформаційної системи, яка сприятиме автоматизації процесу вибору комп'ютерних комплектуючих, оптимізуючи взаємодію користувачів із системою та забезпечуючи високу якість обслуговування.

РОЗДІЛ 3 ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Загальний огляд технологічного стеку

Для розробки інтелектуальної системи було використано мову програмування C#. C# це об'єктно- і компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки такої концепції роботи. Завдяки цьому C# підходить для створення та застосування програмних компонентів. З моменту свого створення мова C# збагатилася функціями для підтримки нових робочих навантажень і сучасними рекомендаціями по розробці програмного забезпечення [10, с. 5].

Мова C# інтегрується з платформою .NET, що надає розробникам багатий набір бібліотек для роботи з мережею, базами даних, графічними інтерфейсами та іншими компонентами. Завдяки цьому розробники можуть створювати як кросплатформні рішення з використанням .NET Core, так і програми, оптимізовані для Windows. Суворі типізація та автоматичне управління пам'яттю в C# забезпечують стабільність і безпеку програм, що є важливим у розробці масштабованих рішень.

Інтерфейс інтелектуальної системи було створено за допомогою Windows Forms, він є компонентом платформи .NET, який використовується для створення настільних застосунків із графічним інтерфейсом користувача. Він забезпечує зручні інструменти для візуального проектування інтерфейсу за допомогою інтегрованих середовищ розробки, таких як Visual Studio. Завдяки Windows Forms розробник отримує доступ до широкого набору стандартних елементів керування, таких як кнопки, текстові поля, списки та таблиці. Ці елементи можна налаштовувати відповідно до потреб конкретного проєкту, що дозволяє створювати унікальні інтерфейси.

Windows Forms дозволяє розробникам створювати Windows додатки шляхом використання візуального редактора. Це дозволяє легко створювати інтерфейс користувача та додавати різні елементи, такі як кнопки, текстові поля, списки тощо [7, с. 105].

Однією з ключових переваг Windows Forms є підтримка подійно-орієнтованого програмування, що дозволяє обробляти взаємодію користувача з елементами інтерфейсу. Крім того, компонент забезпечує доступ до низькорівневих функцій операційної системи Windows, що робить можливим створення складних програм з поглибленим управлінням системними ресурсами.

Поєднання мови C# з Windows Forms відкриває широкі можливості для створення програмного забезпечення. Їх використовують для розробки корпоративних систем, мультимедійних застосунків, утиліт для адміністрування та програм для автоматизації завдань. Завдяки своїй простоті у використанні та потужним функціональним можливостям, ці інструменти дозволяють розробникам створювати естетичні та функціональні настільні застосунки, які відповідають сучасним вимогам.

Для роботи мови програмування python в середовищі мови програмування C# було використано IronPython. IronPython – реалізація мови програмування Python з відкритим кодом, яка тісно інтегрована з .NET. IronPython може використовувати бібліотеки .NET і Python, водночас інші мови .NET теж можуть використовувати код Python [5, с. 33]. Завдяки цьому IronPython стає потужним інструментом для створення програмного забезпечення, яке поєднує простоту й гнучкість Python із продуктивністю та масштабованістю .NET.

IronPython підтримує всі основні функції мови Python, включаючи динамічну типізацію, роботу з модулями, об'єктно-орієнтоване програмування та багатопоточність. Вона дозволяє програмам, написаним на Python, безпосередньо взаємодіяти з класами, бібліотеками та компонентами .NET, такими як Windows Forms, WPF, LINQ, ASP.NET та інші. Ця інтеграція

відкриває нові можливості для створення програм із багатим функціоналом, використовуючи потужну екосистему .NET.

IronPython також базується на Dynamic Language Runtime (DLR), що є розширенням середовища виконання .NET. Це забезпечує можливості динамічного програмування, дозволяючи змінювати та адаптувати поведінку програми під час її виконання. Крім того, IronPython надає засоби для зручної інтеграції Python-скриптів у проекти, написані на C#, що дозволяє створювати змішані застосунки, у яких Python використовується для сценаріїв, а C# – для основної логіки.

Таким чином, IronPython є ідеальним рішенням для розробників, які прагнуть поєднати переваги Python із потужністю та масштабованістю екосистеми .NET.

Для створення запитів до сервісу Vertex AI була використана мова програмування Python. Python – це універсальна мова, що широко використовується в усьому світі для самих різних цілей – бази даних і оброблення текстів, вбудовування інтерпретатора в ігри, програмування GUI, швидке створення прототипів (RAD) програмування Internet і Web додатків – серверних (CGI), клієнтських (роботи), Web-серверів і серверів додатків [22, с. 7].

Обробку запитів користувача виконував сервіс Vertex AI. Vertex AI – це набір інструментів та сервісів машинного навчання від Google. Він дозволяє розробникам легко побудувати, розгорнути та масштабувати моделі машинного навчання [18, с. 29]. Вона поєднує в собі весь спектр необхідних інструментів для роботи зі штучним інтелектом у єдиному середовищі, що значно знижує складність управління проектами ІІІ та МН.

Vertex AI дозволяє створювати як кастомні моделі, так і використовувати попередньо навчені моделі Google для вирішення таких завдань, як класифікація зображень, аналіз тексту, розпізнавання мовлення тощо. Платформа забезпечує автоматизацію ключових процесів, зокрема обробки даних, навчання моделей, вибору гіперпараметрів та моніторингу

продуктивності. Вона також підтримує перенесення навчання, що дозволяє адаптувати вже створені моделі до нових наборів даних.

Vertex AI інтегрується з іншими сервісами Google Cloud, такими як BigQuery, Cloud Storage і Dataflow, що створює умови для повного циклу роботи з даними та їх аналізу. За допомогою Python та SDK google-cloud-aiplatform, розробники можуть використовувати Vertex AI для автоматизації всіх етапів проєкту, від обробки даних до розгортання готових рішень у виробничих середовищах.

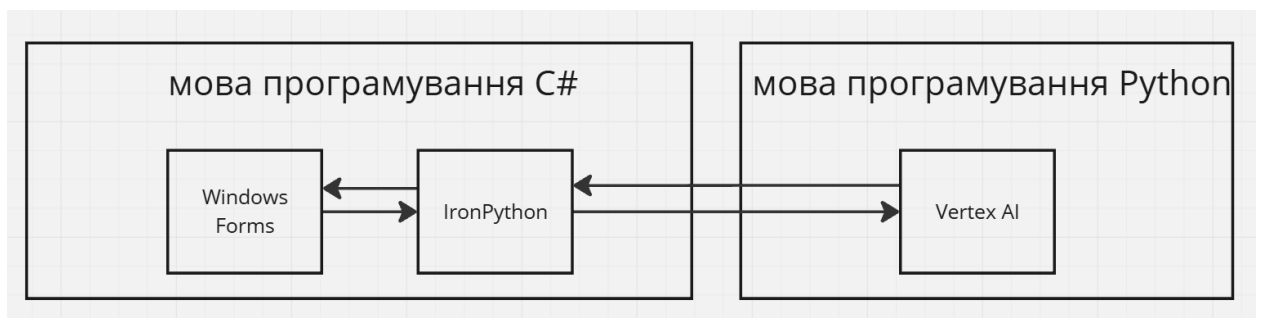


Рисунок 3.1 - Структурна схема взаємодії компонентів технічного стеку

На структурній схемі зображено взаємодію компонентів інтелектуальної системи, яка поєднує мову програмування C# та Python для ефективної роботи з Vertex AI (Рис 3.1).

Система складається з декількох ключових компонентів.

У лівій частині схеми розташовані елементи мови програмування C#, серед яких виділяються Windows Forms та IronPython. Windows Forms відповідає за створення графічного інтерфейсу користувача (GUI), який забезпечує взаємодію користувача із системою.

IronPython виконує роль мосту між C# і Python. Він дозволяє інтегрувати Python-код у середовище C#, забезпечуючи виконання скриптів Python безпосередньо в .NET-додатку.

У правій частині схеми розташована мова програмування Python, яка використовується для безпосередньої взаємодії з Vertex AI.

Взаємодія між компонентами працює таким чином: користувач вводить необхідні дані або команди через інтерфейс Windows Forms. Ці дані передаються в Python через IronPython. Python, використовуючи відповідні бібліотеки, формує запити до Vertex AI, отримує результати, які потім повертаються до C#. Нарешті, результати відображаються користувачеві в інтерфейсі Windows Forms.

3.2 Обґрунтування вибору мов програмування

C# є однією з найбільш продуктивних мов програмування для створення настільних додатків завдяки своїй інтеграції з платформою .NET Framework. Використання Windows Forms забезпечує розробникам простий спосіб створювати графічні інтерфейси користувача (GUI), які є інтуїтивно зрозумілими та зручними для взаємодії з кінцевими користувачами. Це важливо, коли потрібно надавати користувачам доступ до складних функцій, таких як створення запитів до Vertex AI, через прості віконні інтерфейси. Крім того, C# дозволяє інтегрувати широкий спектр функцій, таких як робота з файлами, локальними базами даних і багатопотоковістю, що забезпечує стабільність і високу продуктивність програми.

C# та Windows Forms широко використовуються у створенні графічних інтерфейсів для різноманітних настільних додатків. Наприклад, у Microsoft Office Tools розробляються додаткові модулі та інструменти для автоматизації роботи з Excel і Word, які використовують Windows Forms для створення зручних віконних інтерфейсів. Також ці технології часто застосовуються для створення плагінів у складних програмних комплексах, таких як AutoCAD, де C# дозволяє автоматизувати повторювані завдання у сфері інженерного проектування.

Python, у свою чергу, є ідеальною мовою для роботи з хмарними платформами, зокрема з Vertex AI. Його популярність у сфері машинного

навчання та штучного інтелекту зумовлена доступом до багатого набору бібліотек і фреймворків.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веброзробці, розробці програмного забезпечення та інших галузях [21].

Python забезпечує зручність підключення до Vertex AI, дозволяючи виконувати такі завдання, як навчання моделей, виконання запитів, обробка великих обсягів даних і отримання прогнозів. Завдяки своїй простоті Python дозволяє автоматизувати складні процеси взаємодії з Vertex AI, знижуючи поріг входу для розробників.

Python, завдяки своїй універсальності, активно використовується у проєктах, пов'язаних із великими даними, штучним інтелектом та автоматизацією. Наприклад, YouTube використовує Python для побудови алгоритмів рекомендацій і обробки даних відео, тоді як Dropbox застосовує цю мову для забезпечення роботи внутрішніх сервісів і управління файлами. Відомими прикладами інтеграції Python у хмарні технології є використання цієї мови для аналітики та автоматизації на платформах штучного інтелекту.

Поєднання цих двох мов відкриває значні можливості для створення багатофункціональних додатків. Використання C# із Windows Forms дозволяє створювати привабливий і функціональний інтерфейс для користувачів, який може включати форми для введення параметрів запитів, відображення результатів аналізу чи навчання моделей. З іншого боку, Python може обробляти всі аспекти, пов'язані із взаємодією з Vertex AI: від передавання даних і налаштування гіперпараметрів до отримання результатів прогнозування.

Такий підхід дає кілька важливих переваг. По-перше, це зручність для користувачів, які можуть працювати з програмою через графічний інтерфейс,

що забезпечується Windows Forms. По-друге, це потужність і гнучкість Python для роботи з хмарними сервісами та обробки даних. Поєднання цих компонентів дозволяє створити модульну систему, в якій кожна мова відповідає за свої специфічні завдання: C# обробляє візуалізацію і локальну взаємодію, а Python забезпечує високорівневу обробку даних і взаємодію з Vertex AI.

Результатом такого підходу є додатки, які поєднують найкраще з обох світів: професійний інтерфейс для кінцевих користувачів і потужні інструменти для виконання складних алгоритмів машинного навчання в хмарному середовищі. Це рішення ідеально підходить для проєктів, де потрібна одночасно простота використання і висока функціональність.

3.3 Інструменти для розробки

Інтегроване середовище розробки Visual Studio є творчим стартовим майданчиком, який можна використовувати для редагування, налагодження та складання коду, а також для публікації програми. На додаток до стандартного редактора і відладчика, що надаються більшістю інтегрованих середовищ розроблення, Visual Studio містить компілятори, засоби завершення коду, графічні конструктори та багато інших функцій для поліпшення процесу розроблення програмного забезпечення [29].

Visual Studio є ідеальним інструментом для розробки мовою C# з використанням Windows Forms завдяки своїм широким можливостям і потужній підтримці .NET. Воно забезпечує швидке та ефективне створення графічного інтерфейсу користувача (GUI) завдяки візуальному дизайнеру форм, який дозволяє легко додавати елементи інтерфейсу, такі як кнопки, текстові поля й списки, шляхом простого перетягування. Це робить процес розробки інтуїтивним і значно скорочує час створення інтерфейсів.

Visual Studio також надає можливість ефективно працювати з подіями. Наприклад, додавання обробників для таких подій, як натискання кнопок або введення даних, автоматизується середовищем, яке створює шаблони необхідного коду, що спрощує процес розробки.

Крім того, Visual Studio пропонує доступ до великої кількості бібліотек через платформу .NET, які можна використовувати для створення високофункціональних додатків. Це включає роботу з файлами, багатопотоковість і інтеграцію з базами даних, що є важливими аспектами у створенні комплексних додатків. Інтеграція з базами даних значно спрощується завдяки вбудованим засобам для підключення, виконання запитів і обробки даних.

Ще однією вагомою перевагою є можливість кросплатформної розробки завдяки підтримці .NET Core. Це дозволяє створювати додатки, які можуть працювати не лише на Windows, але й на інших платформах, таких як macOS і Linux.

Зручність і продуктивність Visual Studio значно підвищуються завдяки функціям автоматизації рутинних завдань, таких як збірка проекту, тестування, налагодження та розгортання. Інтелектуальний редактор коду з функціями автодоповнення, підсвічування синтаксису та виправлення помилок також сприяє спрощенню розробки й дозволяє зосередитися на бізнес-логіці програми.

Visual Studio є ідеальним інструментом для розробки додатків мовою C# з використанням Windows Forms. Воно поєднує в собі потужні інструменти для створення графічних інтерфейсів, зручний редактор коду та інструменти для тестування й налагодження, що робить його ключовим вибором для розробників. Завдяки Visual Studio можна швидко створювати, тестувати й розгортати додатки будь-якої складності, забезпечуючи високу продуктивність і якість програмного забезпечення.

PyCharm це один із найпопулярніших інструментів для розробки на Python завдяки своїм багатофункціональним можливостям, зручності

використання та підтримці сучасних технологій. Для кращого розуміння його переваг наведемо порівняння PyCharm з іншими популярними середовищами розробки(табл. 4.1).

Таблиця 4.1 – Порівнювальна таблиця середовищ розробки

Характеристика	PyCharm	Visual Studio Code	Spyder	Thonny
Інтелектуальний редактор коду	Так, з автодоповненням і аналізом помилок	Обмежений, залежить від плагінів	Мінімальний	Мінімальний
Вбудований відладчик	Так	Частково (через плагіни)	Так	Так
Підтримка вебфреймворків	Django, Flask, FastAPI	Залежить від плагінів	Немає	Немає
Робота з базами даних	Вбудована	Через плагіни	Немає	Немає
Інтеграція з Git	Так	Так	Часткова	Немає
Кросплатформеність	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux
Інтеграція з віртуальними середовищами	Так	Так	Так	Немає
Вартість	Безкоштовна (Community) і платна (Professional)	Безкоштовна	Безкоштовна	Безкоштовна
Призначення	Універсальний інструмент для Python	Універсальний редактор для багатьох мов	Наукові обчислення	Для новачків

PyCharm є потужним інструментом для розробки на Python, який забезпечує широкий спектр можливостей для підвищення продуктивності та ефективності роботи [15, с. 21]. Вибір PyCharm для розробки програмного забезпечення обґрунтований низкою його функціональних можливостей та переваг, які спрощують і оптимізують процес розробки.

Перш за все, PyCharm надає інтелектуальний редактор коду, який забезпечує автодоповнення, підсвічування синтаксису та аналіз помилок у реальному часі. Ці функції підвищують швидкість написання коду та знижують ймовірність допущення помилок. Завдяки цьому розробники можуть зосередитися на логіці програми, а не на дрібних технічних деталях.

Крім того, PyCharm інтегрується з популярними вебфреймворками, такими як Django та Flask, що дозволяє ефективно розробляти вебдодатки. Це особливо важливо для проєктів, які включають серверну логіку чи інтерактивні вебінтерфейси. Наявність підтримки для роботи з базами даних прямо з середовища розробки дозволяє швидко налаштовувати взаємодію з базами даних, виконувати SQL-запити, а також перевіряти структуру та стан даних.

Однією з основних переваг PyCharm є його вбудований відладчик, який дозволяє легко знаходити й виправляти помилки в коді. Це знижує ризик непрацездатності програмного забезпечення на етапі його впровадження. Для більшого контролю якості PyCharm також підтримує інтеграцію з фреймворками автоматизованого тестування, такими як `pytest` і `unittest`, що полегшує створення та виконання тестів.

Інтеграція з системами контролю версій, як-от Git, дозволяє зручно відслідковувати зміни у коді, співпрацювати в команді та легко повертатися до попередніх версій проєкту. PyCharm забезпечує зручний інтерфейс для виконання операцій із репозиторіями, таких як коміти, пуші чи мерджі.

Середовище PyCharm є кросплатформним, що дозволяє розробникам працювати на будь-якій операційній системі – Windows, macOS або Linux. Це забезпечує гнучкість у виборі інструментів та платформи для роботи.

Крім того, PyCharm підтримує інтеграцію з віртуальними середовищами Python, такими як `virtualenv` або `Conda`, що дозволяє ізолювати залежності для різних проєктів, уникнувши конфліктів між бібліотеками.

Використання PyCharm обґрунтоване і його здатністю підвищувати продуктивність розробки завдяки широкому набору інструментів та функцій.

Завдяки цьому середовищу розробники можуть досягати вищої якості коду, швидше знаходити помилки та працювати над проєктами різної складності, від невеликих скриптів до великих комерційних додатків.

Загалом, PyCharm є потужним інструментом для розробки на Python, який сприяє підвищенню продуктивності та комфорту програмістів у будьякому розмірі та складності проєктів [13, с. 13].

Висновки до розділу 3

Розглянутий технологічний стек демонструє ефективне поєднання інструментів для розробки інтелектуальної системи, спрямованої на автоматизацію складних процесів і інтеграцію з хмарними сервісами. Мова програмування C# у тандемі з Windows Forms забезпечує створення функціонального та зручного графічного інтерфейсу, а Python, із його потужними бібліотеками, ідеально підходить для роботи з хмарними платформами, зокрема Vertex AI, що спрощує реалізацію алгоритмів машинного навчання.

Інтеграція через IronPython дозволяє ефективно розподіляти завдання між мовами: C# відповідає за інтерфейс, а Python – за взаємодію з хмарними сервісами та обробку даних. Використання середовищ розробки Visual Studio та PyCharm сприяло оптимізації процесу розробки завдяки інструментам автоматизації, тестування та налагодження.

Таким чином, обраний технологічний стек забезпечив ефективне виконання проєкту, поєднуючи простоту використання для кінцевих користувачів із гнучкістю та продуктивністю інструментів для роботи зі складними даними й алгоритмами.

РОЗДІЛ 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

4.1 Особливості програмної реалізації інтелектуальної системи для визначення конфігурації обчислювальної системи

Розробка програмного забезпечення, що здійснює автоматизацію процесу вибору конфігурації обчислювальної системи, передбачала проходження ряду етапів, кожен з яких має ключове значення для забезпечення функціональності, зручності використання та продуктивності системи. Ця інтелектуальна система створювалась із використанням мови програмування C# та інтеграції з Python-скриптами для виконання складних обчислювальних завдань. Основні етапи розробки включають:

1. Аналіз вимог та проектування. На початковому етапі були визначені вимоги до системи, проаналізовані потреби користувачів і виконано дослідження існуючих аналогів. Це дозволило сформулювати ключові функції, серед яких інтерактивний вибір критеріїв, автоматизований запуск Python-скриптів для аналізу та формування рекомендацій, а також формування результатів у зручному форматі.

2. Розробка структури даних. Для зберігання даних використовувались списки, які класифікували можливі конфігурації обчислювальних систем за типами (ПК, ноутбук, міні-ПК). Дані завантажуються із зовнішнього файлу (CFG.data), що містить інформацію про конфігурації, і зберігаються у списках ArrStr1, ArrStr2 та ArrStr3.

3. Інтерфейс користувача. Інтерфейс системи створено у середовищі Visual Studio з використанням Windows Forms. Основні елементи інтерфейсу включають:

- Панель із радіокнопками для вибору типу обчислювальної системи (ПК, ноутбук, міні-ПК).

- Перелік критеріїв, що динамічно змінюється відповідно до вибору користувача.

- Текстове поле для виведення результатів, яке інформує користувача про статус виконання запиту

5. Інтеграція з Python. Система виконує аналіз та формування рекомендацій за допомогою Python-скрипта. Взаємодія реалізована через об'єкт Process, що дозволяє виконувати зовнішні Python-скрипти із передачею параметрів:

- список критеріїв, обраних користувачем.
- тип обчислювальної системи.

6. Обробка результатів. Отримані результати формуються у Python-скрипті та повертаються у вигляді тексту, який відображається у текстовому полі. У разі виникнення помилок система повідомляє користувача про проблему. Основні класи та методи проєкту представлені у таблиці 4.1.

Таблиця 4.1 – Ключові класи та методи

Назва форми/класу	Призначення
Form1	Основне вікно програми, що містить у собі всі елементи з якими має взаємодіяти користувач та поле, яке виводить інформацію користувачу
Form2	Вікно призначене для адміністрування, яке має поле для редагування критеріїв, які може обирати користувач
readFile	Завантаження конфігурацій із зовнішнього файлу та сортування їх за категоріями.
RunPythonScript	Виконання Python-скрипта з передачею параметрів та отримання результатів.

Продовження таблиці 4.1

radioButton_CheckedChanged	Динамічне оновлення списку критеріїв відповідно до обраного типу обчислювальної системи.
button1_Click	Обробка натискання кнопки, формування запиту до Python-скрипта, виведення результатів.

Загальна схема інтелектуальної системи для визначення конфігурації обчислювальної системи подана на рисунку 4.1

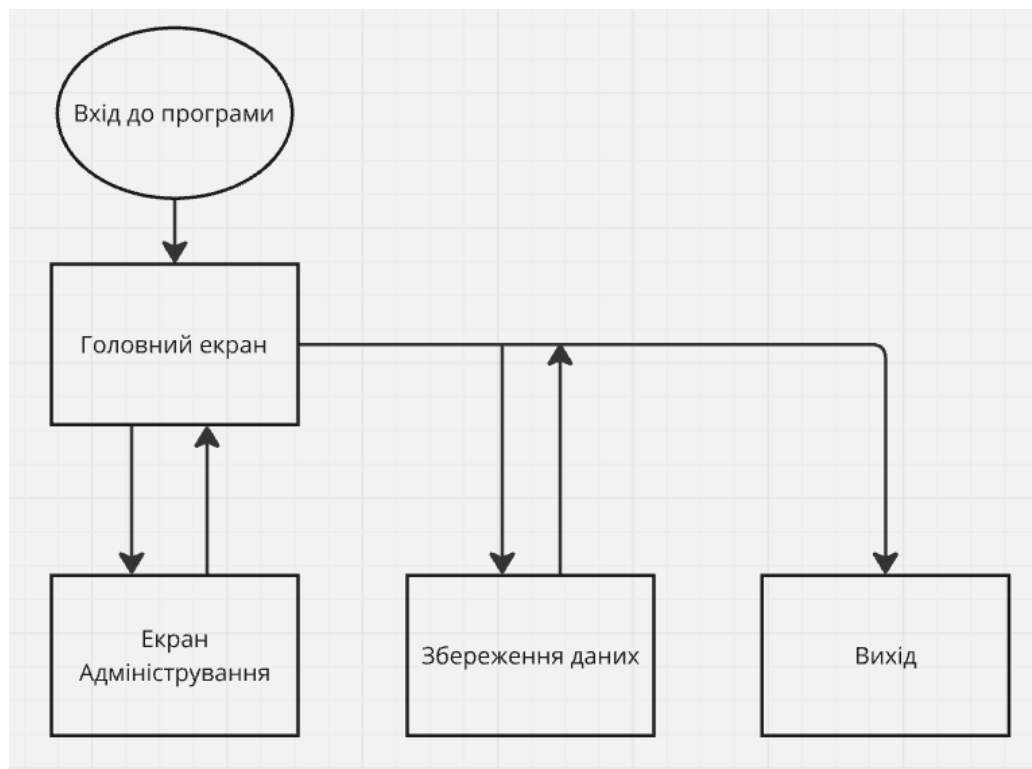


Рисунок 4.1. – Загальна схема блоків програмного засобу

Для обробки даних було використано Vertex AI, інструмент від Google, що дозволяє створювати, тренувати та інтегрувати моделі штучного інтелекту в бізнес-процеси. Порівняно з іншими платформами, Vertex AI відрізняється

високою інтеграцією з екосистемою Google Cloud, що забезпечує масштабованість, зручність роботи з великими даними і потужні інструменти автоматизації, такі як AutoML. Водночас, як і конкуренти, він підтримує кастомізацію моделей і доступ до API для інтеграції в різні середовища. Однак ключова перевага Vertex AI – це орієнтація на спрощення розробки ШІ-рішень завдяки єдиній платформі.

Одним із критичних аспектів роботи з ШІ є точне визначення ролі для моделі. Роль для штучного інтелекту – це чітко визначене завдання або набір завдань, які модель має виконувати. Вона включає мету, межі застосування, контекст використання і очікувані результати. Цей опис дозволяє налаштувати модель так, щоб вона максимально ефективно виконувала поставлені завдання, враховуючи специфіку даних і потреби користувачів. Наприклад, модель, що аналізує текст, повинна враховувати контекст, мовні особливості та стиль.

Точність у визначенні ролі для ШІ є фундаментальною. Це не лише зменшує ризики помилок, але й допомагає моделі діяти в рамках заданого контексту. Якщо межі не окреслені, модель може давати некоректні результати або використовуватись не за призначенням. Наприклад, модель для класифікації зображень не пристосована для аналізу текстів, і її використання в такому контексті може призвести до помилкових висновків.

Більше того, правильно визначена роль ШІ сприяє дотриманню етичних стандартів і підвищує довіру до системи, особливо у сферах, де від рішень залежить здоров'я чи фінансова безпека. Це також оптимізує ресурси, забезпечуючи більш ефективне використання обчислювальних потужностей. Таким чином, роль формує рамки роботи ШІ, визначає його функції і забезпечує ефективну інтеграцію в реальні завдання.

Тому було вирішено використовувати надання трьох різних ролей для кожного типу форм-фактору:

case "Настільний ПК":

return ""твоя роль консультант по підбору комплектуючих ПК у форм-

факторі настільного ПК

відповідь має бути у вигляді:

CPU: *процесор який рекомендується*

GPU: *грифічний процесор який рекомендується*

ОЗУ: *рекомендована оперативна пам'ять*

Материнська плата: *Рекомендована материнська плата*

Накопичувач(i): *рекомендовані накопичувач(i)*

Охолодження процесора: *рекомендоване охолодження процесора*

Корпус: *Назва корпусу*

окрім комплектуючих ПК більше не повинно бути ніякого тексту""

case "Ноутбук":

return ""твоя роль консультант по підбору ноутбуків

відповідь має бути у вигляді:

Ноутбук: *повна назва ноутбуку*

CPU: *процесор встановлений у рекомендованій моделі ноутбуку*

GPU: *грифічний процесор встановлений у рекомендованій моделі ноутбуку*

ОЗУ: *оперативна пам'ять встановлена у рекомендованій моделі ноутбуку*

Накопичувач(i): *накопичувач(i) встановлений у рекомендованій моделі ноутбуку*

Роздільна здатність екрану: *Роздільна здатність екрану встановленого у рекомендованій моделі ноутбуку *

окрім комплектуючих ноутбуку більше не повинно бути ніякого тексту""

case "Міні-ПК":

return ""твоя роль консультант по підбору Міні-ПК

відповідь має бути у вигляді:

Міні-ПК: *повна назва Міні-ПК

CPU: *процесор встановлений у рекомендованій моделі Міні-ПК*

GPU:*грифічний процесор встановлений у рекомендованій моделі
Міні-ПК*

ОЗУ: *оперативна пам'ять встановлена у рекомендованій моделі Міні-
ПК*

Накопичувач(i): *накопичувач(i) встановлений у рекомендованій
моделі ноутбуку*

окрім комплектуючих ноутбуку більше не повинно бути ніякого
тексту""

Надання трьох різних ролей для кожного типу форм-фактора ПК – «Настільний ПК», «Ноутбук», і «Міні-ПК» – має кілька важливих переваг, пов'язаних із гнучкістю, спеціалізацією, і зручністю для користувачів.

По-перше, це дозволяє адаптувати рекомендації до унікальних вимог кожного типу пристрою. Наприклад, для настільних ПК ключовою є можливість модифікації та гнучкість у виборі комплектуючих, таких як CPU, GPU, чи корпус. Для ноутбуків акцент зміщується на компактність, мобільність і інтегровані рішення, де важливо враховувати екран, акумулятор і портативність. Міні-ПК, зі свого боку, орієнтовані на мінімальні розміри та енергоефективність, що робить їх оптимальними для офісів або медіацентрів.

По-друге, такий підхід спрощує взаємодію з користувачем. Різні формати відповідей забезпечують чіткість і цільову інформацію для кожного сценарію. Наприклад, користувач, який шукає комплектуючі для настільного ПК, одразу отримує структуру відповідей, яка відповідає його потребам, уникаючи деталей, що стосуються інтегрованих пристроїв, як у ноутбуках чи Міні-ПК.

По-третє, спеціалізовані ролі покращують ефективність ШІ, оскільки кожен запит обробляється в контексті конкретного форм-фактора. Це дозволяє уникати надлишкових або недоречних рекомендацій, покращуючи релевантність і точність результатів. Наприклад, для Міні-ПК можна пропонувати моделі з оптимальною вбудованою графікою, тоді як для

настільних ПК можливе використання високопродуктивних дискретних відеокарт.

Загалом, створення трьох ролей допомагає забезпечити індивідуальний підхід, врахувати специфіку кожного форм-фактора та зробити рекомендації більш зрозумілими і корисними для кінцевого користувача.

4.2 Функціональні можливості та інструкція користувача інформаційної системи

Після запуску застосунку нас зустрічає мінімалістичний інтерфейс (рис. 4.2), на якому одразу помітна виділена область «Оберіть форм-фактор». Після вибору форм-фактору з'являються критерії (рис. 4.3), згідно з якими будуть надані рекомендації користувачу. Після вибору критеріїв та натискання кнопки «Отримати рекомендацію» з'являється повідомлення: «Запит у обробці. На відповідь зазвичай потрібно не більше 1 хвилини», після очікування з'являється результат роботи програми (рис. 4.4).

За бажанням, після отримання результату його можна зберегти, натиснувши кнопку «Зберегти рекомендацію» та обравши розташування та назву майбутнього файлу (рис. 4.5). Перед збереженням користувач може редагувати рекомендацію або додавати власні примітки безпосередньо у вікні з результатами. Усі внесені зміни, включно з мітками, будуть збережені у файлі рекомендацій, що дозволяє персоналізувати результат під конкретні вимоги або уточнити деталі для подальшого аналізу. Це значно покращує гнучкість роботи із системою та робить процес підбору більш адаптивним до потреб користувача.

Експертна система

Оберіть форм-фактор Відмітьте бажані критерії прапорцем

Настільний ПК
Ноутбук
Міні ПК

Отримати рекомендацію

Зберегти рекомендацію

Рисунок 4.2 – інтерфейс застосунку

Експертна система

Оберіть форм-фактор Відмітьте бажані критерії прапорцем

Настільний ПК
Ноутбук
Міні ПК

☐ Вам потрібна висока продуктивність для ігор?
☐ Вам потрібна висока продуктивність для відеомонтажу?
☐ Вам потрібна висока продуктивність для 3D-моделювання?
☐ Вам потрібна висока продуктивність для складних розрахунку фізичних властивостей матеріалів або рідин?
☐ Плануєте розширювати обсяг оперативної пам'яті в майбутньому?
☐ Ви б віддали перевагу використанню твердотілого накопичувача (SSD) для швидкого доступу до даних?
☐ Ви віддаєте перевагу продукту високого цінового сегменту?
☐ Ви віддаєте перевагу продукту середнього цінового сегменту?
☐ Ви віддаєте перевагу продукту низького цінового сегменту?

Отримати рекомендацію

Зберегти рекомендацію

Рисунок 4.3 – критерії для надання рекомендації

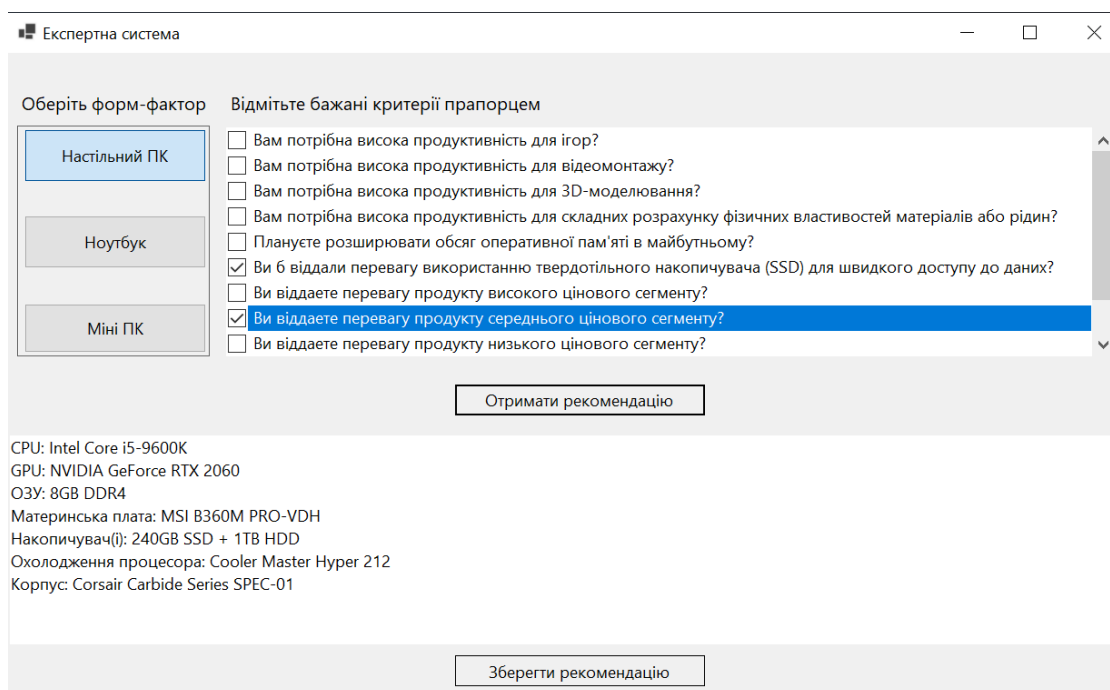


Рисунок 4.4 – Результат роботи програми

Під час збереження рекомендацій користувач може обрати місце, де буде збережено файл, а також задати його найменування (рис. 4.5). Інтерфейс збереження файлу є стандартним для Windows, що забезпечує зручність та зрозумілість процесу для більшості користувачів.

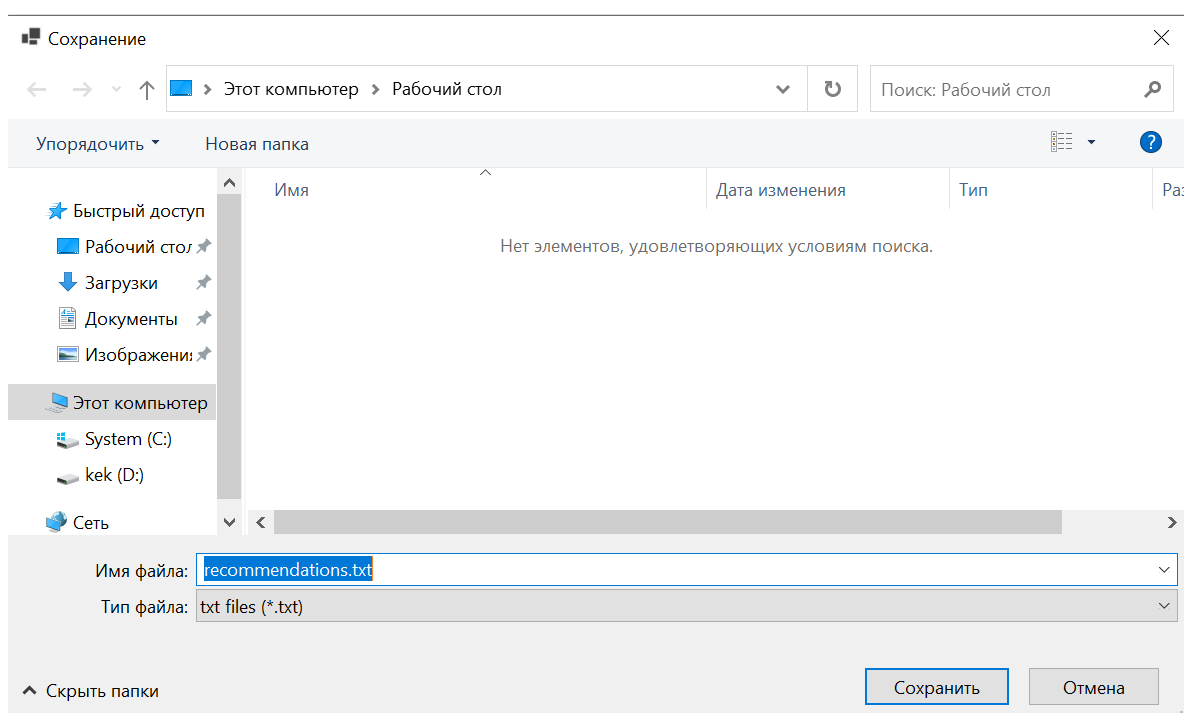


Рисунок 4.5 – Збереження файлу

4.2.1 Покрокова інструкція користування

1. У рамочці під назвою «Оберіть форм-фактор» потрібно обрати потрібний вам форм-фактор ПК (позначка 1 на рис. 4.6) .

2. Після появи критеріїв майбутнього пристрою відмітьте бажані прапорцем у полі під назвою «Відмітьте бажані критерії прапорцем» (позначка 2 на рис. 4.6).

3. Після вибору критеріїв майбутнього пристрою натисніть кнопку «Отримати рекомендацію» (позначка 3 на рис. 4.6). Після натискання у полі під позначкою 4 на Рисунку 4.6 з'явиться напис: «Запит у обробці. На відповідь зазвичай потрібно не більше 1 хвилини», слід зачекати зазначений час.

4. Після отримання рекомендації у полі під позначкою 4 на Рисунку 4.6 за бажанням її можна зберегти, натиснувши кнопку «Зберегти рекомендацію»(позначка 5 на рис. 4.6), після цього слід обрати розташування майбутнього файлу та натиснути кнопку «Зберегти».

The screenshot shows a window titled "Експертна система" (Expert System). It contains two main sections: "Оберіть форм-фактор" (Select form factor) and "Відмітьте бажані критерії прапорцем" (Mark desired criteria with a flag). The first section has three buttons: "Настільний ПК" (Desktop PC), "Ноутбук" (Laptop), and "Міні ПК" (Mini PC). The second section has a list of criteria with checkboxes. The third section has a button "Отримати рекомендацію" (Get recommendation). The fourth section displays a list of hardware specifications. The fifth section has a button "Зберегти рекомендацію" (Save recommendation).

1. Оберіть форм-фактор

2. Відмітьте бажані критерії прапорцем

3. Отримати рекомендацію

4. Запит у обробці. На відповідь зазвичай потрібно не більше 1 хвилини

5. Зберегти рекомендацію

Hardware specifications displayed:

- CPU: Intel Core i5-9400F
- GPU: Nvidia GeForce GTX 1660 Super
- ОЗУ: 16 Гб DDR4
- Материнська плата: ASUS TUF B450-PLUS GAMING
- Накопичувач(і): Kingston A2000 NVMe PCIe SSD 500GB, Seagate Barracuda 2TB HDD
- Охолодження процесора: Cooler Master Hyper 212 LED
- Корпус: Cooler Master MasterBox Lite 5 RGB

Рисунок 4.6 – Ілюстрація до інструкції використання

4.2.2 Адміністрування

Для редагування бажаних критеріїв для підбору комплектуючих ПК потрібно виконати такі дії:

1. Відкрити інтерфейс користувача та натиснути невидиму кнопку в області під позначкою 1 на рисунку 4.7.
2. Після відкриття Редактора вже існуючі рекомендації мають автоматично з'явитися у полі під позначкою 2 на рисунку 4.8.
3. У випадку, якщо поле після відкриття вікна пусте, потрібно знову зчитати збережені рекомендації, натиснувши кнопку «Зчитати дані» (позначка 3 рис. 4.8).
4. У випадку часткового або повного видалення конфігурації слід натиснути кнопку «Зчитати дані» (позначка 3 рис. 4.8).
5. Після редагування слід натиснути кнопку «Зберегти дані» (позначка 4 рис. 4.8).
6. Після збереження даних застосунків слід перезавантажити.

Слід додати, що для додавання параметрів для настільного ПК, ноутбука та міні-ПК їх потрібно розміщувати під відповідними категоріями: [PC], [Laptop] та [MiniPC]. Кожен окремий параметр має займати не більше одного рядка для коректної роботи. Важливо також уникати зайвих відступів перед і після параметра.

Якщо після перезавантаження програми внесені зміни не застосувалися, необхідно перевірити правильність формату запису та повторно зберегти дані. Також рекомендується періодично створювати резервну копію файлу конфігурації, щоб уникнути втрати важливих параметрів у разі збою або некоректного редагування.

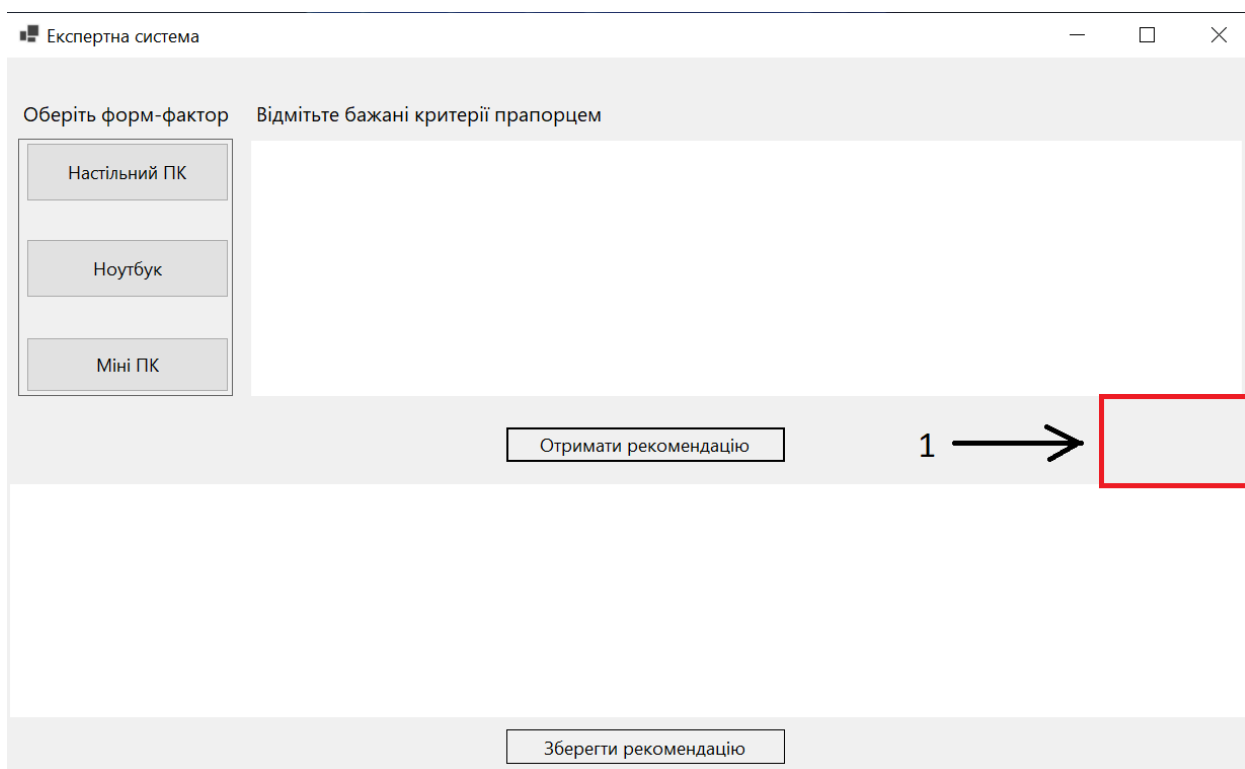


Рисунок 4.7 – Ілюстрація до інструкції налаштування Адміністратора

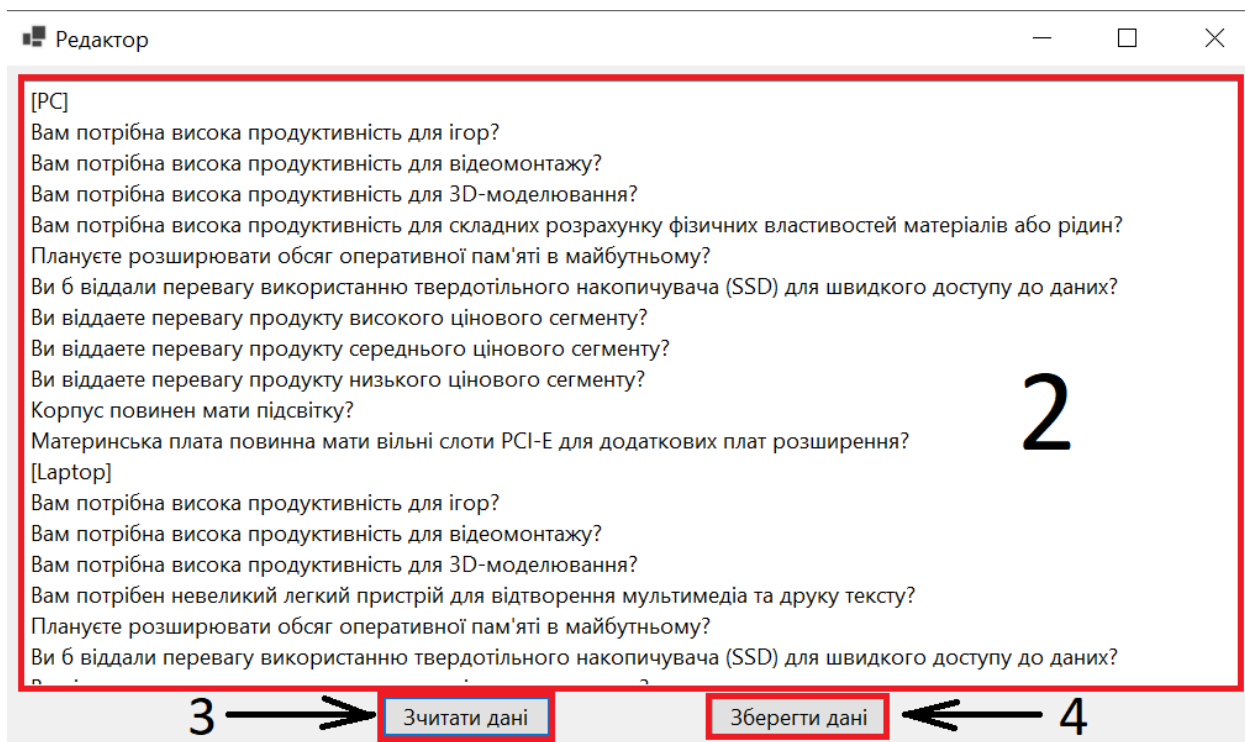


Рисунок 4.8 – Редактор Адміністратора

4.3 Тестування інтелектуальної системи

Тестування програмного забезпечення є ключовим етапом розробки, який дозволяє забезпечити якість продукту, виявити дефекти та гарантувати його стабільне функціонування. Цей процес охоплює аналіз роботи програмного забезпечення відповідно до визначених вимог і специфікацій, забезпечуючи перевірку його відповідності очікуванням користувачів. Тестування не лише сприяє підвищенню рівня довіри до продукту, але й запобігає виникненню критичних проблем на етапі впровадження.

Одним із ключових моментів тестування є визначення критеріїв його завершення. Як зазначає Писаренко, недолік тестування може призвести до випуску продукту з критичними дефектами, а надлишкове тестування – до зайвих витрат, затримки випуску продукту та відволікання тестувальників від пріоритетних завдань [16, с. 118]. Це підкреслює важливість збалансованого підходу до процесу тестування.

Програмне забезпечення глибоко інтегроване в повсякденне життя людини, його дефекти можуть мати серйозні наслідки. Некоректне функціонування програм може викликати не лише погіршення досвіду користувачів, але й призводити до значних фінансових втрат або навіть ставити під загрозу життя людей. Саме тестування дозволяє мінімізувати ризики, виявляючи дефекти на ранніх стадіях розробки. Як зазначає Кудряшова, це сприяє формуванню чітких сценаріїв реалізації процедур і безпосередньо впливає на якість кінцевого продукту [12, с. 11].

Окрім технічних переваг, тестування також відіграє важливу роль у підвищенні прозорості розробки. Воно дозволяє учасникам проєкту краще розуміти стан продукту, ідентифікувати проблемні місця та оперативно приймати відповідні рішення. Як наголошує Авраменко, тестування дає розробникам впевненість у правильному розумінні завдань, менеджерам – чітке уявлення про прогрес проєкту та готовність продукту до експлуатації або продажу [1, с. 9].

Тестування готового програмного забезпечення на базі навченої моделі Vertex AI вимагає особливого підходу, враховуючи специфіку роботи платформи та її інтеграцію в інші системи. Одним із ключових аспектів є перевірка функціональності API моделі, що дозволяє переконатися, що вона правильно обробляє запити, повертає очікувані результати та адекватно реагує на помилки. Оцінка точності моделі здійснюється шляхом тестування її на реальних даних чи контрольних наборах, щоб визначити якість прогнозів та узгодженість роботи.

Важливим є тестування продуктивності, яке визначає швидкість обробки запитів, стабільність роботи під навантаженням і ефективність використання ресурсів. Перевірка моделі в реальних умовах дозволяє оцінити, як вона поводить себе в інтеграції з іншими компонентами програмного забезпечення та відповідає очікуванням користувачів. Особлива увага приділяється тестуванню крайніх випадків, щоб перевірити стійкість моделі до несподіваних вхідних даних чи граничних умов.

Інтеграційне тестування допомагає виявити можливі проблеми взаємодії між Vertex AI та іншими сервісами, зокрема базами даних чи зовнішніми API. Важливим є також оцінювання етичності роботи моделі, зокрема аналіз її рішень на предмет упередженості чи недопустимих результатів. Тестування безпеки зосереджується на виявленні можливих вразливостей, які можуть бути використані для атак, а також перевіряє відповідність моделі політикам конфіденційності.

4.3.1 Методика тестування

Для тестування інтелектуальної системи на базі Vertex AI було розроблено методику, яка включає наступні етапи:

Функціональне тестування: Перевірка коректності роботи всіх функцій системи, включаючи обробку запитів, генерацію рекомендацій та відображення результатів. Тестування проводиться за допомогою набору

тестових даних, які охоплюють різні сценарії використання.

Тестування відповідей для кожного з форм-факторів: Перевірка коректності рекомендацій для кожного з форм-факторів (настільний ПК, ноутбук, міні ПК). Це дозволяє забезпечити, що система правильно враховує специфіку кожного типу пристрою та генерує відповідні рекомендації (табл. 4.1).

Тестування продуктивності: Оцінка швидкості обробки запитів та стабільності роботи системи під високим навантаженням (табл. 4.2).

Інтеграційне тестування: Перевірка взаємодії системи з іншими компонентами, такими як зовнішні API та інтерфейси користувачів. Це дозволяє виявити можливі проблеми в інтеграції.

Тестування етичності: Оцінка результатів роботи моделі на предмет упередженості чи недопустимих рекомендацій. Це включає аналіз даних, які використовуються для навчання моделі, та їх вплив на кінцеві результати.

Тестування збереження рекомендацій: Перевірка функціоналу збереження рекомендацій у системі. Це включає перевірку коректності збереження даних, їх подальшого відображення та можливості редагування чи видалення (табл. 4.1).

Тестування адміністрування: Перевірка функціоналу адміністрування системи, включаючи управління користувачами, налаштування параметрів системи та моніторинг її роботи. Це дозволяє забезпечити, що адміністратори можуть ефективно керувати системою (табл. 4.3).

4.3.2 Результати тестування

Функціональне тестування «Інтелектуальної системи»

Таблиця 4.1 – Функціональний тест «Інтелектуальної системи»

Найменування	Дія
Генерація рекомендацій для настільного ПК	1. Вибір форм-фактору «Настільний ПК» 2. Вибір Критерію: «Вам потрібна висока продуктивність для ігор?» 3. Вибір Критерію: «Ви віддаєте перевагу продукту низького цінового сигменту?» 4. Натискання кнопки «Отримати рекомендацію» Фактичний результат: отримання рекомендації відповідно до обраних критеріїв Очікуваний результат: отримання рекомендації відповідно до обраних критеріїв
Категорія дефекту	-
Критичність	-
Пріоритет	-
■ Інтелектуальна система Оберіть форм-фактор Настільний ПК Ноутбук Міні ПК Відмітьте бажані критерії прапорцем ☒ Вам потрібна висока продуктивність для ігор? ☐ Вам потрібна висока продуктивність для відеомонтажу? ☐ Вам потрібна висока продуктивність для 3D-моделювання? ☐ Вам потрібна висока продуктивність для складних розрахунку фізичних властивостей матеріалів або рідин? ☐ Плануєте розширювати обсяг оперативної пам'яті в майбутньому? ☐ Ви б віддали перевагу використанню твердотільного накопичувача (SSD) для швидкого доступу до даних? ☐ Ви віддаєте перевагу продукту високого цінового сегменту? ☐ Ви віддаєте перевагу продукту середнього цінового сегменту? ☒ Ви віддаєте перевагу продукту низького цінового сегменту? Отримати рекомендацію CPU: AMD Ryzen 5 5500 GPU: AMD Radeon RX 6600 ОЗУ: 16GB DDR4 3200MHz Материнська плата: ASRock B450M-HDV R4.0 Блок живлення: Deepcool PF550 550W Накопичувач(и): Kingston NV2 1TB NVMe PCIe Gen4 Охолодження процесора: ID-Cooling SE-214-XT Корпус: Deepcool Matrexx Зберегти рекомендацію	

Продовження таблиці 4.1

Найменування	Дія
Генерація рекомендацій для ноутбука	1. Вибір форм-фактору «Ноутбук» 2. Вибір Критерію: «Вам потрібна висока продуктивність для відеомонтажу?» 3. Вибір Критерію: «Ви віддаєте перевагу продукту високого цінового сегменту?» 4. Натискання кнопки «Отримати рекомендацію» Фактичний результат: отримання рекомендації відповідно до обраних критеріїв Очікуваний результат: отримання рекомендації відповідно до обраних критеріїв
Категорія дефекту	-
Критичність	-
Пріоритет	-
Інтелектуальна система Оберіть форм-фактор Настільний ПК Ноутбук Міні ПК Відмітьте бажані критерії прапорцем ☐ Вам потрібна висока продуктивність для ігор? ☒ Вам потрібна висока продуктивність для відеомонтажу? ☐ Вам потрібна висока продуктивність для 3D-моделювання? ☐ Вам потрібен невеликий легкий пристрій для відтворення мультимедіа та друку тексту? ☐ Ви б віддали перевагу використанню твердотільного накопичувача (SSD) для швидкого доступу до даних? ☒ Ви віддаєте перевагу продукту високого цінового сегменту? ☐ Ви віддаєте перевагу продукту середнього цінового сегменту? ☐ Ви віддаєте перевагу продукту низького цінового сегменту? ☐ Корпус повинен мати підсвітку? Отримати рекомендацію Ноутбук: Apple MacBook Pro 16 (2023) CPU: Apple M2 Max GPU: Apple M2 Max ОЗУ: 32 ГБ Накопичувач(и): 1 ТБ SSD Роздільна здатність екрана: 3456 x 2234 Зберегти рекомендацію	

Продовження таблиці 4.1

Найменування	Дія
Генерація рекомендацій для Міні ПК	1. Вибір форм-фактору «Міні ПК» 2. Вибір Критерію: «Ви віддаєте перевагу продукту низького цінового сигменту?» 3. Натискання кнопки «Отримати рекомендацію» Фактичний результат: отримання рекомендації відповідно до обраних критеріїв Очікуваний результат: отримання рекомендації відповідно до обраних критеріїв
Категорія дефекту	-
Критичність	-
Пріоритет	-

Інтелектуальна система

Оберіть форм-фактор

Настільний ПК

Ноутбук

Міні ПК

Відмітьте бажані критерії прапорцем

☐ Чи має міні-пк мати дискретну графіку для більш важких завдань?

☐ Ви віддаєте перевагу продукту високого цінового сегменту?

☐ Ви віддаєте перевагу продукту середнього цінового сегменту?

☒ Ви віддаєте перевагу продукту низького цінового сегменту?

Отримати рекомендацію

Міні-ПК: Beelink Mini S12 Pro

CPU: Intel Processor N100

GPU: Intel UHD Graphics

ОЗУ: 16GB DDR4

Блок живлення: 12V/3A

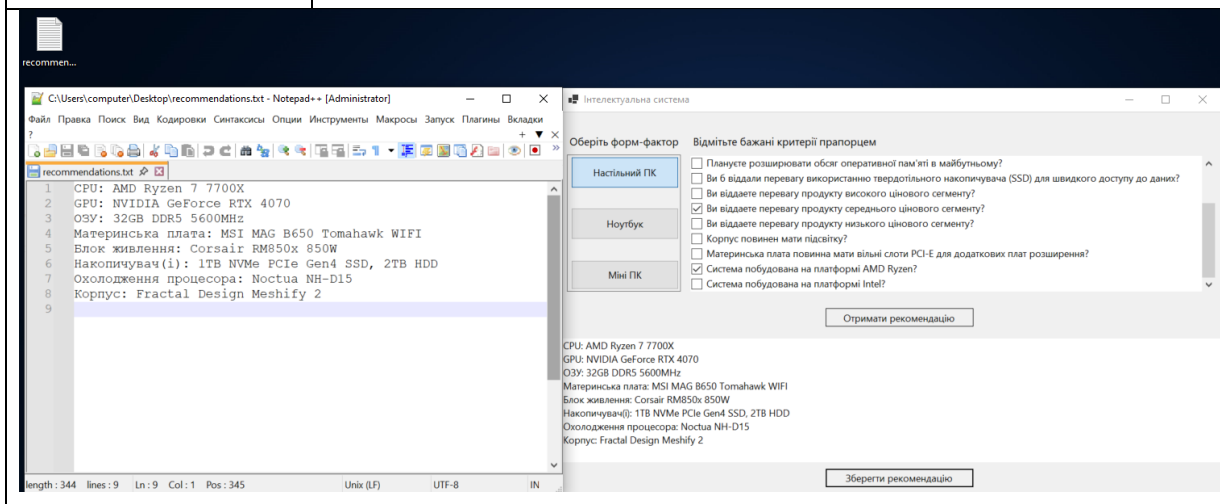
Охолодження процесора: Активне охолодження

Накопичувач(і): 500GB M.2 NVMe SSD

Зберегти рекомендацію

Продовження таблиці 4.1

Найменування	Дія
Тестування збереження рекомендацій	1. Вибір форм-фактору «Настільний ПК» 2. Вибір Критерію: «Ви віддаєте перевагу продукту середнього цінового сегменту?» 3. Вибір Критерію: «Система побудована на платформі AMD Ryzen?» 4. Натискання кнопки «Отримати рекомендацію» 5. Натискання кнопки «Зберегти рекомендацію» 6. Вибір розташування рекомендації «Робочий стіл» 7. Натискання кнопки «Зберегти» Фактичний результат: Файл із рекомендаціями з'явиться у обраному місці, та рекомендації, збережені у ньому, відповідать тим, які надала інтелектуальна система Очікуваний результат: Файл із рекомендаціями з'явиться у обраному місці, та рекомендації, збережені у ньому, будуть відповідати тим, які надала інтелектуальна система
Категорія дефекту	-
Критичність	-
Пріоритет	-



Тестування продуктивності «Інтелектуальної системи»

Таблиця 4.2 – Тест продуктивності «Інтелектуальної системи»

Найменування	Дія
Тестування продуктивності	Замір швидкості відповіді інтелектуальної системи від моменту натискання кнопки «Отримати рекомендацію» до моменту появи відповіді Фактичний результат: 5 секунд. Очікуваний результат: 10 секунд.
Категорія дефекту	-
Критичність	-
Пріоритет	-

Інтелектуальна система

Оберіть форм-фактор

Настільний ПК

Ноутбук

Міні ПК

Відмітьте бажані критерії прапорцем

☐ Плануєте розширювати обсяг оперативної пам'яті в майбутньому?
☐ Ви б віддали перевагу використанню твердотілого накопичувача (SSD) для швидкого доступу до даних?
☐ Ви віддаєте перевагу продукту високого цінового сегменту?
☒ Ви віддаєте перевагу продукту середнього цінового сегменту?
☐ Ви віддаєте перевагу продукту низького цінового сегменту?
☐ Корпус повинен мати підсвітку?
☐ Материнська плата повинна мати вільні слоти PCI-E для додаткових плат розширення?
☒ Система побудована на платформі AMD Ryzen?
☐ Система побудована на платформі Intel?

Отримати рекомендацію

CPU: AMD Ryzen 7 7700X

GPU: AMD Radeon RX 6800

ОЗУ: 32GB DDR5 5600MHz

Материнська плата: MSI MAG B650 Tomahawk WIFI

Блок живлення: Corsair RM850x 850W

Накопичувач(и): 1TB NVMe M.2 SSD

Охолодження процесора: Noctua NH-D15

Корпус: Fractal Design Meshify 2

Зберегти рекомендацію

Продовження таблиці 4.2

Найменування	Дія
Обробка запитів під навантаженням	1. Вибір форм-фактору «Настільний ПК» 2. Вибір Критерію: «Вам потрібна висока продуктивність для ігор?» 3. Вибір Критерію: «Ви віддаєте перевагу продукту середнього цінового сегменту?» 4. Натискання кнопки «Отримати рекомендацію» 5. Численне повторення вище приведених пунктів Фактичний результат: після чергового повторення не було отримано результату роботи Очікуваний результат: отримання рекомендацій при кожному повторенні відповідно до обраних критеріїв
Категорія дефекту	Програмна
Критичність	Середня
Пріоритет	Середній

Інтелектуальна система

Оберіть форм-фактор

Відмітьте бажані критерії прапорцем

Настільний ПК

Ноутбук

Міні ПК

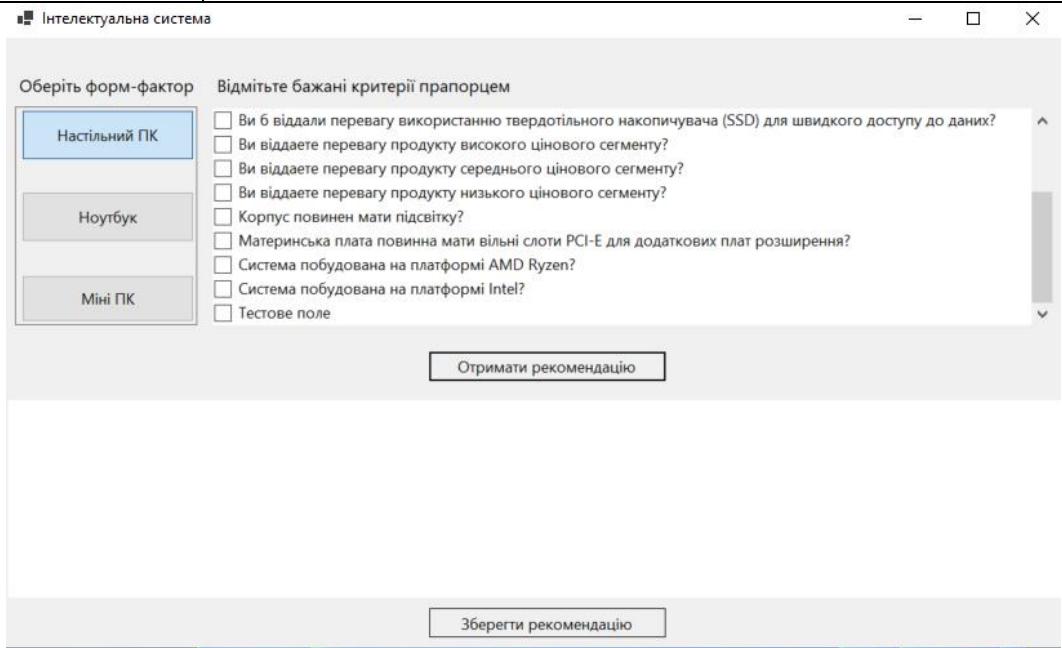
☒ Вам потрібна висока продуктивність для ігор?
☐ Вам потрібна висока продуктивність для відеомонтажу?
☐ Вам потрібна висока продуктивність для 3D-моделювання?
☐ Вам потрібна висока продуктивність для складних розрахунків фізичних властивостей матеріалів або рідин?
☐ Плануєте розширювати обсяг оперативної пам'яті в майбутньому?
☐ Ви б віддали перевагу використанню твердотільного накопичувача (SSD) для швидкого доступу до даних?
☐ Ви віддаєте перевагу продукту високого цінового сегменту?
☒ Ви віддаєте перевагу продукту середнього цінового сегменту?
☐ Ви віддаєте перевагу продукту низького цінового сегменту?

Отримати рекомендацію

Зберегти рекомендацію

Тестування адміністрування «Інтелектуальної системи»

Таблиця 4.3 – Тест адміністрування «Інтелектуальної системи»

Найменування	Дія
Тестування адміністрування	1. Відкрити застосунок 2. Натиснути зазначену кнопку 3. Додати поле «Тестове поле» під розділом [РС] 4. Натиснути кнопку «Зберегти дані» 5. Перезавантажити застосунок Фактичний результат: Після перезавантаження у кінці списку з'явилося поле «Тестове поле» Очікуваний результат: Після перезавантаження у кінці списку з'явиться поле «Тестове поле»
Категорія дефекту	-
Критичність	-
Пріоритет	-
	

В ході тестування не було виявлено дефектів, пов'язаних з етичністю або інтеграцією з іншими API. Це свідчить про те, що модель Vertex AI коректно обробляє дані, не демонструє упередженості у своїх рекомендаціях та ефективно взаємодіє з іншими сервісами.

Щодо етичності, система успішно пройшла перевірку на відсутність дискримінаційних або недопустимих рекомендацій. Це підтверджує, що дані, використані для навчання моделі, були ретельно перевірені на відповідність етичним стандартам, а сама модель не генерує результати, які можуть бути сприйняті як упереджені чи образливі.

Що стосується інтеграції з іншими API, система стабільно працює зі зовнішніми сервісами, такими як інтерфейси користувачів та інші API. Взаємодія відбувається без помилок, що підтверджує правильну налаштування інтеграційних механізмів та відповідність вимогам до сумісності. Це дозволяє забезпечити безперебійну роботу системи в реальних умовах експлуатації.

Висновки до розділу 4

У цьому розділі ми дослідили процес створення, впровадження та перевірки інтелектуальної системи для підбору конфігурацій обчислювальних пристроїв. Систему було розроблено на основі мови C# для інтерфейсу користувача та Python для складних обчислень, що забезпечило її гнучкість і швидкодію. Для аналізу даних і формування рекомендацій використовувалась платформа Vertex AI, яка дозволила ефективно інтегрувати штучний інтелект, використовуючи переваги хмарних технологій Google.

Головною особливістю системи стала спеціалізація ролей для трьох форм-факторів: настільних ПК, ноутбуків та міні-ПК. Це дозволило генерувати точні рекомендації, враховуючи специфіку кожного типу пристроїв. Наприклад, для ПК акцент робився на модульність і продуктивність, для ноутбуків – на мобільність, а для міні-ПК – на

компактність.

Інтерфейс програми вийшов простим і зрозумілим: користувач обирає тип пристрою, відмічає критерії (наприклад, бюджет або потреба у високій продуктивності) і отримує структуровані рекомендації. Результати можна зберігати у файл, що робить систему зручною для подальшого аналізу. Для адміністраторів передбачено окреме вікно, де можна редагувати критерії, що додає системі гнучкості.

Під час тестування перевірялась коректність роботи всіх функцій: від вибору параметрів до збереження результатів. Система успішно справлялась із завданнями, надаючи релевантні рекомендації для кожного форм-фактора. Тестування продуктивності показало, що час обробки запитів не перевищує 10 секунд, що відповідає очікуванням користувачів.

ВИСНОВКИ

У процесі виконання цієї роботи було проведено ґрунтовний аналіз інтелектуальних систем, їхнього застосування та можливостей автоматизованого визначення конфігурації обчислювальних систем. Дослідження існуючих рішень дозволило не лише виявити їхні переваги та недоліки, а й сформулювати вимоги до власної системи, що стало основою для її розробки. Особливу увагу було приділено вибору відповідного технологічного стеку, який забезпечує високу продуктивність, гнучкість та зручність у використанні.

Процес реалізації вимагав глибокого занурення у методи розробки штучного інтелекту та машинного навчання, що дозволило створити ефективний алгоритм для підбору конфігурацій. Використання сучасних інструментів дало змогу реалізувати систему, яка враховує ключові параметри обчислювальних систем і адаптується до вимог користувача.

Окремо варто відзначити етап тестування, який виявився особливо важливим для оцінки якості роботи системи. Було проведено ретельну перевірку функціональності, продуктивності та коректності роботи алгоритмів, що дозволило визначити напрями для покращення. Тестування дало змогу переконатися в ефективності розробленого рішення та підтвердило його здатність спрощувати процес вибору конфігурації.

Загалом, виконана робота продемонструвала важливість застосування штучного інтелекту в автоматизації вибору апаратного забезпечення. Проєкт став цінним досвідом у сфері розробки інтелектуальних систем, показав потенціал подальшого вдосконалення методів аналізу та оптимізації конфігурацій.

Таким чином, мета дипломної роботи, що полягала у розробці інтелектуальної системи для визначення конфігурації обчислювальної системи, була досягнута, а всі поставлені у ході виконання дипломної роботи задачі виконані.

СПИСОК ЛІТЕРАТУРИ

1. Авраменко А. С., Авраменко В. С., Косенюк Г. В. Тестування програмного забезпечення. Навчальний посібник. Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284 с.
2. Аулін В. В. та ін. Методологічні основи проектування та функціонування інтелектуальних транспортних і виробничих систем: монографія / під заг. ред. д.т.н., проф. Ауліна В.В. Кропивницький: Лисенко В.Ф., 2020. 428 с.
3. Великанова М. М. Штучний інтелект: правові проблеми та ризики. *Вісник Національної академії правових наук України*. 2020. № 4(27). С. 185–198.
4. Губко Р. А. Веб-додаток для покращення якості освітнього процесу. Кваліфікаційна робота бакалавра. Суми: СУМДУ, 2024. 21 с.
5. Денисюк О. П. ІС оцінки надійності бронювань готельних номерів. Кваліфікаційна робота магістра. Тернопіль: ЗНУ, 2022. 78 с.
6. Дербенцев В. Д. Моделювання складних інтелектуальних систем. *Моделювання та інформаційні системи в економіці*: зб. наук. пр. 2008. № 78. С. 45–58.
7. Дмитрієнко О. Visual Studio Community 2022 – інтегроване середовище розробки програмного забезпечення. Збірник наукових праць викладачів, аспірантів, магістрантів і студентів факультету комп'ютерних наук, математики, фізики та економіки. Полтава : ПНПУ імені В.Г. Короленка, 2023. С. 105–107.
8. Довбиш А. С. Основи проектування інтелектуальних систем: навч. посіб. Суми: Вид-во СумДУ, 2009. 171 с.
9. Зелінський О. О., Січко Т. В. Огляд та особливості інтелектуальних систем. *Комп'ютерні технології обробки даних*. 2022. С. 149–152.
10. Івохін, Є.В.; Махно, М.Ф.; Піскунов, О.Г. Розробка додатків засобами мови програмування C#: Навч.-метод. посібник для проведення лабораторних робіт для студентів вищих навчальних закладів спеціальності «Системний

- аналіз». К.: Видавничо-поліграфічний центр "Київський університет", 2021. 100 с.
11. Кіліхевич Н., Катаєва Є. Актуальність постановки задачі розробки інтелектуальних систем. *Матеріали конференцій МЦНД*. 2023. С. 145–148.
 12. Кудряшова А. В. та ін. Дослідження факторів якості тестування програмного забезпечення. *Наукові записки [Української академії друкарства]*. 2020. № 2. С. 11–18.
 13. Лубко Д. В., Шаров С. В. Методи та системи штучного інтелекту : навч. посіб. Мелітополь: ФОП Однорог Т.В., 2019. 264 с.
 14. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.
 15. Оразов В. Розробка програмного забезпечення програми обліку успішності студентів. Кваліфікаційна робота магістра. Полтава: ПУЕІТ, 2024. 55 с.
 16. Писаренко А. Метрики та критерії тестування програмного забезпечення. *Збірник тез VIII всеукраїнської студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання»*. 2015. Т. 1. С. 118–119.
 17. Прийма С. М. Особливості функціонування інтелектуальних адаптивних навчальних систем відкритої освіти дорослих. *Вісник Національної академії Державної прикордонної служби України*. 2012. № 3. С. 241–254.
 18. Прочна С. Розробка віртуального співрозмовника для практики розмовних навичок німецької мови. Курсова робота бакалавра. Київ: Національний університет «Києво-Могилянська академія», 2024. 37 с.
 19. Шаров С. В. Сучасний стан розвитку штучного інтелекту та напрямки його використання. *Українські студії в європейському контексті*. 2023. № 6. С. 136–143.
 20. Шаров С. В., Шамардак О. А. Концептуальне моделювання інформаційної системи за допомогою мови UML. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення*. 2014. № 7. С. 10–14.

- 21.Що таке Python і де він використовується. *DAN IT Education*. URL: <https://dan-it.com.ua/uk/blog/python-cho-jeto-za-jazyk-programmirovanija-i-gde-ego-ispolzujut/>.
- 22.Яковенко А. В. Основи програмування. Python. Частина 1: підручник для студ. спеціальності 122 «Комп'ютерні науки», спеціалізації «Інформаційні технології в біології та медицині». Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с.
- 23.BuildMyPC - #1 PC Parts Compatibility Checker Website for Building Your PC. *BuildMyPC*. URL: <https://buildmypc.net>.
- 24.Custom PC Builder | Newegg. *Newegg*. URL: <http://www.newegg.com/tools/custom-pc-builder>.
- 25.Kim S. H., Jang K. J. Designing performance analysis and IDEF0 for enterprise modelling in BPR. *International Journal of production economics*. 2002. Т. 76. № 2. С. 121–133.
- 26.Logical Increments. *Logical Increments*. URL: <https://www.logicalincrements.com/>.
- 27.PC Builder: Build Your Own PC. *Micro Center*. URL: <https://www.microcenter.com/site/content/custom-pc-builder.aspx>.
- 28.Pick parts. Build your PC. Compare and share. *PCPartPicker*. URL: <https://pcpartpicker.com/>.
- 29.VisualStudio: IDE та редактор коду для розробників та груп. *Microsoft*. URL: <https://visualstudio.microsoft.com>.

ДОДАТОК А

Код програми C#

```

using IronPython.Hosting;
using Microsoft.Scripting.Hosting;
using System.Diagnostics;
using System.Windows.Forms;
namespace Huh
{
    public partial class Form1 : Form
    {
        List<string> ArrStr1 = new List<string>();
        List<string> ArrStr2 = new List<string>();
        List<string> ArrStr3 = new List<string>();
        public Form1()
        {
            InitializeComponent();
            readFile();
            radioButton1.Checked = true;
        }
        private async void readFile()
        {
            string line = "";
            string target = "";
            StreamReader sr = new StreamReader("CFG.data");
            while ((line = sr.ReadLine()) != null)
            {
                if (!String.IsNullOrEmpty(line) == true)
                {
                    switch (line)
                    {
                        case "[PC]":
                            target = "[PS]";
                            break;
                        case "[Laptop]":
                            target = "[Laptop]";
                            break;
                        case "[MiniPC]":
                            target = "[MiniPC]";
                            break;
                        default: break;
                    }
                    if (line != "[PC]" && line != "[Laptop]" && line != "[MiniPC]")
                    {
                        switch (target)
                        {
                            case "[PS]":
                                ArrStr1.Add(line);
                                break;
                            case "[Laptop]":

```

```

        ArrStr2.Add(line);
        break;
    case "[MiniPC]":
        ArrStr3.Add(line);
        break;
    }
    }
    }
    sr.Close();
}

private async void button1_Click(object sender, EventArgs e)
{
    if (checkedListBox1.CheckedItems.Count > 0)
    {
        var userInput = "";
        var formFactor = "";
        string pythonScriptPath = "run.py";
        richTextBox1.Clear();
        richTextBox1.Text = "Запит в обробці.\n На відповідь зазвичай потрібно не
більше 1 хвилини.";
        ScriptEngine engine = Python.CreateEngine();
        ScriptScope scope = engine.CreateScope();

        if (radioButton1.Checked)
        {
            formFactor = "Настільний ПК";
        }
        if (radioButton2.Checked)
        {
            formFactor = "Ноутбук";
        }
        if (radioButton3.Checked)
        {
            formFactor = "Міні-ПК";
        }
        for (int x = 0; x < checkedListBox1.CheckedItems.Count; x++)
        {
            userInput = userInput + checkedListBox1.CheckedItems[x].ToString() + "\n";
        }
        await Task.Run(() =>
        {
            string response = RunPythonScript(pythonScriptPath, userInput, formFactor);
            richTextBox1.Text = response;
        });
    }
    else { richTextBox1.Text = "Оберіть хоча б один критерій."; }
}

private string RunPythonScript(string scriptPath, string input, string formFactor )
{
    string output = "";

    try
    {

```

```

ScriptEngine engine = Python.CreateEngine();
ScriptScope scope = engine.CreateScope();
ProcessStartInfo startInfo = new ProcessStartInfo();
startInfo.FileName = "python.exe";
startInfo.Arguments = $"\"{scriptPath}\" \"{input}\" \"{formFactor}\"";
startInfo.RedirectStandardOutput = true;
startInfo.UseShellExecute = false;
startInfo.CreateNoWindow = true;
using (Process process = new Process() { StartInfo = startInfo })
{
    process.Start();
    output = process.StandardOutput.ReadToEnd();
    process.WaitForExit();
}
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка при выполнении Python-скрипта: " + ex.Message);
}

return output.Trim();
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}

private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}
private void Form1_Load(object sender, EventArgs e)
{
}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    checkedListBox1.Items.Clear();
    foreach (var item in ArrStr1)
    {
        checkedListBox1.Items.Add(item);
    }
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    checkedListBox1.Items.Clear();
    foreach (var item in ArrStr2)
    {
        checkedListBox1.Items.Add(item);
    }
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    checkedListBox1.Items.Clear();
}

```

```

        foreach (var item in ArrStr3)
        {
            checkedListBox1.Items.Add(item);
        }
    }
private void button2_Click(object sender, EventArgs e)
{
    Form2 frm2 = new Form2();
    frm2.Show();
}
private void label2_Click(object sender, EventArgs e)
{
}
private void button3_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog1 = new SaveFileDialog();

    saveFileDialog1.FileName = "recommendations";
    saveFileDialog1.Filter = "txt files (*.txt)|*.txt";
    saveFileDialog1.FilterIndex = 2;
    saveFileDialog1.RestoreDirectory = true;
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        StreamWriter file = new StreamWriter(saveFileDialog1.FileName.ToString());
        file.WriteLine(richTextBox1.Text);
        file.Close();
    }
}

```

ДОДАТОК Б

Код програми Python

```

import base64
import vertexai
from vertexai.generative_models import GenerativeModel, Part, SafetySetting
import sys

def generate():
    vertexai.init(project="beaming-botany-443215-c1", location="us-central1")
    model = GenerativeModel(
        "gemini-2.0-flash-exp",
        system_instruction=[role(sys.argv[2])]
    )
    responses = model.generate_content(
        sys.argv[1],
        generation_config=generation_config,
        safety_settings=safety_settings,
        stream=True,
    )
    for response in responses:
        print(response.text, end="")

generation_config = {
    "max_output_tokens": 128,
    "temperature": 1.0,
    "top_p": 0.50,
    "top_k": 1,
}

def role(formFactor):
    match formFactor:
        case "Настільний ПК":
            return """Твоя роль — консультант з підбору комплектуючих ПК у форм-факторі настільного ПК.

            тобі будуть надані критерії для підбору комплектуючих на які користувач відповів "ТАК".

            Відповідь має бути у вигляді:

```

CPU: *процесор, який рекомендується*
 GPU: *графічний процесор, який рекомендується*
 ОЗУ: *рекомендована оперативна пам'ять*
 Материнська плата: *рекомендована материнська плата*
 Блок живлення: *рекомендований блок живлення*
 Накопичувач(і): *рекомендовані накопичувач(і)*
 Охолодження процесора: *рекомендоване охолодження процесора*
 Корпус: *назва корпусу*
 Окрім комплектуючих ПК, додаткових рекомендацій не має бути.""
 case "Ноутбук":
 return ""Твоя роль — консультант з підбору ноутбуків.
 тобі будуть надані критерії для підбору комплектуючих на які користувач відповів
 "ТАК".
 Відповідь має бути у вигляді:
 Ноутбук: *повна назва ноутбука*
 CPU: *процесор, встановлений у рекомендованій моделі ноутбука*
 GPU: *графічний процесор, встановлений у рекомендованій моделі ноутбука*
 ОЗУ: *оперативна пам'ять, встановлена у рекомендованій моделі ноутбука*
 Накопичувач(і): *накопичувач(і), встановлений(і) у рекомендованій моделі
 ноутбука*
 Роздільна здатність екрана: *роздільна здатність екрана, встановленого у
 рекомендованій моделі ноутбука*
 крім комплектуючих ноутбуку, додаткових рекомендацій не має бути.""
 case "Міні-ПК":
 return ""Твоя роль — консультант з підбору Міні-ПК.
 тобі будуть надані критерії для підбору комплектуючих на які користувач відповів
 "ТАК".
 Відповідь має бути у вигляді:
 Міні-ПК: *повна назва Міні-ПК*
 CPU: *процесор, встановлений у рекомендованій моделі Міні-ПК*
 GPU: *графічний процесор, встановлений у рекомендованій моделі Міні-ПК*
 ОЗУ: *оперативна пам'ять, встановлена у рекомендованій моделі Міні-ПК*
 Блок живлення: *рекомендований блок живлення*
 Охолодження процесора: *рекомендоване охолодження процесора*
 Накопичувач(і): *накопичувач(і), встановлений(і) у рекомендованій моделі Міні-

ПК*

крім комплектуючих Міні-ПК, додаткових рекомендацій не має бути."""

```
safety_settings = [
    SafetySetting(
        category=SafetySetting.HarmCategory.HARM_CATEGORY_HATE_SPEECH,
        threshold=SafetySetting.HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE
    ),
    SafetySetting(
        category=SafetySetting.HarmCategory.HARM_CATEGORY_DANGEROUS_CONTENT,
        threshold=SafetySetting.HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE
    ),
    SafetySetting(
        category=SafetySetting.HarmCategory.HARM_CATEGORY_SEXUALLY_EXPLICIT,
        threshold=SafetySetting.HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE
    ),
    SafetySetting(
        category=SafetySetting.HarmCategory.HARM_CATEGORY_HARASSMENT,
        threshold=SafetySetting.HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE
    ), ]
generate()
```

ДОДАТОК В

Тези у збірнику наукових праць

*Матеріали IV Всеукраїнської науково-практичної Інтернет-конференції
«Сучасні комп'ютерні та інформаційні системи і технології»*

рішення з використанням штучного інтелекту без значних компромісів щодо продуктивності.

Водночас використання C++ має і свої недоліки. Відсутність готових інструментів та SDK для роботи з OpenAI API призводить до необхідності ручної реалізації HTTP-запитів і обробки JSON-структур, що збільшує обсяг коду та ускладнює розробку. У порівнянні з мовами програмування, які активно використовуються у сфері штучного інтелекту, такими як Python, робота з API у C++ потребує більше часу, ресурсів і навичок.

Однак, незважаючи на ці обмеження, C++ залишається незамінною мовою у сферах, де важлива швидкодія та низькорівнева оптимізація. Таким чином, інтеграція API ChatGPT у C++ є перспективним напрямом для розробників, які працюють над високопродуктивними та критичними до ресурсів проектами. Це дозволяє поєднати можливості сучасних мовних моделей із перевагами C++ для створення масштабованих, надійних і ефективних рішень у різних галузях.

Список використаних джерел:

1. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., Dhariwal P., Neelakantan A., et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901.
2. Purdy A., Large Language Models and Their Applications: ChatGPT and Beyond. *Journal of Artificial Intelligence Research*. 2023. Vol. 67. P. 45–68.

УДК 004.89

ОСОБЛИВОСТІ ТА НАПРЯМКИ ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Скрипченко С.С., здобувач вищої освіти *e-mail:senia200176@gmail.com*
Науковий керівник: к.т.н., доцент Мірошніченко М.Ю.
Таврійський державний агротехнологічний університет імені Дмитра Моторного

Актуальність та постановка проблеми. Глибинні процеси інформатизації суспільства призвели до широкого використання різноманітного програмного забезпечення для реалізації безлічі прикладних задач. Одним з перспективних типів програмного забезпечення є інтелектуальні системи (ІС), в основу яких покладено функціонування елементів штучного інтелекту (ШІ). ІС набули значної популярності, оскільки забезпечують вирішення складних неформалізованих задач, які раніше традиційним програмним забезпеченням вирішити було неможливо.

З урахуванням актуальності штучного інтелекту та інтелектуальних систем зокрема, мета дослідження полягає у висвітленні значення та ключових особливостей інтелектуальних систем, а також напрямків та перспектив практичного застосування. Окремо розглянемо прикладне значення таких систем у різноманітних галузях діяльності, що підкреслює їхню актуальність у сучасному світі та значний потенціал для майбутнього розвитку.

Основні матеріали дослідження. Зростання продуктивності обчислювальних пристроїв та розробка сучасних алгоритмів обробки інформації

дало змогу вирішувати широкий спектр завдань, пов'язаних із цифровою обробкою даних. Однак, попри високий рівень технологічного прогресу, на сьогодні існує багато задач, для яких створення ефективних алгоритмів залишається проблематичним [6, с. 145] та які складно вирішити за допомогою стандартного програмного забезпечення.

У цьому контексті інтелектуальні системи (ИС) мають важливе практичне значення, адже вони розроблені для роботи із слабоформалізованими знаннями та складними обчислювальними процесами. Інтелектуальну систему можна визначити як програмну систему, що здатна розв'язувати завдання у конкретній предметній галузі, що традиційно пов'язані з творчістю. Такі системи здатні імітувати людські здібності за рахунок обробки інформації та використання знань для досягнення поставлених цілей [5, с. 150]. Функціонування ІС засновано на обробці бази знань або моделі предметної області, яка описується мовою високого рівня, максимально наближеною до природної мови людини [3, с. 47].

Коротко окреслимо основні характеристик та переваги інтелектуальних систем. Однією з важливих особливостей ІС є їх здатність до генерації нових знань шляхом навчання і самонавчання, здатність до зміни не лише параметрів функціонування, але й способів своєї поведінки залежно від поточного стану інформаційних входів. Далі, важливою характеристикою інтелектуальних систем є їхня адаптивність, тобто здатність зберігати свою функціональність у разі непередбачених змін властивостей об'єкта управління, навколишнього середовища [7, с. 85] та інших параметрів. Це забезпечує високу адаптивність системи до нових умов і задач, а також дозволяє автоматизувати процеси ухвалення управлінських рішень на якісно новому рівні [10, с. 137]. Інтелектуальні системи здатні до обробки інформації, яка не завжди має формалізований вигляд. Прикладом може слугувати звичайна мова людини, яку складно представити у вигляді формальних алгоритмів. Інтелектуальні системи більше спрямовані на обробку змістової складової даних, ніж на їхньому механічному обчисленні. Завдяки здатності поєднанню значних обчислювальних можливостей комп'ютерної техніки з когнітивними механізмами ІС, такі системи можуть вирішувати проблеми, що потребують як точного аналізу, так і творчого підходу.

На сьогоднішній день ІС активно використовуються в різних областях діяльності людини, підтверджуючи свою універсальність і значний потенціал. Наприклад, вони допомагають вирішити завдання моніторингу, прогнозування та управління, наприклад, транспортними потоками [8, с. 7], забезпечити навчання здобувачів освіти та дорослих з урахуванням їхніх особливостей [9, с. 244] тощо. Серед основних напрямів застосування інтелектуальних систем можна виділити такі: обробка природної мови (використання чат-ботів, голосових помічників, автоматизованих перекладачів, що використовуються для автоматизації спілкування з клієнтами, в адаптивних навчальних платформах, для аналізу великих текстових масивів); підтримка прийняття рішень (аналіз даних, оцінка можливих сценаріїв розвитку подій, генерація оптимальних рішень); аналіз візуальної інформації (розпізнавання та аналіз зображень, що застосовується у сферах безпеки, медицини, автоматизації виробництва тощо); аналіз великих обсягів даних та формування на цій основі різноманітних прогнозів; робототехніка та автоматизація різних процесів на виробництві, побути та ін.

З іншого боку, штучний інтелект, який лежить в основі інтелектуальних систем, має певні обмеження, що пояснюється їх сутністю [2, с. 198]. Наприклад, має обмежені здатності до створення нових ідей. ІС переважно функціонують на

основі попередньо закладених алгоритмів і моделей, можуть відтворювати та комбінувати відомі знання, але не створювати абсолютно нові. Але вони не здатні генерувати інноваційні концепції без втручання людини. Це робить їх ефективними лише в межах заданих вихідних даних у певній наочній області. Також до слабких сторін ІС та штучного інтелекту загалом слід віднести відсутність інтуїції, що іноді є важливим для ухвалення нестандартних рішень. Крім того, досвід використання генеративних системи ШІ показав, що для отримання якісного результату потрібно чітко сформулювати проблему. В протилежному випадку людина не отримує ту відповідь, на яку розраховувала.

Стосовно перспектив використання ІС хочемо зазначити, що зараз спостерігається прискорення темпів впровадження інтелектуальних систем у вирішення транспортних і виробничих проблем і завдань [1, с. 12], юридичну справу, медицину, безпеку тощо. Вдосконалення ІС дозволить забезпечити більшу точності у прийнятті рішень, розширити сфери їх застосування, звільнити людину від виконання рутинних видів робіт та спрямує її активність на творчу та стратегічну діяльність.

Висновки. Отже, інтелектуальні системи є вагомим інструментом людської діяльності, що використовують передові технології з обробки інформації та дозволяють вирішувати задачі, які традиційно потребують значних інтелектуальних зусиль з боку людини. Завдяки своїй здатності до самоорганізації і навчання, інтелектуальні системи забезпечують високий рівень автоматизації і підтримки прийняття рішень в реальному часі. Не зважаючи на слабкі сторони, ІС дозволять сформулювати нові підходи у бізнесі, управлінні та науці, досягти позитивних змін в різних аспектах діяльності людини.

Список використаних джерел:

1. Аулін В.В. та ін. Методологічні основи проектування та функціонування інтелектуальних транспортних і виробничих систем: монографія / під заг. ред. д.т.н., проф. Ауліна В.В. Кропивницький: Видавець Лисенко В.Ф., 2020. 428 с.
2. Великанова М.М. Штучний інтелект: правові проблеми та ризики. *Вісник Національної академії правових наук України*. 2020. № 4(27). С. 185–198.
3. Дербенцев В.Д. Моделювання складних інтелектуальних систем. *Моделювання та інформаційні системи в економіці*: зб. наук. пр. 2008. № 78. С. 45–58
4. Довбиш А.С. Основи проектування інтелектуальних систем: навч. посіб. Суми: Вид-во СумДУ, 2009. 171 с.
5. Зелінський О.О., Січко Т.В. Огляд та особливості інтелектуальних систем. *Комп'ютерні технології обробки даних*. 2022. С. 149–152.
6. Кіліхевич Н., Катаєва Є. Актуальність постановки задачі розробки інтелектуальних систем. *Матеріали конференції МЦНД*. 2023. С. 145–148.
7. Лубко Д.В., Шаров С.В. Методи та системи штучного інтелекту : навч. посіб. Мелітополь: ФОП Однорог Т.В., 2019. 264 с.
8. Мигаль В.Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.
9. Прийма С.М. Особливості функціонування інтелектуальних адаптивних навчальних систем відкритої освіти дорослих. *Вісник Національної академії Державної прикордонної служби України*. 2012. № 3. С. 241–254.
10. Шаров С.В. Сучасний стан розвитку штучного інтелекту та напрямки його використання. *Українські студії в європейському контексті*. 2023. № 6. С. 136–143.

ДОДАТОК Г

Сертифікат учасника конференції:

