

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб для віддаленого моніторингу та конфігурації
ресурсів комп'ютера

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»


Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 022082.22.02.67 ПЗ

Виконав здобувач IV курсу, група КІ2-22-2


Підпис

Ігор СЕНЧИШИН

Ініціали, прізвище

Керівник канд.тех.наук, доцент
Науковий ступінь, учене звання


Підпис

Микола ФЕДУЛА

Ініціали, прізвище


Нормоконтролер канд.фіз.-мат.наук, доцент
Науковий ступінь, учене звання


Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«01» червня 2026 р.

дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

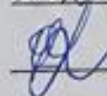
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІПС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Сенчишину Ігорю Ігоровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера

Керівник проекту (роботи) Федула Микола Васильович, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Дослідження предметної області та постановка задачі

Проектування програмно-технічного засобу “HELPERPC”

Реалізація та тестування програмно-технічного засобу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Блок-схеми алгоритмів програмно-технічного засобу

Тестування застосунок на різних ПК

Схема програмно-технічного засобу HELPERPC


6. Консультанти розділів кваліфікаційної роботи

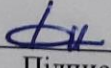
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1, а саме дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2, вибір методів проєктування програмно-технічного засобу	01.04.2026	виконано
5	Робота над розділом 3, проєктування програмно-технічного засобу	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Підпис Ігор СЕНЧИШИН
Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи  Підпис Микола ФЕДУЛА
Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера».

Автор роботи: Сенчишин Ігор Ігорович.

Керівник роботи: Федула Микола Васильович.

Пояснювальна записка: 73 с., 23 рис., 9 табл., 3 дод., 80 джерел.

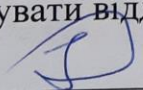
Графічна частина: 4 креслення.

АНАЛІЗ ПАРАМЕТРІВ, АРХІТЕКТУРА ПК, ВЕБ-ІНТЕРФЕЙС,
МОНІТОРИНГ, ПЕРСОНАЛЬНИЙ КОМП'ЮТЕР, ФОРМУВАННЯ
КОНФІГУРАЦІЇ

Метою дипломної роботи є розробка веб-системи для аналізу технічних параметрів персонального комп'ютера та формування оптимальної конфігурації апаратного забезпечення з урахуванням показників продуктивності, та сумісності компонентів.

Об'єктом дослідження є процес функціонування апаратної частини персонального комп'ютера та взаємодія її компонентів у процесі виконання обчислювальних задач. Розроблено програмно-технічний засіб для збору, аналізу та обробки технічних параметрів ПК, а також формування конфігурацій фізичних компонентів системи.

Під час виконання роботи було використано методи аналізу різних літературних джерел та документацій, а також методи розробки веб-системи, яка буде виконувати віддалений моніторинг та аналіз параметрів ПК.






Підпис здобувача

30.05.2026

Дата

ЗМІСТ

Вступ.....	4
1 Дослідження предметної області та постановка задачі.....	7
1.1 Аналіз предметної області і виявлення наявних проблем і завдань	7
1.2 Аналіз існуючих рішень, а також недоліків і переваг	8
1.3 Головні проблеми апаратної частини ПК	15
1.4 План реалізації	17
1.5 Висновки до першого розділу	19
2 Проєктування програмно-технічного засобу “HELPERPC”	22
2.1 Архітектура програмно-технічного засобу та модель взаємодії модулів	22
2.2 Формули розрахунку параметрів ПК	25
2.4 Інтелектуальний метод прийняття рішень та формування рекомендацій для конфігурації комп’ютера	32
2.5 Джерела і архітектура отримання апаратної інформації	35
2.6 Блок-схеми алгоритмів роботи	39
2.7 Висновки до другого розділу.....	45
3 Реалізація та тестування програмно-технічного засобу.....	46
3.1 Загальна характеристика реалізованого засобу	46
3.2 Програмна реалізація ключових модулів.....	51
3.2.1 Модуль ініціалізації та організації міжпроцесної взаємодії	52
3.2.2 Модуль апаратного сканування та термального моніторингу.....	53
3.2.3 Модуль інтелектуального аналізу та формування рекомендацій	54
3.2.4 Модуль інтерфейсу користувача та представлення даних	55

КвРКІ 022082.22.02.67 ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата
Виконав		Ігор СЕНЧИШИН		
Перевір.		Микола ФЕДУЛА		
Н.контр.		Тетяна КИСІЛЬ		
Затвер.		Ольга ПАВЛОВА		
Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп’ютера				
		Літера	Аркуш	Аркушів
		у	2	94
ХНУ КІ2-22-2				

ВСТУП

Актуальність дослідження зумовлена стрімким поширенням систем, які поєднують у собі обчислювальні засоби, програмне забезпечення та фізичні компоненти у єдине середовище. Подібні рішення функціонують у сферах розумної інфраструктури, автоматизованого виробництва, цифрових сервісів. Одним із різновидів таких систем є програмно-технічні системи, орієнтовані на сканування, оптимізацію та аналіз роботи персональних комп'ютерів. Вони поєднують засоби збору даних, обробку і формування рекомендацій на основі отриманої інформації.

Метою дипломної роботи є обґрунтування умов і особливостей побудови програмно-технічного засобу для комплексної оцінки технічного і програмного стану персонального комп'ютера, а також дослідження механізмів обробки системної інформації у межах даного підходу з метою підвищення ефективності виявлення проблем і обмежень у продуктивності, також головним критерієм є формування обґрунтованих рекомендації щодо оптимізації.

Для досягнення поставленої мети у роботі необхідно вирішити такі завдання:

1. Проаналізувати сучасні програмні засоби для діагностики та моніторингу персональних комп'ютерів і визначити їх основні функціональні можливості.
2. Дослідити методи отримання апаратної інформації з операційної системи та апаратних інтерфейсів.
3. Розробити архітектуру програмного засобу для збору, обробки та нормалізації даних про апаратне забезпечення ПК.
4. Реалізувати модуль збору системної інформації та модуль аналізу параметрів продуктивності й температурного стану компонентів.
5. Розробити алгоритми оцінювання стану системи та формування рекомендацій щодо оптимізації та модернізації ПК.

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

6. Провести тестування програмного засобу з використанням схожих утиліт та виконати аналіз точності отриманих результатів.

7. Оцінити якість роботи програмно-технічного засобу і сформувати таблиці із результатами.

Об'єктом дослідження виступає процес функціонування апаратних компонентів персонального комп'ютера в умовах реальної експлуатації та їх взаємодія з програмними засобами моніторингу та аналізу.

Предмет дослідження охоплює методи й алгоритми збору, структурування та використання системних даних у механізмі рекомендації, а також у складі програмно-технічного засобу.

У межах аналізу предметної області здійснено дослідження сучасних підходів до оцінювання технічного стану обчислювальних систем. Було виявлено, що наявні програми переважно орієнтовані на моніторинг окремих компонентів, зокрема центрального процесора або графічного адаптера. Також частина із них обмежується відображенням базових характеристик без глибокої аналітичної обробки. Більша частина із доступних додатків працюють за відсутності інтегрованих рекомендацій на базі отриманих показників, що знижує ефективність використання таких інструментів у навчальному та дослідницькому середовищі. Окремі рішення потребують постійного мережевого з'єднання або складного налаштування, що обмежує можливість автономного використання.

Проведений аналіз дозволив окреслити коло завдань, реалізація яких передбачається у створюваному програмно-технічному засобі. Заплановано автоматизований збір відомостей про ключові апаратні складові, включаючи процесор, графічний адаптер, оперативну пам'ять, накопичувачі та інші системні параметри. Отримана інформація підлягатиме впорядкуванню у структурований формат, придатний для подальшої аналітичної обробки. Це створює умови для швидкого визначення обмежувальних факторів продуктивності, виявлення

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

дисбалансу конфігурації та обґрунтування доцільності модернізації окремих елементів.

Особлива увага приділяється розробці механізмів показу і аналізу результатів, що забезпечують формування рекомендацій щодо оптимізації та підвищення стабільності системи. Передбачено реалізацію представлення показників у вигляді карток, графічних залежностей і числових індикаторів, що сприятиме підвищенню наочності та зручності аналізу. Додатково планується впровадження локального збереження історії перевірок для подальшого застосування у відстеженні змін конфігурації і її швидкодії.

Концепція програмно-технічного засобу передбачає автономний режим роботи без обов'язкового підключення до віддалених серверів, а також модульну структуру програмної реалізації, що створює умови для подальшого розширення функціональних можливостей. Рекомендаційна підсистема може функціонувати як у локальному режимі із використанням вбудованих алгоритмів аналізу, так і з можливістю передавання узагальнених даних на спеціалізований сервер для отримання розширених висновків. У зв'язку з цим перспективним напрямом є поєднання функцій автоматичного збору системної інформації з механізмами інтелектуального аналізу, що дозволяють формувати узагальнені висновки щодо стабільності роботи ПК, рівня навантаження компонентів і можливих причин зниження продуктивності.

Узагальнення результатів аналізу предметної області свідчить про доцільність розроблення програмно-технічного засобу HELPERPC як сучасного інструменту оцінки стану ПК. Реалізація автоматизованого збору системної інформації, інтегрованої аналітики, наочного відображення показників, механізмів формування рекомендацій та підтримки історії досліджень формує хорошу основу для підвищення ефективності апаратними і програмними ресурсами й їх раціональним використанням.

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Дослідження предметної області охоплює аналіз сучасних методів контролю та оцінювання технічного стану персональних комп'ютерів, а також узагальнення накопичених проблем, які виникають у процесі використання, експлуатації апаратних і програмних ресурсів. У результаті опрацювання наявних рішень встановлено, що значна частина інструментів орієнтована на спостереження за окремими складовими системи, зокрема за процесором [4-6], або графічним адаптером [3,7], без формування загального уявлення про взаємодію компонентів. Інші програмні продукти обмежуються базовими характеристиками, не забезпечуючи глибокого аналізу отриманих показників. За таких умов аналіз результатів ускладнюється, а визначення причин зниження продуктивності або нестабільної роботи потребує спеціаліста, а також додаткових базових знань. Додатковим обмежувальним фактором є залежність програм від постійного мережевого з'єднання або необхідність складного налаштування, що знижує доцільність застосування у дослідницькій, навчальній та аналітичній діяльності.

Поглиблений аналіз дозволив сформулювати сукупність функціональних вимог до програмно-технічного засобу. Передбачається реалізація автоматичного зчитування відомостей про апаратні і програмні характеристики пристрою, серед яких процесор, графічний адаптер, оперативна пам'ять, жорсткий диск та інші важливі параметри, що характеризують стан ПК. Отримані дані підлягатимуть структурованому поданню з метою подальшої обробки, порівняння та узагальнення. Такий підхід створює умови для виявлення дисбалансу конфігурації, перевантаження окремих елементів, нераціонального використання ресурсів і потенційних напрямів модернізації [3-7].

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Також головним елементом є застосування інтелектуальних алгоритмів аналізу, здатних формувати обґрунтовані рекомендації щодо покращення швидкодії, стабільності та ефективності функціонування системи. Результати планується відображати у зручному форматі з використанням графіків і карток параметрів, що сприятиме легкому і швидкому сприйняттю та аналізу висновків по системі. Передбачається також збереження останньої перевірки у локальному сховищі браузера, або системи (по бажанню користувача).

Модель програмно-технічного засобу базується на принципах модульності, та автономності. Передбачається, щоб додаток міг працювати без зовнішнього сервера, якщо так потрібно. Це забезпечить стабільність і конфіденційність. Архітектура системи орієнтується на можливість розширення шляхом додавання потрібних аналітичних модулів і вдосконалення під них алгоритмів обробки інформації. Рекомендаційна підсистема може здійснювати оцінювання у локальному режимі, а за потреби підтримувати зв'язок із віддаленим сервером для отримання розширених висновків [8-10].

Результати дослідження підтверджують доцільність створення HELPERPC як комплексного інструменту для оцінювання стану ПК. Реалізація автоматичного збору системних даних, забезпечення деталізованих і зрозумілих результатів формують основу для ефективного дослідження та керування ресурсами сучасного персонального комп'ютера. Отримані результати будуть використані як основа для подальшої розробки програмно-технічного засобу.

1.2 Аналіз існуючих рішень, а також недоліків і переваг

У сучасному середовищі програмних забезпечень представлено значну кількість інструментів, призначених для тестування, контролю, або оптимізації комп'ютера. Незважаючи на це, більшість із них реалізує обмежений набір функцій, або орієнтується на вирішення окремих завдань без оцінювання загального стану ПК.

					КвРКІ 022082.22.02.67 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

Прикладом такого рішення є NVIDIA GeForce Experience, що здійснює аналіз конфігурації та автоматично підбирає потрібні параметри для різних застосунків. Програма характеризується швидкістю роботи та зручністю, проте функціонує виключно з відеокартами NVIDIA та зазвичай орієнтується на ігровий процес. Питання модернізації апаратної частини, або комплексного аналізу системи залишаються поза межами можливості даного застосунку [7,12].

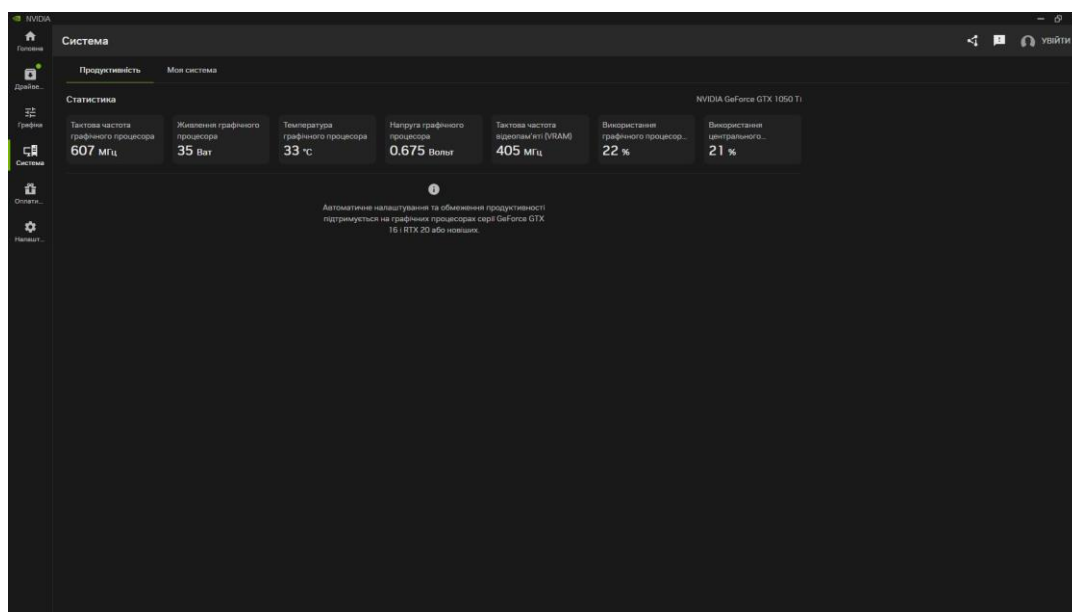


Рисунок 1.1 – NVIDIA GeForce Experience

Сервіс UserBenchmark має змогу тестувати центральний процесор, графічний адаптер, оперативну пам'ять та накопичувачі із можливістю порівняння результатів із іншими ПК. Сильними сторонами є деталізована статистика та формування рейтингу продуктивності. Але із іншої сторони акцент робиться тільки на вимірювані показники, тоді як механізми аналізу та персоналізовані рекомендації щодо вдосконалення не реалізовані. Крім цього, сервіс орієнтований переважно на онлайн-використання та залежить від підключення до мережі, що обмежує можливість автономного аналізу стану комп'ютерної системи [13].

Одним із професійних інструментів є AIDA64, який надає розширені засоби тестування, зокрема вимірювання напруг на різних компонентах ПК, проведення стресових випробувань та моніторинг температурних режимів. Висока точність даних поєднується із складною структурою інтерфейсу, що ускладнює використання непідготованих користувачів. Функція автоматичного формування рекомендацій відсутня, що знижує можливості аналізу результатів [30].

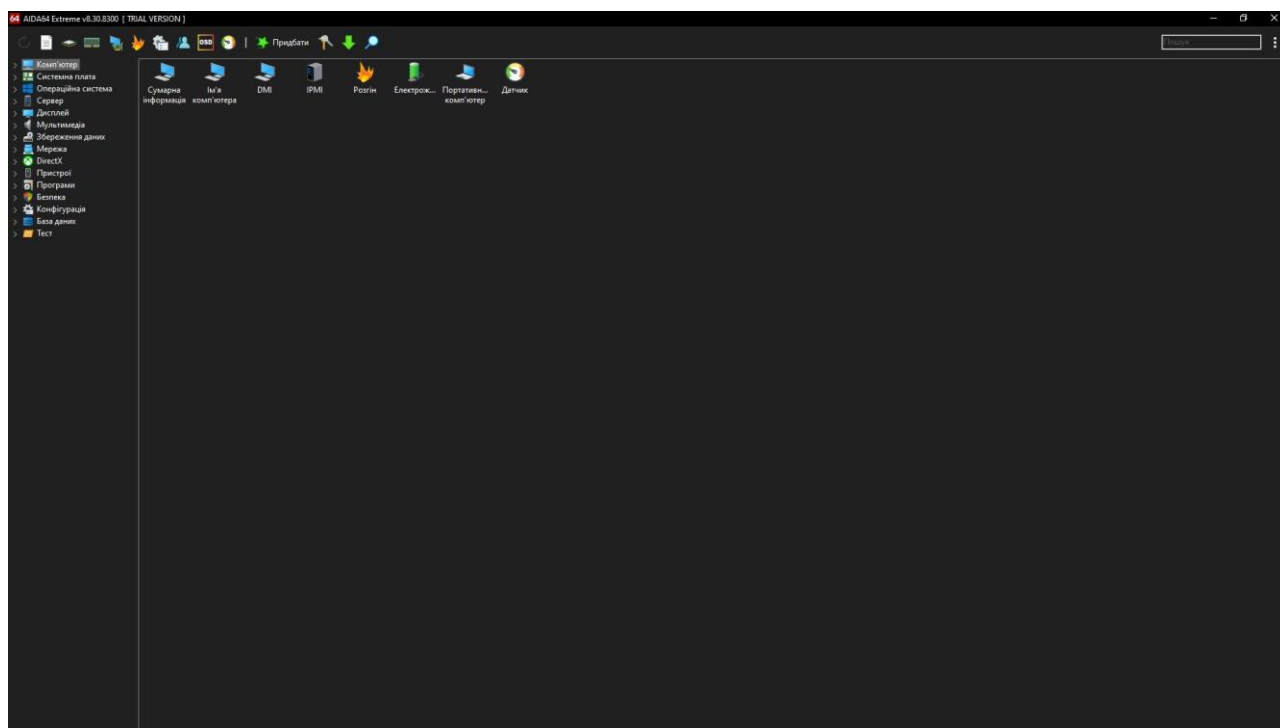


Рисунок 1.4 – AIDA64

Діагностична утиліта CPU-Z [2] відображає технічні характеристики процесора, материнської плати та оперативної пам'яті з високою точністю й мінімальним навантаженням на систему. Однак програма обмежується інформативним поданням параметрів без аналізу або побудови прогнозів продуктивності. Основною перевагою CPU-Z є швидке отримання детальної інформації про апаратну конфігурацію комп'ютера. Разом із тим функціональність зосереджена на технічних параметрах без аналізу і обробки.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

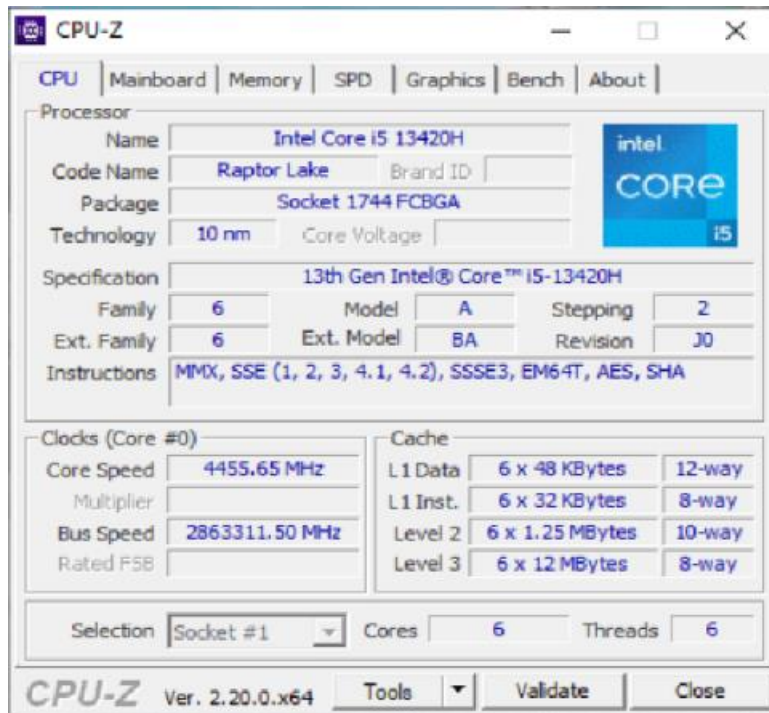


Рисунок 1.5 – CPU-Z

Інструмент MSI Afterburner [3] призначений для контролю та розгону графічних адаптерів, забезпечуючи відображення температур і частот у режимі реального часу. Його функціональність концентрується виключно на відеокартах, що не дозволяє використати його для цілісного аналізу роботи комп'ютера.

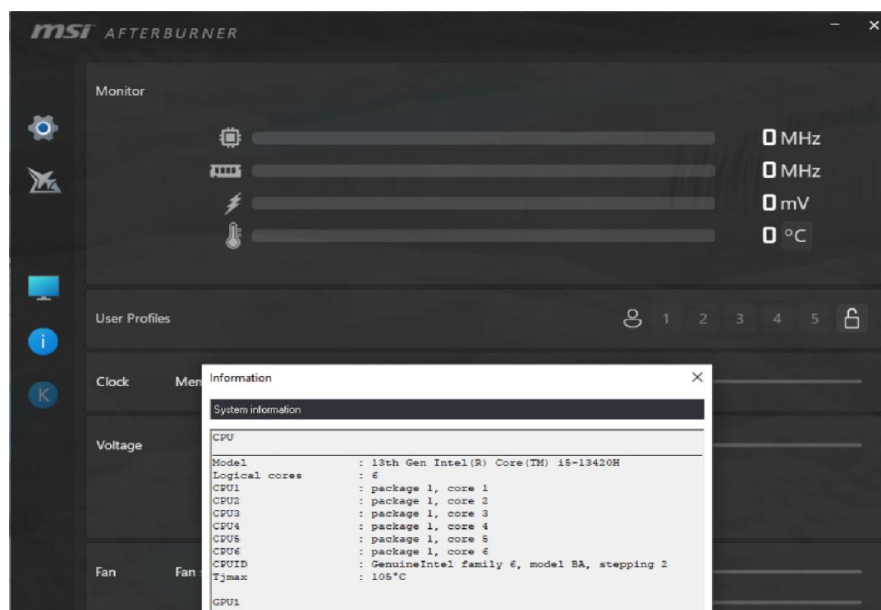


Рисунок 1.6 – MSI Afterburner

Програма Spessy [29] надає базову інформацію про конфігурацію з елементами моніторингу температур, формуючи зручні звіти. Разом із тим відсутність аналізу та прогнозу обмежує її використання у оптимізаційних цілях.

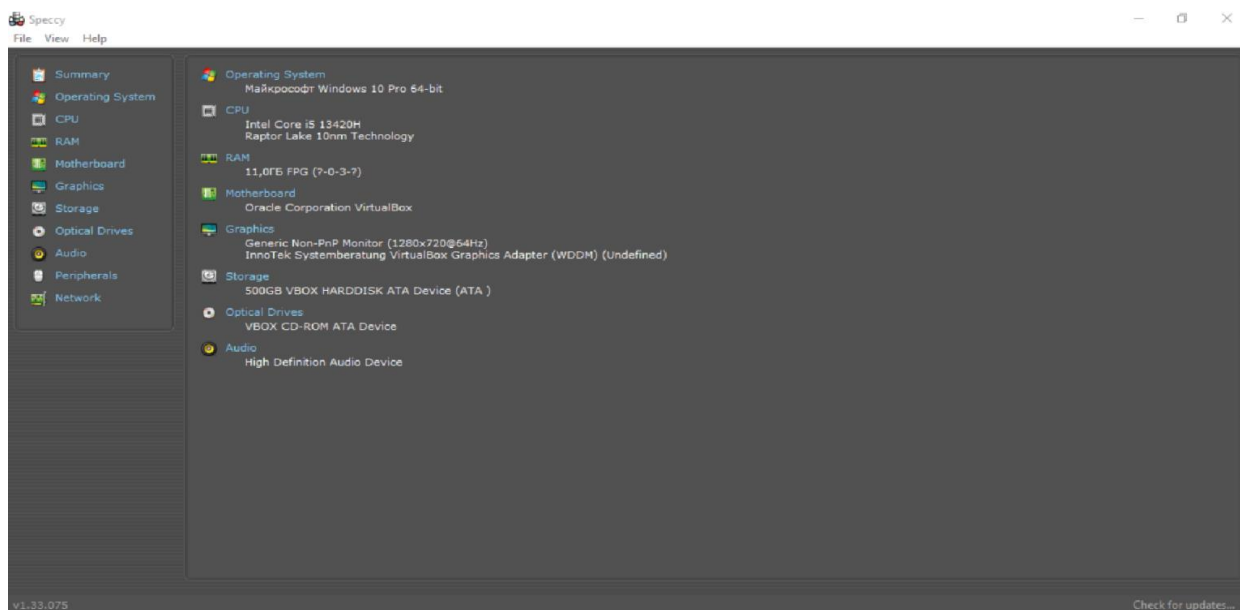


Рисунок 1.7 – Spessy

Окрему групу становлять фірмові утиліти виробників обладнання, зокрема NitroSense [1] від компанії Acer, що реалізує моніторинг температур та керування системою охолодження ноутбуків відповідної серії. Подібні рішення ефективні в межах конкретної платформи, проте не забезпечують універсальності й комплексного аналізу конфігурації.

Фірмові програмні засоби зазвичай орієнтовані на роботу із пристроями лише власного виробництва та використовують спеціалізовані механізми взаємодії з апаратним забезпеченням. Водночас такі рішення мають обмежену функціональність поза межами підтримуваних моделей пристроїв і не можуть використовуватися як інструменти діагностики для різних конфігурацій персональних комп'ютерів. Крім того більшість фірмових утиліт орієнтовані на сканування поточного стану системи, а не проведення комплексного аналізу продуктивності чи формування рекомендацій щодо аналізу продуктивності[19].

системи. Після завершення сканування формується звіт із характеристиками, показниками та рекомендаційними висновками, придатними для подальшого використання у практичній діяльності. Інтеграція аналізу програмної та апаратної складових забезпечує підтримку прийняття рішень щодо оптимізації, модернізації та раціонального використання ресурсів [21].

Отже, реалізація HELPERPC відповідає актуальним потребам у створенні універсального програмно-технічного засобу комплексної діагностики та оптимізації комп'ютерних систем, що поєднує аналіз, автоматизоване формування рекомендацій та адаптивність до різних конфігурацій.

1.3 Головні проблеми апаратної частини ПК

Апаратна складова сучасного комп'ютера формується сукупністю компонентів, від характеристик і збалансованості яких залежить швидкодія, стабільність та енергоефективність системи. До ключових елементів належать центральний процесор, оперативна пам'ять, графічний адаптер, підсистема зберігання даних, материнська плата, блок живлення та система охолодження. Узгоджена робота зазначених складових визначає реальний рівень продуктивності та можливості подальшої модернізації [22].

Центральний процесор виконує роль основного обчислювального вузла, забезпечуючи виконання інструкцій, обробку потоків даних і координацію між апаратними модулями. Актуальними факторами, що впливають на ефективність системи є: підвищене тепловиділення, обмеження продуктивності через недостатню оптимізацію програмного забезпечення, а також значне енергоспоживання. Усунення подібних обмежень пов'язується із застосуванням продуктивних систем охолодження та використанням програм здатних раціонально розподіляти навантаження між ядрами [4-6, 27,31].

Оперативна пам'ять відповідає за тимчасове зберігання даних і швидкий доступ до них під час виконання прикладних задач. Недостатній обсяг, різна

частота модулів або несумісність можуть призводити до зниження пропускну здатності та виникнення затримок у роботі програм. З плином часу можливе часткове зниження стабільності окремих модулів. Підвищення ефективності досягається шляхом використання сучасних стандартів, дотримання принципу симетричної конфігурації каналів та оптимізації механізмів керування пам'яттю на рівні операційної системи.

Графічний адаптер забезпечує обробку візуальних даних, апаратне прискорення відео та виконання обчислень, потрібних для виведення зображення. Значне навантаження під час ресурсомістких задач спричиняє підвищення температури та споживання електроенергії, що потребує належної організації охолодження. Додатковими чинниками виступають сумісні драйвера і відповідність можливостей адаптера сучасним програмним вимогам. Підвищення ефективності досягається оновленням програмного забезпечення, налаштуванням параметрів графіки та застосуванням технологій апаратного прискорення [3,7].

Зберігання даних істотно впливає на швидкість завантаження операційної системи, запуск програм і загальну швидкість відгуку комп'ютера. Традиційні жорсткі диски характеризуються обмеженою швидкістю, тоді як твердотільні накопичувачі демонструють значно вищі показники, проте мають ресурсні обмеження флеш-пам'яті. Сумісність із сучасними інтерфейсами також відіграє важливу роль. Раціональним підходом вважається використання швидкісних SSD або NVMe як системних носіїв із регулярним контролем стану та оновлення мікропрограмного забезпечення.

Материнська плата виконує функцію платформи, що визначає підтримку процесорів, типів пам'яті та інтерфейсів розширення. Обмеження за кількість слотів, відсутність підтримки новітніх стандартів або застарілий контролер можуть стримувати модернізацію. Перспективність конфігурації забезпечується вибором плат із підтримкою актуальних технологій та можливістю поетапного оновлення складових.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

Система живлення й охолодження забезпечує стабільність функціонування всіх компонентів. Недостатня блоку живлення або неефективна вентиляція спричиняють перегрів і нестабільність роботи під час навантаження. Застосування сертифікованих джерел живлення, оптимальна організація повітряного потоку і постійний контроль температур сприяють надійності системи.

У межах веборієнтованого програмно-технічного засобу аналіз апаратних частин став основою для формування логіки сканування та відображення параметрів. Реалізація функціоналу у форматі сайту має деякі обмеження браузерного середовища, яке не забезпечує прямого доступу до низькорівневих апаратних сенсорів. Саме це стало одним із ключових чинників, що ускладнюють створення повноцінного механізму збору даних без використання додаткових програмних модулів або серверної взаємодії [11].

Таким чином, дослідження апаратної структури персонального комп'ютера дозволило визначити параметри, які доцільно враховувати під час формування аналітичного модуля сайту. Врахування характеристик процесора, пам'яті, графічного драйвера, накопичувачів та енергозабезпечення створює підґрунтя для коректної оцінки продуктивності й підготовки рекомендацій щодо оптимізації або модернізації. Поєднання технічного аналізу з вебреалізацією формує основу для створення інструменту, здатного забезпечити узагальнену та структуровану оцінку стану комп'ютера [23,25].

1.4 План реалізації

Програмно-технічний засіб HELPERPC проєктується як система для комплексного аналізу характеристик персонального комп'ютера з подальшим формуванням обґрунтованих рекомендацій щодо підвищення продуктивності та стабільності функціонування. Концепція передбачає поєднання веборієнтованого підходу із можливостями настільного застосунку, що забезпечує гнучкість та автономність. Реалізація планується на застосування HTML, CSS і JavaScript [42,69,70,73-77] із використанням механізмів локального збереження даних у

					КвРКІ 022082.22.02.67 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

браузері, що дозволить мінімізувати залежність від серверної частини, пришвидшить доступ до результатів аналізу та спростить демонстрацію функцій системи. Для розширення доступу до системних ресурсів передбачається використання середовища Electron, яке забезпечує запуск на настільних платформах і поєднує переваги вебтехнологій із можливостями локального виконання коду[17].

Архітектурна модель базується на розподілі процесів із метою підвищення безпеки та стабільності. Основний процес відповідає за керування життєвим циклом застосунку, окремий модуль забезпечує підготовку та передачу системних даних, а інтерфейсний рівень реалізує відображення інформації. Такий підхід дозволить ізолювати доступ до апаратних від візуального середовища, що знижує ризик некоректної обробки та витоку даних [24].

Користувачський інтерфейс проєктується з орієнтацією на наочність і структурованість подання інформації. Після ініціювання сканування відбувається автоматизований збір ключових характеристик, серед яких версія операційної системи, модель процесора, кількість ядер і тактова частота, параметри графічного адаптера, конфігурація оперативної пам'яті, типі і місткість накопичувачів, а також загальний тип пристрою. Отримані відомості відображаються у вигляді інформаційних карток і числових індикаторів, що формують узагальнений профіль системи. Передбачається періодичне оновлення температурних показників із накопиченням історії вимірювань для побудови динамічних графіків. За відсутності доступу до апаратних сенсорів реалізується механізм використання збережених або розрахункових даних, що забезпечує безперервність відображення. Для Windows передбачено використання PowerShell як альтернативного способу отримання температурних даних, що підвищує надійність моніторингу [26].

Аналітична підсистема функціонує у двох режимах. За наявності мережевого з'єднання структуровані параметри системи можуть передаватися до спеціалізованого API для розширеної обробки та формування рекомендацій із

використанням алгоритмів інтелектуального аналізу. У разі відсутності доступу до мережі активується локальний модуль оцінювання, який здійснює аналіз конфігурації, температурних режимів і потенціалу модернізації без залучення серверних ресурсів. Передбачено можливість коригування зібраних даних вручну з подальшим автоматичним оновленням висновків і збереження змін у локальному сховищі [18].

Розроблення HELPERPC пов'язане з необхідністю подолання низки технічних обмежень, зокрема обмеженого доступу вебсередовища до апаратних сенсорів, різноманітних конфігурацій персональних комп'ютерів та неоднорідного драйверного забезпечення. Саме це зумовило впровадити гібридну архітектуру із використанням настільного середовища використання та резервних механізмів отримання даних. Водночас орієнтація на автономність і відсутність обов'язкової серверної складової дозволила зменшити складність розгортання та забезпечити стабільну роботу у різних умовах використання.

Запропонований підхід поєднує локальний збір параметрів із аналітичними алгоритмами, що створює умови для визначення слабких місць конфігурації, оцінювання потенціалу модернізації та формування практичних порад щодо модернізації і оптимізації. Інтеграція вебтехнологій, механізмів локального збереження та інтелектуальної обробки даних демонструє можливість створення універсального цифрового інструменту, адаптованого до різноманітних апаратних середовищ і сценаріїв застосування, з орієнтацією на підвищення ефективності використання комп'ютерної техніки.

1.5 Висновки до першого розділу

У межах першого розділу кваліфікаційної роботи здійснено дослідження предметної області, що охоплює системи моніторингу, тестування та оцінювання комп'ютерної техніки. Проведено аналіз сучасних програмно технічних засобів і веборієнтованих сервісів, призначених для збору відомостей про апаратні складові персонального комп'ютера, відображення технічних характеристик і

					КвРКІ 022082.22.02.67 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

формування рекомендацій щодо підвищення ефективності функціонування системи. Розглянуто функціональні можливості наявних рішень, принципи їх побудови, особливості інтерфейсної організації, підходи до візуалізації показників та способи реалізації аналітичної складової [38-43].

У процесі дослідження виявлено обмеження існуючих продуктів, що проявляються у складності інтерфейсу окремих програм, орієнтованих переважно на фахівців, відсутності пояснення отриманих параметрів, залежності від постійного мережевого з'єднання або необхідності інсталяції додаткових компонентів. Виявлено також недостатню реалізацію механізмів збереження історії перевірок і відсутність динамічної візуалізації конфігурації.

Також виявлено, що багато утиліт мають платформну або апаратну залежність, обмежуючись підтримкою обладнання окремих виробників. Частина програм не надає механізми інтелектуального аналізу отриманих даних, а результати моніторингу потребують додаткової інтерпретації користувачем. Це ускладнює використання таких засобів для оцінки стану ПК та прийняття рішень для оптимізації.

На основі отриманих результатів сформульовано головні параметри програмно-технічного засобу для комплексного аналізу характеристик ПК. Також обґрунтовано доцільність розроблення інструмента, що поєднуватиме автоматизований збір параметрів, їх опрацювання та генерацію рекомендацій щодо оптимізації, модернізації щодо апаратної конфігурації. Визначено загальні вимоги до програмної реалізації, передбачено побудову інтерфейсу з відображенням даних у вигляді інформаційних блоків, таблиць і графіків, а також реалізацію механізму аналізу.

У межах виконання поглиблено знання у сфері веброзробки, опрацьовано принципи створення структурованої розмітки, стилізації інтерфейсу та програмної реалізації логіки функціонування засобами сучасних технологій. Досліджено підходи до модульної організації коду, використання локального

сховища даних і відображення результатів опрацювання у браузерному середовищі.

Додатково у ході аналізу предметної області було визначено ключові принципи організації подібних програмних систем, зокрема багаторівневу архітектуру обробки даних, яка включає етапи збору системної інформації, її нормалізації та подальшої інтерпретації. Окрему увагу приділено способам взаємодії програмного забезпечення з операційною системою через стандартні інтерфейси, що забезпечують доступ до апаратних ресурсів без необхідності прямого апаратного втручання [44].

Також встановлено доцільність використання комбінованого підходу до аналізу даних, який поєднує правила на основі формалізованих обчислень та елементи інтелектуальної обробки інформації. Такий підхід дозволяє підвищити точність оцінювання стану комп'ютерної системи та забезпечити більш адаптивне формування рекомендацій щодо її оптимізації та модернізації.

Таким чином, у першому розділі здійснено комплексний аналіз предметної області, досліджено програмні аналоги, визначено їхні особливості та обмеження, а також обґрунтовано концепцію й основні напрями розроблення програмно-технічного засобу для оцінювання та оптимізації продуктивності комп'ютерної техніки, що створює підґрунтя для подальшої реалізації проєкту. Це дозволило визначити перспективні напрями реалізації програмно-технічного засобу HELPERPC, орієнтованого на поєднання функцій моніторингу, оцінювання та інформаційної підтримки користувача в межах єдиного програмного середовища [46-50].

У подальшому було узагальнено отримані результати аналізу та сформовано концептуальне бачення архітектури майбутнього програмно-технічного засобу, що передбачає інтеграцію модулів збору системної інформації, аналітичної обробки та представлення результатів у зручному для користувача вигляді.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ “HELPERPC”

2.1 Архітектура програмно-технічного засобу та модель взаємодії модулів

Архітектура програмної частини побудована у форматі застосунку на Electron [37], що забезпечує кросплатформеність, ізольовану взаємодію між процесами та стабільний доступ до системних ресурсів. Функціональна структура організована як збірна модель, у межах якої кожен елемент виконує окрему роль у загальному циклі аналізу. Один із сегментів відповідає за зчитування апаратних параметрів персонального комп'ютера та фіксацію технічних характеристик. Інший здійснює математичну обробку даних, їх приведення до уніфікованого масштабу, розрахунок індикаторів продуктивності та визначення похідних коефіцієнтів. Третій забезпечує формування експертних рекомендацій шляхом комбінування локальної моделі з мовною моделлю великого масштабу, доступ до якої організовано через Groq API.

Отримання системної інформації та апаратних характеристик реалізовано у модулі scanner.js. Він відповідає за отримання джерел апаратної інформації операційної системи та збір параметрів центрального процесора, оперативної пам'яті, графічного адаптера та підсистеми зберігання даних. На цьому етапі первинні дані приводяться до зручного для подальшої обробки вигляду. Основний процес Electron відповідає за запуск застосунку, створення вікна, керування життєвим циклом програми та організацію взаємодії між модулями програмно-технічного застосунку. У межах цього процесу підключаються модулі збору даних даних і передаються результати сканування до головного інтерфейсу. Рендер-процес відповідає за відображення користувацького інтерфейсу, виведення отриманих характеристик, результатів діагностики та рекомендації. Саме у цій частині користувач взаємодіє із застосунком, запускає перевірку системи та отримує повний звіт. Передача даних між основним і рендер-процесом виконується через preload.js та IPC-канали. Такий підхід дозволяє не надавати

					КвРКІ 022082.22.02.67 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

інтерфейсній частині прямого доступу до Node.js API, а передавати лише дозволені функції та структуровані дані. Це підвищує безпеку Electron-застосунку і зменшує ризик несанкціонованого доступу до системної інформації. Модуль ai-advisor.js реалізує аналітичну частину системи. Він готує профіль комп'ютера, формує контекст для рекомендацій, виконує локальний rule-based аналіз системи та, за наявності доступу, звертається до Groq API для формування ШІ-рекомендацій. Якщо зовнішній сервіс недоступний, застосунок може виконувати локальну логіку. localStorage використовується для збереження частини даних у браузерному середовищі, наприклад результатів останнього сканування або стану інтерфейсу. Це дозволяє повторно відображати дані швидко і без повторного збору всієї системної інформації. Groq API виступає зовнішнім хмарним сервісом для генерації текстових рекомендацій. Перед відправленням дані мають бути мінімізовані: до запиту передаються лише технічні характеристики необхідні для аналізу, без персональних даних, серійних номерів, шляхів до файлів або інших приватних відомостей [53-57].

У процесі роботи застосунку апаратні дані проходять послідовний шлях від операційної системи до користувацького інтерфейсу та аналітичного модуля. Спочатку значення зчитується з доступних системних джерел, після чого перетворюються у структурований JSON-профіль. Такий профіль містить уніфіковані поля для основних компонентів ПК. Також для цих даних потрібна нормалізація, щоб різні джерела інформації подавалися в однаковому форматі та могли бути використаними як інтерфейсом, так і модулем рекомендацій без додаткової ручної обробки.

Такий підхід дозволяє розділити відповідальність між модулями, підвищити структурованість програмного рішення та забезпечити його гнучкість і масштабованість. Завдяки цьому досягається узгоджена взаємодія між усіма компонентами системи, що є ключовою умовою стабільної роботи застосунку та коректного формування аналітичних висновків і рекомендацій.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

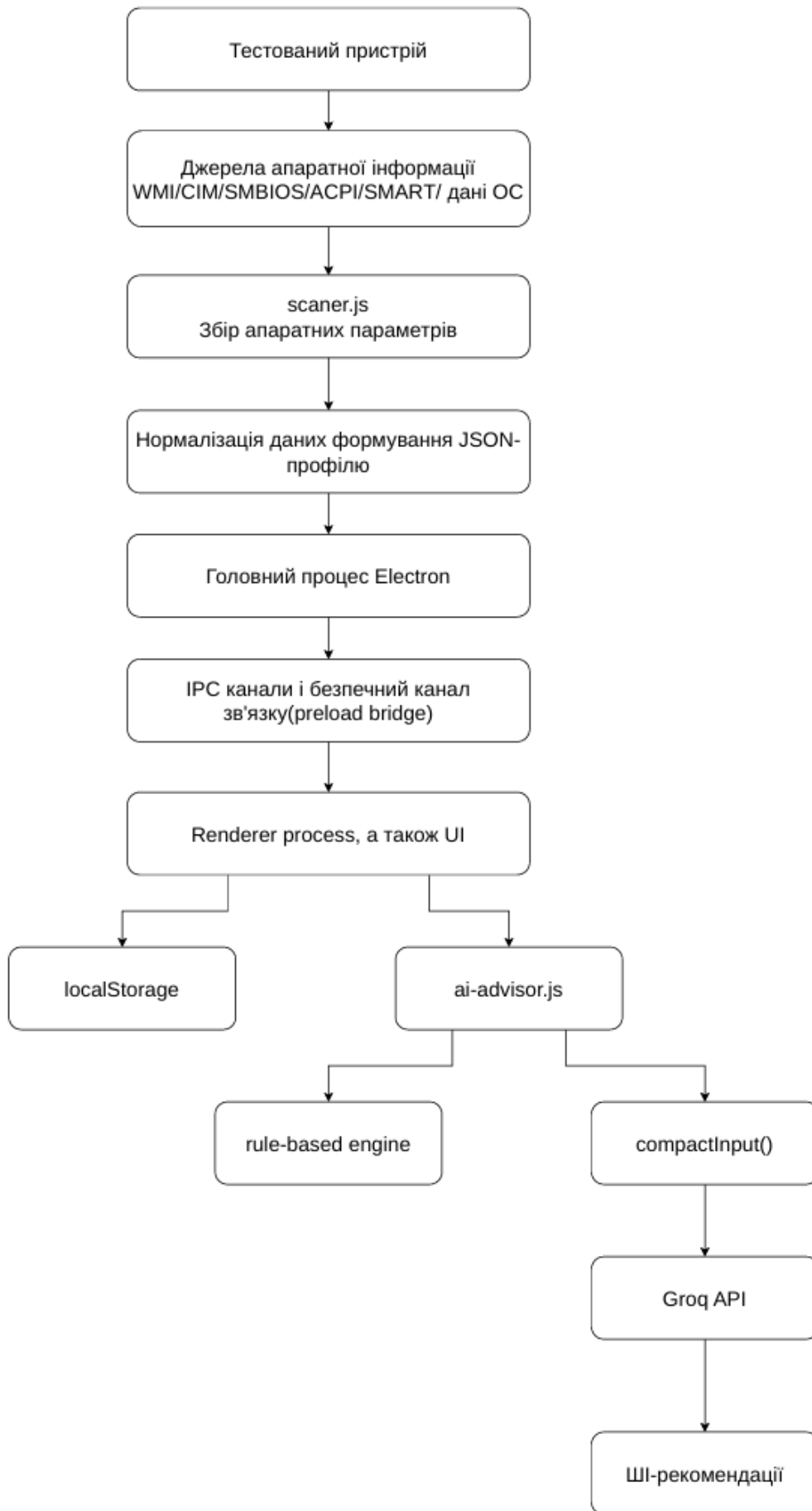


Рисунок 2.1 – Алгоритм сканування системи

Зм.	Арк.	№ докум.	Підпис	Дата

Таким чином архітектура побудована за модульним принципом де scanner.js збирає апаратні дані, основний Electron-процес координує роботу застосунку, preload.js забезпечує контрольований обмін через IPC, рендер-процес відображає результати користувачу, а ai-advisor.js виконує локальний та хмарний аналіз характеристик отриманого системного профілю. Загалом, програмно-технічний засіб функціонує як система, у якій поєднуються формули оцінювання продуктивності, алгоритми нормалізації параметрів та інтелектуальний модуль прийняття рішень, що забезпечує комплексний підхід до аналізу апаратної конфігурації.

2.2 Формули розрахунку параметрів ПК

У межах програми здійснюється приведення обсягів даних до гігабайтового представлення двійкової системи вимірювання, що відповідає архітектурній природі обчислювальної техніки. Перерахунок базується на співвідношенні:

$$GB = \frac{B}{1024^3} \quad (2.1)$$

де B – обсяг використаної пам'яті в байтах

Для відображення результатів у користавацькому інтерфейсі застосовується операція округлення до найближчого цілого значення, що дозволяє уникнути надлишкової деталізації та спростити сприйняття технічних характеристик:

$$GB_{int} = \text{round} \left(\frac{B}{1024^3} \right) \quad (2.2)$$

де B – обсяг використаної пам'яті в байтах

Описаний механізм використовується під час обробки параметрів оперативної пам'яті, де аналізується поле загальної пам'яті, підсистеми зберігання

					КВРКІ 022082.22.02.67 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

даних, зокрема значення розміру диска, а також відеопам'яті графічного адаптера. Для останньої початкові дані подаються у мегабайтах, після чого здійснюється перехід до гігабайтів шляхом ділення загальної пам'яті на 1024, що узгоджується з двійковою моделлю перерахунку одиниць вимірювання.

У процесі реалізації резервного механізму визначення температурних параметрів у середовищі операційної системи Windows використовується інтерфейс WMI, який надає значення температури у специфічному форматі. Отриманий результат отримується у деци-Кельвінах, тобто у десятих частках кельвіна, що потребує додаткового перетворення для приведення до загальноприйнятої шкали градусів Цельсія:

$$T_{\circ C} = \frac{T_{raw} - 27315}{10} \quad (2.3)$$

де T_{raw} задана в деци-Кельвінах;

Така операція враховує перехід від абсолютної температури шкали Кельвіна до шкали Цельсія та компенсацію десяткового масштабу вимірювання. Число 2732 відповідає температурі абсолютного нуля, вираженій у деци-Кельвінах, що забезпечує коректне зміщення від початку відліку.

Застосування цієї залежності дозволяє отримати температурний показник процесора або іншої термозони навіть за відсутності доступу до альтернативних джерел моніторингу, гарантуючи узгодженість результатів вимірювання температури в обчислювальних системах.

Для перетворення температури, отриманої із процесора, контролера або іншого джерела у шкалі Кельвіна, у градуси Цельсія застосовується фізична залежність. Оскільки шкала Кельвіна та Цельсія відрізняються лише нульовою точкою відліку, перетворення виконується шляхом віднімання 273.15 від виміряного значення у кельвінах:

$$T_{\circ C} = T_K - 273.15 \quad (2.4)$$

Цей підхід дозволяє отримати зрозумілий і сумісний із стандартами показник температури. Він застосовується для процесора, відеокарти та інших сенсорів, де надається значення у Кельвінах, забезпечуючи коректну обробку даних у системі моніторингу.

Для забезпечення коректності даних температури у застосунку використовується механізм фільтрації та валідації. Будь-яке значення температури T , яке подається у допустимий інтервал від 10°C до 125°C , вважається валідним і може використовуватися у подальших розрахунках та алгоритмах моніторингу.

$$10 \leq T \leq 125(^{\circ}\text{C}) \quad (2.5)$$

Значення що виходять за ці межі відкидаються, оскільки вони є шумом сенсора або помилковим показником, що може спотворити аналіз системи та призвести до некоректних рекомендацій.

Для аналізу термальних даних у клієнтській частині застосунку використовується статистичний підхід до серії вимірянних температур X_1, X_2, \dots, X_n . Головні показники вимірюються за допомогою таких формул:

Середнє значення температури T розраховується як арифметична середина усіх температур:

$$\bar{T} = \frac{1}{n} \sum_{i=1}^n t_i \quad (2.6)$$

Максимальне значення температури X_{\max} визначається як найбільше значення всіх температур:

					КвРКІ 022082.22.02.67 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

$$T_{\max} = \max(T_1, T_2, \dots, T_n) \quad (2.7)$$

При подачі даних у ШІ використовується лише остання частина серії, що забезпечує актуальність та релевантність рекомендацій.

Для візуалізації температур на графіку застосовується нормування значень у визначеному діапазоні:

$$[T_{\min}, T_{\max}], \text{ де } T_{\min} = 20, T_{\max} = 105 \quad (2.8)$$

Це дозволяє зобразити температуру у вигляді координати на вертикальній осі незалежно від абсолютних значень.

Спершу обчислюється нормоване значення температури t з використанням операції обмеження, щоб уникнути виходу за межі діапазону:

$$t = \frac{C \cdot (T, T_{\min}, T_{\max}) - T_{\min}}{T_{\max} - T_{\min}} \quad (2.9)$$

де C – обмеження

Далі координата обчислюється з урахуванням висоти області відображення H та відступу p :

$$y = H - p - t(H - 2p) \quad (2.10)$$

Ця нормалізація забезпечує коректне відображення температури в межах графіка, де H – висота, p – відступ зверху і знизу для вигляду та уникнення накладень.

2.3 Інженерні формули для розрахунку параметрів ПК

Для оцінки теоретичної пропускної здатності оперативної пам'яті застосовується інженерна формула для DDR-модулів. Пропускна здатність обчислюється як добуток ефективної частоти пам'яті, ширини каналу та кількості каналів:

$$BW_{RAM} = f_{MT/s} \cdot 10^6 \cdot W_{bytes} \cdot N_{ch} \quad (2.11)$$

де $f_{MT/s}$ – ефективна швидкість передачі даних у мегатрансферах на секунду, W_{bytes} – ширина одного каналу в байтах, а N_{ch} – кількість каналів пам'яті.

Також для зручності порівняння та відображення у системі значення конвертуються у гігабайти на секунду:

$$BW_{RAM, GiB/s} = \frac{f_{MT/s} \cdot W_{bytes} \cdot N_{ch}}{1000} \quad (2.12)$$

Ця оцінка безпосередньо залежить від конфігурації слотів пам'яті, а саме впри встановлені двох ідентичних модулів зазвичай активується dual-channel, що подвоює ефективну пропускну здатність. Також швидкість dual-channel залежить від підтримки цієї технології процесором, чипсетом та материнською платою. Для коректної роботи двоканального режиму модулі пам'яті повинні бути встановлені у відповідні слоти, визначені виробником материнської плати, а також підтримуватися поточними налаштуваннями BIOS/UEFI.

Крім апаратної сумісності, ефективність роботи dual-channel може залежати від характеристик самих модулів пам'яті, зокрема обсягу, частоти та таймінгів. У разі неправильного встановлення модулів або відсутності підтримки

апаратної платформи система може автоматично перейти у одноканальний режим роботи пам'яті.

Для оцінки теплового запасу процесора та відеокарти використовується інженерна формула, яка визначає різницю між граничною допустимою температурою та поточним значенням:

$$H_{\text{thermal}} = T_{\text{limit}} - T_{\text{current}} \quad (2.13)$$

де T_{limit} – гранична температура для конкретного процесора або відеокарти, а T_{current} – поточне виміряне значення температури.

Якщо H_{thermal} малий або від'ємний, це сигналізує, що система знаходиться в зоні ризику тротлінгу, тобто активується захист від перегріву, який може обмежувати продуктивність.

Для десктопних систем практичними орієнтирами є:

$T \geq 90 - 95^{\circ}\text{C}$ – критична температури CPU;

$T \geq 85 - 90^{\circ}\text{C}$ – критична температура GPU.

Досягнення таких порогів є сигналом для користувача або системи, про необхідність покращення охолодження, а також покращення оптимізації навантаження.

Для оцінки узагальненого ризику перегріву системи використовується індекс ризику, який комбінує показники CPU та GPU з урахуванням вагових коефіцієнтів:

$$R_T = \max\left(w_1 \frac{T_{\text{CPU}}}{T_{\text{CPU,lim}}}, w_2 \frac{T_{\text{GPU}}}{T_{\text{GPU,lim}}}\right) \quad (2.14)$$

Додатково для комплексної оцінки стану може використовуватися зважений індекс ризику:

$$R_T = w_1 \frac{T_{CPU}}{T_{CPU,lim}} + w_2 \frac{T_{GPU}}{T_{GPU,lim}} \quad (2.15)$$

де R_T – індекс ризику, T_{CPU} та T_{GPU} – поточні температури процесора і відеокарти, $T_{CPU,lim}$ та $T_{GPU,lim}$ – граничні температури для CPU і GPU відповідно, а w_1 та w_2 – вагові коефіцієнти, вони мають відповідати $w_1 + w_2 = 1$.

Інтерпретація значень індексу:

$R_T < 0.8$ – нормальний режим, система працює без ризику перегріву;

$0.8 \leq R_T < 1.0$ – підвищений ризик, рекомендується стежити за температурою та охолодженням;

$R_T \geq 1.0$ – критична зона, існує високий ризик тротлінгу та можливого зниження продуктивності або автоматичного захисту.

Такий індекс дозволяє оцінювати ризики перегріву в комплексі для процесора та відеокарти, комбінуючи їх у єдиний параметр для ШІ-моделі та локальних правил.

Для оцінки ефективності використання дискової підсистеми та планування апгрейту SSD застосовується метрика заповнення диска. Яка визначається як відношення об'єму зайнятого простору до загального об'єму накопичувача:

$$U_{SSD} = \frac{V_u}{V_t} \cdot 100\% \quad (2.16)$$

де, V_u – об'єм зайнятого користувачем простору на диску, а V_t - загальна ємність SSD.

Для забезпечення стабільної продуктивності диска практичний поріг рекомендується тримати на рівні:

$$U_{SSD} \leq 80\% \quad (2.17)$$

Перевищення цього значення може призводити до деградації швидкості запису та читання, зниження ефективності кешування і загального часу відгуку диска.

2.4 Інтелектуальний метод прийняття рішень та формування рекомендацій для конфігурації комп'ютера

У програмно-технічному засобі реалізовано гібридний підхід до формування інтелектуальних рекомендацій, що поєднує використання локального алгоритмічного механізму аналізу та мовної моделі великого масштабу. Архітектура системи передбачає два рівні обробки інформації. Перший базується на використанні мовної моделі доступ до якої відбувається через Groq API. Другий рівень представлений локальним механізмом із базовими правилами, який працює незалежно від мережевого підключення та забезпечує формування рекомендацій на основі заздалегіть визначених інженерних правил. Такий підхід дозволяє поєднати гнучкість генерації тексту з передбачуваністю та стабільністю алгоритмів.

Застосування мовної моделі забезпечує можливість інтерпретації технічних параметрів системи з урахуванням апаратних характеристик, що дозволяє формувати змістовні рекомендації та поради щодо оптимізації роботи комп'ютера. Водночас локальний механізм аналізу гарантує відмовостійкість програмно-технічного засобу. У випадках відсутності мережевого з'єднання, помилки доступу до API або отримання некоректної відповіді зовнішньої моделі формування рекомендацій виконується внутрішнім алгоритмом. Завдяки такій організації навіть за відсутності доступу до зовнішніх сервісів система здатна сформувати набір практичних рекомендацій щодо покращення стану комп'ютера на основі наявних технічних показників.

Локальний механізм прийняття рішень ґрунтується на аналізі параметрів апаратної конфігурації. Вхідні дані системи формують вектор характеристик,

					КвРКІ 022082.22.02.67 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

який містить значення температур центрального процесора та графічного адаптера, обсяг оперативної пам'яті, кількість встановлених модулів, тип накопичувача, категорію пристрою, виробника процесора та інші параметри, отримані під час сканування системи. Даний вектор характеристик:

$$x = [T_{\text{CPU}}, T_{\text{GPU}}, \text{RAM}, \text{SSD}_{\text{TYPE}}, \text{Device}, \text{CPU}_{\text{VENDOR}}, \dots] \quad (2.18)$$

На основі цього набору значень виконується перевірка проблем пов'язаних із стабільністю роботи або потенційною необхідністю модернізації апаратних компонентів. Кожне правило має результат, яке приймає значення істини або хибності. У випадку виконання умови до списку рекомендацій додається відповідне текстове повідомлення, яке описує можливі покращення продуктивності або стабільності системи. Набір правил:

$$r_i = r_i(x) \in \{0,1\} \quad (2.19)$$

Кожне правило активує свій текст:

$$\text{if } r_i(x) = 1 \Rightarrow \text{add recommendation } m_i \quad (2.20)$$

Прикладом таких умов може бути перевищення температури процесора чи відеокарти, що означає роботу на критичному тепловому режимі. Для CPU цей рівень становить близько дев'яноста градусів, для GPU - в діапазоні вісімдесяти восьми градусів і вище розглядаються як потенційно небезпечні для правильної роботи ПК. Окремі правила стосуються конфігурації оперативної пам'яті. Якщо користувач має невеликий обсяг оперативної пам'яті, це може бути підставою для рекомендації щодо її розширення. У разі наявності лише одного модуля пам'яті, можна повідомити про можливість активації двоканального режиму шляхом

встановлення другого аналогічного модуля у вільний слот. Аналіз типу накопичувача дозволяє визначити, чи використовується сучасний високошвидкісний інтерфейс NVMe або більш повільні рішення, що також може слугувати основою для рекомендації по покращенню. Тип пристрою враховується для коректного формування підказок, оскільки для портативних пристроїв доцільніше акцентувати увагу на модернізації пам'яті або накопичувачів, тоді як рекомендації, характерні для стаціонарних системи, можуть бути недоречними.

Після виконання перевірки всіх умов формується набір рекомендацій, отриманий як об'єднання правил базових загальних порад. Далі здійснюється усунення повторів і вибір обмеженої кількості найбільш релевантних результатів. У фінальному вигляді користувач отримає шість рекомендацій, які відображають ключові аспекти стану системи та можливі напрями її оптимізації.

$$I_f = \text{Top6}(U(I_p \cup I_b)) \quad (2.21)$$

де, I_f – фінальні показники, I_p – первинні показники, I_b – базові показники, U – унікальні.

Мовна модель у структурі програмно-технічного засобу використовується для узагальнення результатів аналізу та формування зрозумілого текстового пояснення стану комп'ютера. До моделі передається коротке представлення результатів сканування у форматі структурованих даних, що містить основні апаратні параметри та температурні показники. У відповідь формується структурований результат, який включає коротке узагальнення технічного стану системи та набір рекомендацій у фіксованій кількості пунктів. Такий підхід забезпечує контрольовану генерацію тексту, оскільки робота моделі обмежується попередньо визначеним системним запитом і набором переданих технічних даних, що дозволяє використовувати можливості штучного інтелекту як інструмент аналізу.

2.5 Джерела і архітектура отримання апаратної інформації

					КВРКІ 022082.22.02.67 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

Для отримання апаратних характеристик комп'ютера програмно-технічний засіб використовує інформацію доступну через операційну систему та її системні інтерфейси. У середовищі Windows такі дані можуть надходити із декількох джерел: WMI/CMI, SMBIOS/DMI, SCPI, SMART/NVMe, а також зі стандартних даних ОС про процесор, оперативну пам'ять, відеоадаптер, накопичувачі, а також інші елементи системи.

WMI та CIM є одним із основних механізмів отримання системної інформації. Через них можна отримати дані про модель і кількість потоків процесора, обсяг оперативної пам'яті, версію і розробника операційної системи, відеоадаптер, накопичувачі та інші апаратні параметри. Такі джерела зручні тим що надають структуровану інформацію без прямого звернення до низькорівневого обладнання.

SMBIOS/DMI містить відомості, які передаються прошивкою BIOS/UEFI. До них належать дані про виробника, модель системи, тип корпусу, модулі оперативної пам'яті та інші характеристики. Ця інформація корисна для визначення форм-фактора пристрою, однак її точність залежить від того, наскільки коректно виробник заповнював відповідні поля.

ACPI Thermal Zone використовується для отримання інформації про температурні зони. Через ACPI операційна система може отримувати частину даних про тепловий стан пристрою, режими енергоспоживання та роботу охолодження. Проте такі значення не завжди надають точні температури GPU або CPU, оскільки можуть описувати лише загальні теплові зони пристрою.

Smart та NVMe-інтерфейси застосовуються для отримання діагностичної інформації про накопичувачі. Вони можуть надавати дані та інші службові показники. Для NVMe-накопичувачів частина параметрів може бути доступна через окремі команди або драйвери, а для SATA-дисків через SMART.

Операційна система також надає інформацію про CPU, RAM, і GPU через власні системні API та драйвери. Наприклад, можна отримати назву процесора,

					КвРКІ 022082.22.02.67 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

кількість логічних потоків, загальний обсяг пам'яті, модель відеоадаптера та доступний обсяг відеопам'яті. У програмному засобі ці дані збираються через бібліотеку “systeminformation”, яка уніфікує доступ до різних системних джерел і повертає дані у структурованому вигляді.

Водночас доступ до апаратних сенсорів має певні обмеження. Температура CPU, GPU або SSD може бути недоступною через відсутність відповідного драйвера, обмеження BIOS/UEFI, політику безпеки ОС або особливості конкретної моделі пристрою. На ноутбуках виробники часто приховують частину сенсорів або надають їх через власні утиліти. Тому програмно-технічний засіб має враховувати можливість неповних даних і використовувати резервну логіку, коли певний параметр неможливо отримати автоматично.

Збір апаратної інформації у програмно-технічному засобі виконується послідовно: спочатку операційна система надає доступ до даних про компоненти комп'ютера через системні інтерфейси, після чого ці дані зчитуються за допомогою бібліотеки systeminformation, PowerShell або WMI-запитів. Далі модуль scanner.js формує набір первинних параметрів, які проходять етап нормалізації. На цьому етапі виконується уніфікація назв полів, приведення одиниць вимірювання та підготовка структури для подальшого використання в інтерфейсі й аналітичних модулях.

Додатково під час збору системної інформації важливим етапом є перевірка достовірності та узгодженості отриманих параметрів, оскільки різні джерела можуть повертати значення у відмінних форматах або з різним рівнем деталізації. Саме тому у програмно-технічному засобі реалізовано механізм нормалізації та узгодження даних, що дозволяє сформувати єдиний структурований профіль комп'ютера, придатний для подальшого аналізу, відображення в інтерфейсі та формування рекомендацій щодо оптимізації роботи системи. Блок схема зображена на Рис.2.2:

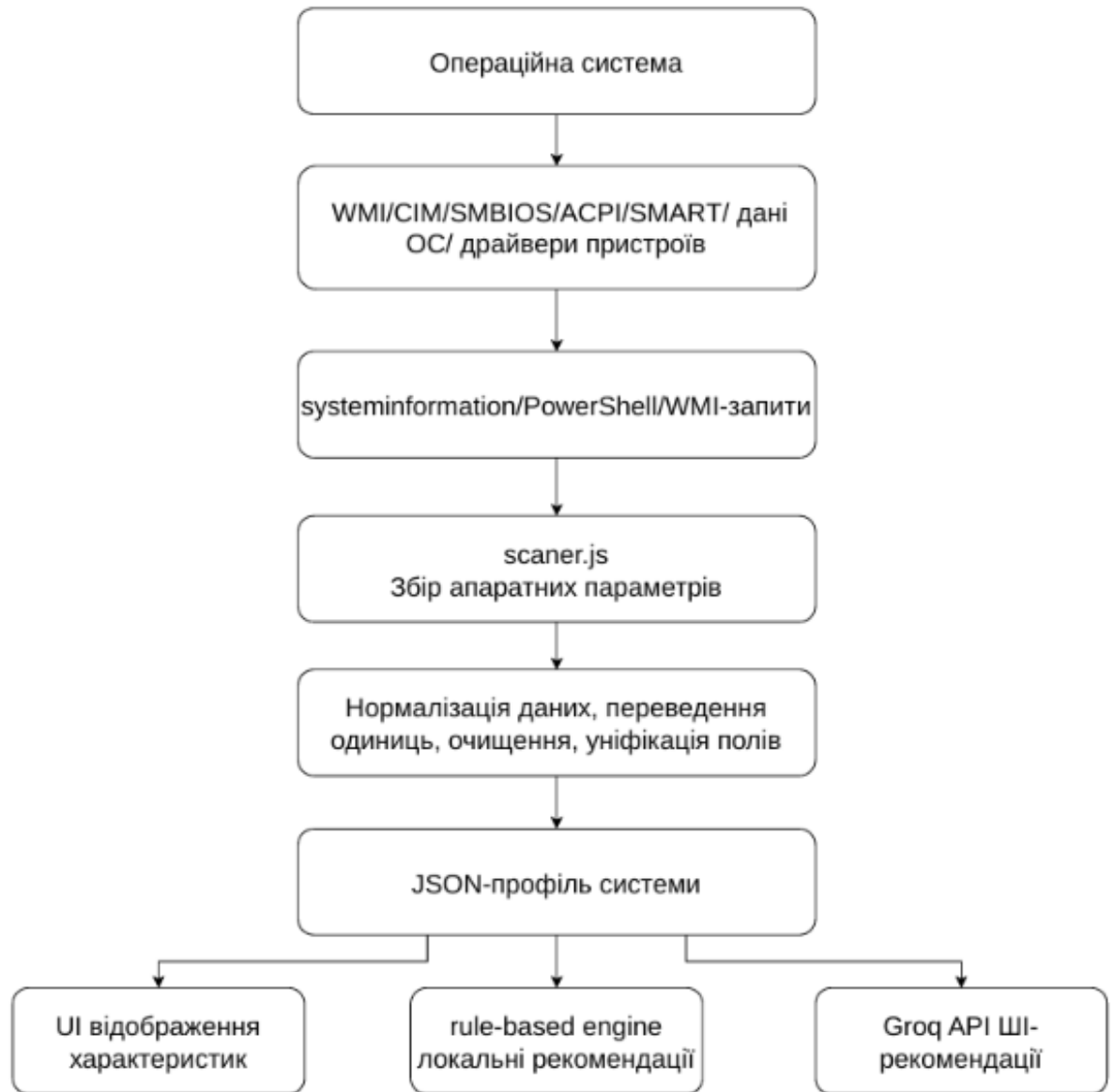


Рисунок 2.2 – Архітектурна схема збору та обробки апаратних даних

3) Модель даних системного профілю

```

{
  "os": "Microsoft Windows 11 Home",
  "device": {
    "type": "Laptop",
    "manufacturer": "Acer",
    "model": "Nitro ANV15-51",
    "architecture": "x64"
  },
  "cpu": {

```

```

    "model": "13th Gen Intel Core i5-13420H",
    "threads": 12,
    "baseClockGHz": 2.1,
    "maxClockGHz": 4.4,
    "temperatureC": 74
  },
  "ram": {
    "totalGiB": 32,
    "modules": 2,
    "speedMTs": 5200,
    "manufacturer": "Samsung",
    "channelsEstimated": 2
  },
  "gpu": [
    {
      "model": "NVIDIA GeForce RTX 3050 6GB Laptop GPU",
      "type": "Discrete GPU",
      "vramGiB": 6
    },
    {
      "model": "Intel UHD Graphics",
      "type": "Integrated GPU"
    }
  ],
  "storage": [
    {
      "model": "NVMe WD PC SN740 SDDQNQD-512G-1014",
      "type": "NVMe SSD",
      "capacityGiB": 477
    }
  ]
}

```

Наведена структура є узагальненою моделлю системного профілю, яка використовується для подальшого аналізу. Поле os описує операційну систему,

					КвРКІ 022082.22.02.67 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

device визначає тип пристрою, блок cpu містить параметри центрального процесора, блок ram описує оперативну пам'ять, блок gpu містить характеристики графічного адаптера, а блок storage описує підсистему зберігання даних. Саме такий профіль може передаватися до користувацького інтерфейсу, локального rule-based engine або скорочуватися перед відправленням до Groq API.

2.6 Блок-схеми алгоритмів роботи

Алгоритм сканування призначений для автоматичного збору детальної інформації про апаратні ресурси комп'ютера та формування повного профілю системи для подальшого аналізу та генерації рекомендацій. Першим етапом виконується запуск процедури сканування, під час якої ініціюється запит scan-system, що дозволяє отримати основні характеристики ПК. Далі здійснюється паралельне зчитування даних про операційну систему, центральний процесор і його температуру, конфігурацію модулів оперативної пам'яті, графічні адаптери та накопичувачі. Отримані показники перевіряються на коректність і в разі наявності достовірного значення воно використовується без змін, якщо ж виникли проблеми застосовується резервний метод через WMI або системні лічильники для Windows. Усі зібрані дані проходять через етап нормалізації, під час якого перевіряється правильність форматів і відповідність параметрів. Наступним кроком виконується обчислення обсягу оперативної та відеопам'яті, сумарного обсягу SSD-накопичувачів та кількості встановлених модулів пам'яті, після чого формується профіль комп'ютера, що містить усі характеристики і передається до інтерфейсу для відображення та подальшого використання. Блок схема зображена на Рис.2.3:

					КвРКІ 022082.22.02.67 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

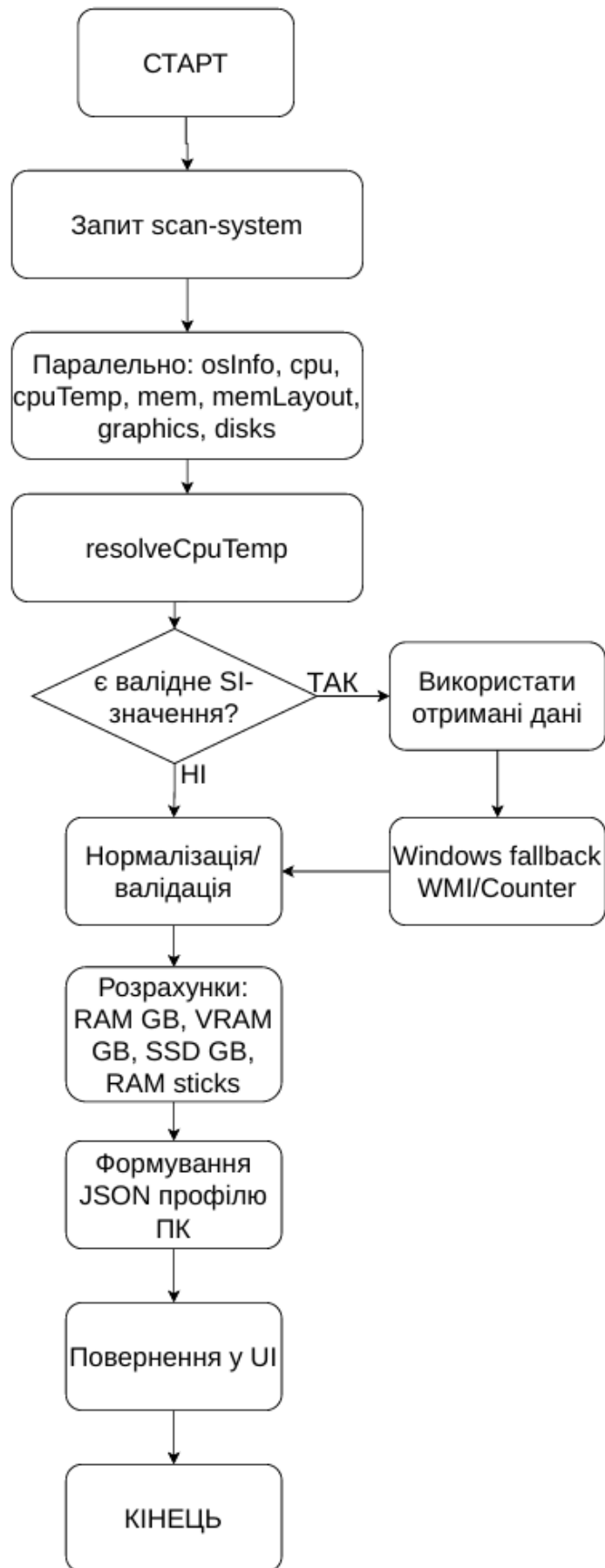


Рисунок 2.3 – Алгоритм сканування системи

Алгоритм термомоніторингу відповідає за постійне відстеження температурних показників процесора та графічного адаптера з метою контролю стабільності та працездатності системи. Після запуску процедури встановлюється інтервал оновлення кожні п'ять секунд, під час якого здійснюється зчитування актуальних температурних даних через модуль scan-thermals. Показники перевіряються на коректність, і якщо вони недоступні або некоректні, використовується резервне джерело інформації, отримане вручну або автоматично. Отримані значення додаються до історії вимірювань з обмеженням максимального обсягу даних, що дозволяє підтримувати актуальність та оптимальне використання пам'яті. На основі історії обчислюється поточне, середнє та максимальне значення температури, після чого оновлюється користувацький інтерфейс, графіки та дані для модуля ШІ, що забезпечує подальший аналіз. Процедура повторюється автоматично при наступному спрацюванні таймера, підтримуючи безперервний моніторинг. Додатково алгоритм передбачає механізм стабілізації процесу збору даних, що дозволяє зменшити вплив випадкових коливань температурних показників та забезпечити більш точний аналіз стану системи. У разі перевищення встановлених порогових значень може формуватися попередження або передаватися сигнал до інших модулів системи для виконання відповідних дій, зокрема зміни режиму роботи, регулювання навантаження чи активації додаткового охолодження. Такий підхід забезпечує підвищення надійності функціонування програмно-апаратного комплексу та своєчасне реагування на потенційно критичні ситуації.

Крім цього, збереження історії температурних змін дозволяє виконувати подальший аналіз динаміки нагрівання компонентів у різних режимах експлуатації. Це дає можливість визначати тенденції зміни температур, оцінювати ефективність системи охолодження та використовувати накопичені дані для прогнозування можливих перевантажень. Завдяки інтеграції з модулем штучного інтелекту забезпечується формування більш точних аналітичних висновків і підвищується ефективність системи. Блок схема зображена на Рис.2.4:

					КвРКІ 022082.22.02.67 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

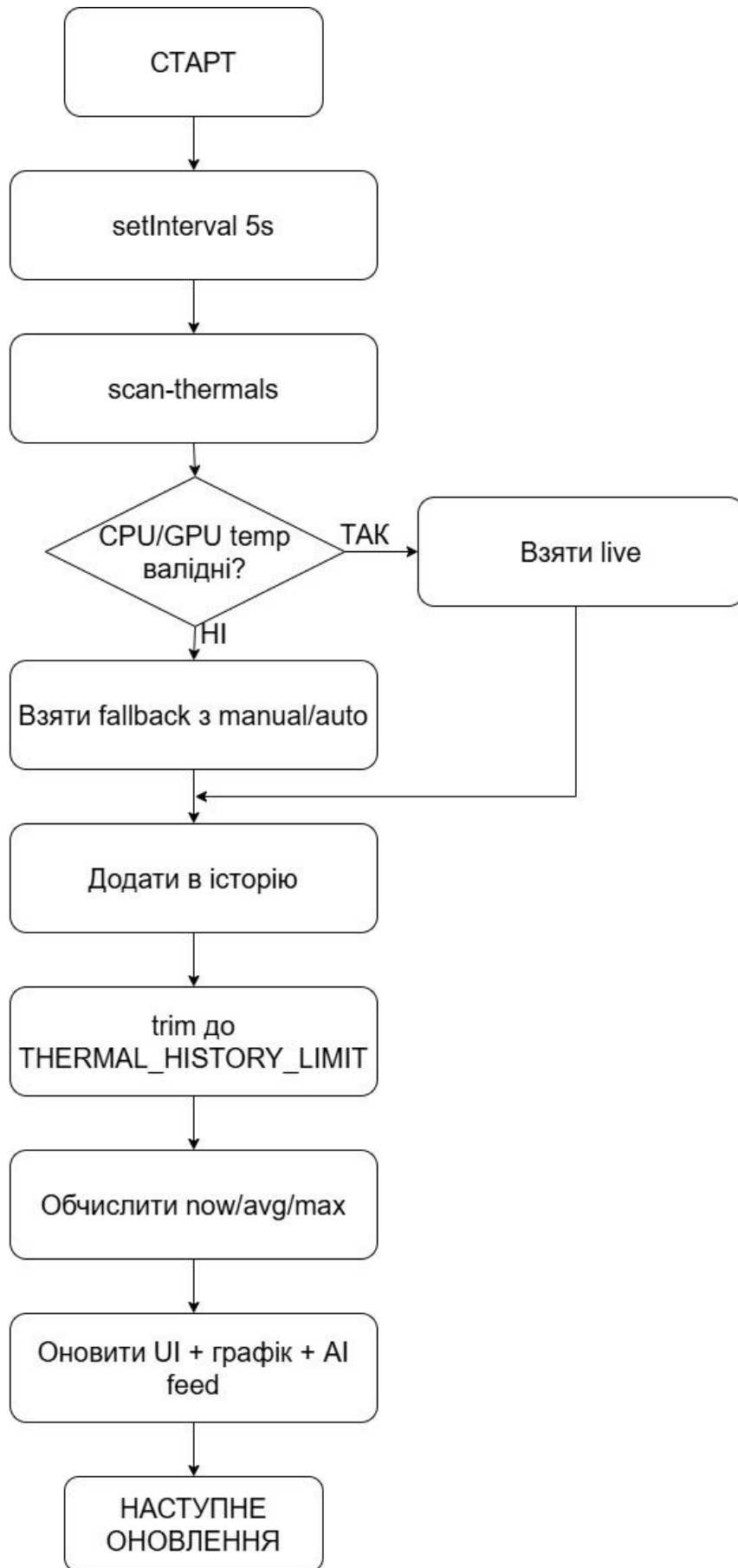


Рисунок 2.4 – Алгоритм термомоніторингу

Алгоритм формування рекомендацій на основі штучного інтелекту забезпечує аналіз конфігурації та теплових показників для створення порад щодо підвищення продуктивності та стабільності роботи обладнання. На початковому етапі збирається потрібна інформація, що містить апаратні, програмні та температурні параметри, яка використовується для створення рекомендацій за допомогою rule-based механізму. Якщо ж наявний доступ до зовнішнього сервісу Groq, дані надсилаються для обробки з дотриманням обмеження часу відповіді. Після отримання відповіді проводиться перевірка запиту, відповідей та їх структури. У разі невідповідності даних повертаються локальні рекомендації із повідомленням про помилку. У разі успішного отримання інформації, рекомендації сформовані штучним інтелектом, об'єднуються з локальними, різні дублікати видаляються, відбираються кращі відповіді до шести штук і передаються на відображення в інтерфейс користувача. Інформація забезпечує своєчасну і точну підтримку в прийнятті рішень щодо оптимізації роботи системи та контролю за її станом. Алгоритм передбачає механізм попередньої обробки та нормалізації отриманих даних, що дозволяє забезпечити коректність подальшого аналізу незалежно від типу обладнання або джерела інформації. Під час формування рекомендацій враховуються поточне навантаження на систему, динаміка зміни температурних показників, рівень використання процесора та графічного адаптера, а також особливості програмного середовища. Це дає можливість створювати більш адаптивні та релевантні поради щодо оптимізації роботи системи, зменшення теплового навантаження та підвищення енергоефективності обладнання.

Також, інтеграція локального rule-based механізму із зовнішнім сервісом штучного інтелекту дозволяє підвищити надійність та гнучкість роботи системи рекомендацій. У випадку відсутності підключення до мережі або перевищення часу очікування відповіді система продовжує функціонувати автономно, використовуючи локально сформовані рекомендації. Блок-схема зображена на Рис.2.5:

					КвРКІ 022082.22.02.67 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

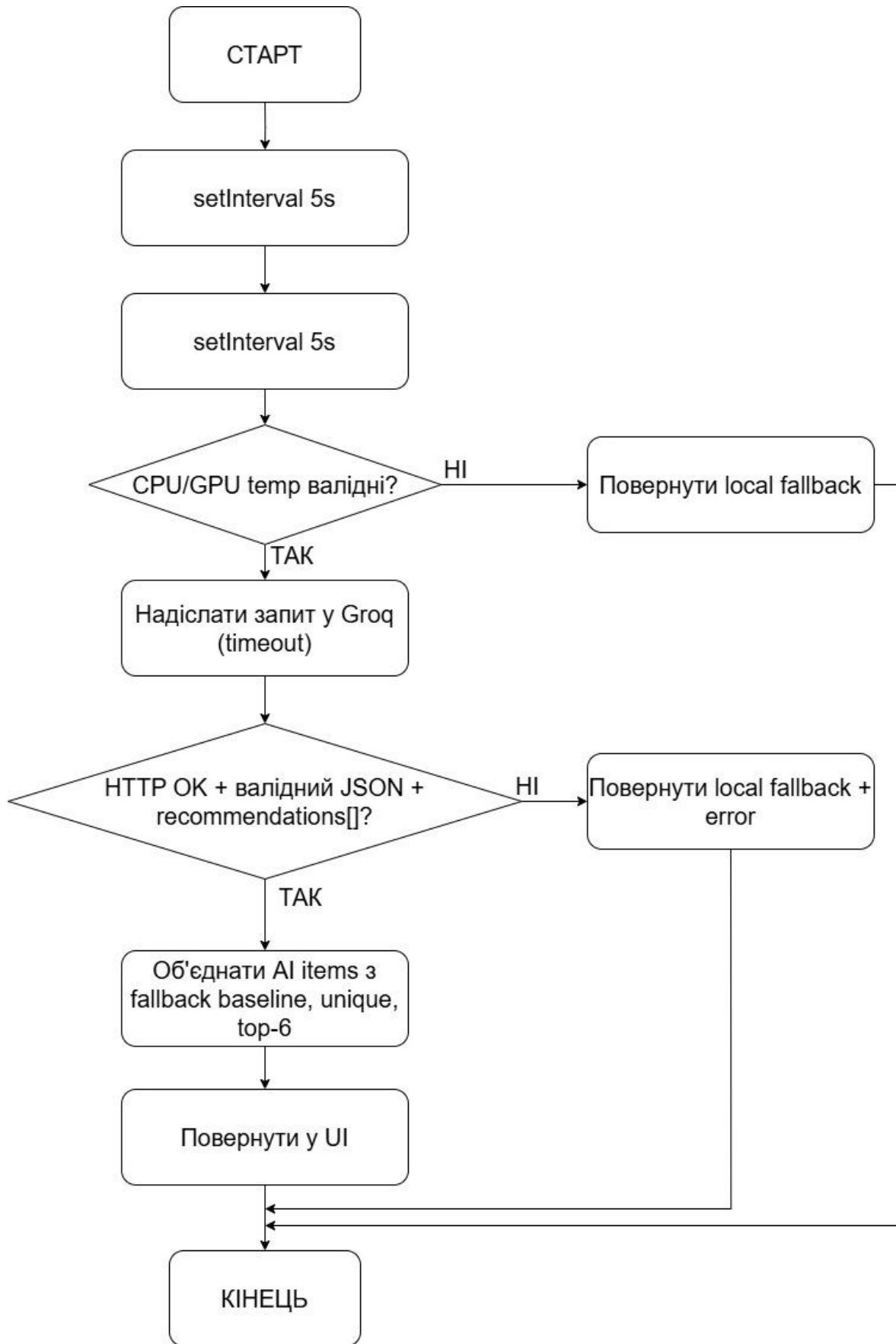


Рисунок 2.5 – Алгоритм ІІІ-рекомендацій

2.7 Висновки до другого розділу

У даному розділі було розглянуто математичне, алгоритмічне та інтелектуальне забезпечення програмно-технічного засобу віддаленого моніторингу комп'ютерної системи. Описані методи обробки та перетворення даних, забезпечуючи представлення апаратних і температурних параметрів у зручному для аналізу вигляді. Зокрема, реалізовано механізми перетворення обсягів пам'яті у гігабайтове представлення, перерахунку температур із різних шкал вимірювання у градуси Цельсія, а також фільтрацію та нормалізацію температурних показників для підвищення достовірності даних.

Для обробки температур застосовано різні методи, що дозволяють визначати середні та максимальні значення температур, а також забезпечують коректну візуалізацію даних за допомогою нормування значень для побудови графіків. Крім того, наведено інженерні формули, що використовуються для оцінки характеристик комп'ютерної системи, зокрема пропускної здатності оперативної пам'яті, теплового запасу компонентів, температурного індексу ризику та рівня заповненості дисків.

Особливу увагу надано реалізації моделі для формувань рекомендацій, яка поєднує локальну модель rule-based механізму та мовну модель великого масштабу. Такий підхід забезпечує гнучкість аналізу та водночас гарантує відмовостійкість у разі відсутності доступу до зовнішніх серверів.

Також було розглянуто алгоритми сканування системи термомоніторингу та формування рекомендацій, які забезпечують збір та обробку інформації про апаратні ресурси, постійний контроль температурних показників і генерацію рекомендацій щодо оптимізації роботи комп'ютера.

Таким чином, запропонований підхід дозволяє створити застосунок для аналізу стану ПК, який поєднує методи алгоритмічної обробки даних, можливості сучасних технологій і методи інженерних розрахунків для підвищення ефективності моніторингу та допомоги у прийнятті рішень.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Загальна характеристика реалізованого засобу

Розроблений програмно-технічний засіб HELPERPC є настільним застосунком класу діагностично-аналітичних систем, призначеним для автоматизованого збору параметрів ПК, безперервного віддаленого моніторингу температур та формування рекомендацій щодо покращення і модернізації системи. Основною метою створення такого засобу є надання користувачеві інструменту, який дозволяє швидко оцінити технічний стан комп'ютера, визначити потенційні проблеми у роботі обладнання та отримати поради щодо їх усунення. Особливістю реалізації є поєднання функції апаратного сканування, аналізу температурних показників і формування рекомендацій у межах програми, що значно підвищує зручність та ефективність використання користувача із системою.

У сучасних умовах експлуатації комп'ютерів користувачі часто стикаються з проблемами перегріву компонентів, нестачі оперативної пам'яті або недостатньої продуктивності системи. При цьому більшість доступних рішень спрямовані лише на одну вузьку задачу: або на аналіз продуктивності, або на відображення характеристик, або на моніторинг температур. Розроблений програмно-технічний засіб HELPERPC інтегрує ці функції в єдину систему, що дозволяє здійснювати комплексний аналіз стану комп'ютера. Таким чином користувач отримає не лише набір показників, але й їх пояснення, а також рекомендації щодо шляхів оптимізації або модернізації системи.

Функціонально програмно-технічний засіб об'єднує три ключові класи задач, які традиційно розглядаються окремо. Першим із яких є сканування апаратної конфігурації комп'ютера, яка включає визначення процесора, оперативної пам'яті, графічного адаптера, накопичувачів та операційної системи. Другим компонентом є контроль температур, що передбачає періодичне

					КвРКІ 022082.22.02.67 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

зчитування показників із датчиків та аналіз їх динаміки у реальному часі. Третім компонентом є система підтримки прийняття рішень, яка на основі отриманих параметрів формує рекомендації щодо модернізації або оптимізації роботи комп'ютера. Саме ці функціональні блоки формують практичну цінність системи, оскільки дозволяють отримати не лише інформацію про стан системи, але й оцінити її з точки зору доцільності у модернізації

З архітектурної точки зору програмно-технічний засіб реалізовано на основі платформи Electron, яка поєднує можливості веб-технологій із функціональністю настільних застосунків. Такий підхід дозволяє використовувати сучасні методи побудови інтерфейсу користувача та доступу до системних ресурсів операційної системи. Архітектура застосунку поділена на кілька рівнів відповідальності, кожен із яких виконує окрему функціональну роль у роботі [51,52].

Перший рівень архітектури представлений рендер-процесом, який відповідає за реалізацію графічного інтерфейсу користувача. На цьому рівні здійснюється відображення апаратних характеристик ПК, виведення рекомендацій, графіків температур і забезпечення взаємодії між системою та користувачем. Інтерфейс реалізований за допомогою HTML, CSS та JS, що забезпечує гнучкість у створенні елементів та сучасного дизайну.

Другий рівень виконує головний процес, який відповідає за виконання системних операцій. Саме у цьому процесі виконуються виклики бібліотек для отримання інформації про апаратне забезпечення, обробляються запити від інтерфейсу користувача та запускаються модулі аналізу даних. Головний процес також забезпечує взаємодію із зовнішніми сервісами, зокрема із системою формування рекомендації на основі мовної моделі.

Взаємодія між головним та рендер процесами здійснюється через спеціальний механізм міжпроцесної взаємодії. Для підвищення рівня безпеки у застосунку використовується проміжний рівень preload, який виконує роль контрольованого мосту між користувацьким інтерфейсом і системним середовищем Node.js. У preload-модулі реалізовано механізм мосту, який дозволяє

					КвРКІ 022082.22.02.67 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

експортувати обмежений набір функцій у глобальний простір браузерного середовища. У результаті інтерфейс користувача отримує доступ лише до визначених даних, представлених методами `windows.pc.scan()`, `windows.pc.thermals()` та `windows.pc.recommendations(payload)`. Така модель взаємодії дозволяє ізолювати графічний інтерфейс від прямого доступу до системних ресурсів, що значно зменшує ризики несанкціонованого використання системних функцій [59].

У програмно-апаратному аспекті система складається з кількох функціональних модулів, кожен з яких виконує окрему роль у процесі роботи застосунку. Першим із них є модуль сканування апаратних характеристик. Основним завданням цього модуля є отримання інформації про конфігурацію комп'ютера. Для реалізації цього завдання використовується бібліотека `systeminformation`, яка надає доступ до даних про операційну систему, центральний процесор, оперативну пам'ять графічні адаптери та накопичувачі. Отримані дані проходять етап нормалізації, під час якого перевіряється коректність форматів, узгоджуються одиниці вимірювання та формується структурований профіль системи.

Другим важливим компонентом системи є модуль термального моніторингу. Його функція полягає у регулярному опитуванні датчиків температури центрального та графічних процесорів. Отримані показники зберігаються у внутрішній історії вимірювань, що дозволяє аналізувати їх зміну у часі. На основі накопичених даних визначаються поточні, середні та максимальні значення температури, які відображаються у вигляді графіків у користувацькому інтерфейсі. Такий підхід дозволяє оцінити не лише поточний стан системи, але й її поведінку під час різних режимів навантаження.

Наступним компонентом є модуль візуалізації, який відповідає за представлення результатів у зручній для користувача формі. У межах цього модуля реалізовано систему інформаційних карток, які відображають ключові параметри комп'ютера, а також графічні елементи для візуалізації температурних

					КвРКІ 022082.22.02.67 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

змін. Побудова графіків здійснюється на основі HTML-елемента canvas, що дозволяє ефективно відображати часову динаміку температурних показників.

Окрему роль у структурі системи відіграє модуль рекомендацій, який реалізує механізм аналітичної підтримки користувача. У цьому модулі використовується гібридний підхід до формування результатів. З одного боку застосовується зовнішній сервіс через GroqAPI. З іншого боку реалізовано локальний механізм аналізу, який працює незалежно від наявності мережевого з'єднання. Такий підхід дозволяє забезпечити відмовостійкість системи, оскільки навіть у випадку недоступності зовнішнього сервісу користувач отримає набір рекомендацій на основі локального аналізу параметрів системи [65-68].

Важливим елементом системи є модуль керування станом користувача. Його функція полягає у збереженні та відновленні результатів сканування. Для цього використовується механізм localStorage, який дозволяє зберігати профілі конфігурації комп'ютера безпосередньо у середовищі застосунку. Завдяки цьому користувач може виконувати повторні експерименти, порівнювати результати різних вимірювань та аналізувати зміни системи після модернізації або обслуговування.

З точки зору обробки інформації програмно-технічний засіб працює за послідовною схемою обробки даних. На першому етапі виконується збір первинної інформації про систему. На другому етапі проводиться нормалізація даних, яка включає перевірку діапазонів значень, приведення одиниць вимірювання та усунення потенційних помилок форматування. На третьому етапі виконуються обчислення похідних параметрів, зокрема визначення обсягу оперативної пам'яті у гігабайтах, кількості встановлених модулів пам'яті та показників температур. На завершальному етапі результати обробки зображені у графічному інтерфейсі та передаються до модуля формування рекомендацій.

Система рекомендацій базується на принципі гібридної аналітичної моделі. Якщо доступний зовнішній сервіс штучного інтелекту, виконується аналіз профілю комп'ютера з використанням мовної моделі. У разі відсутності доступу

					КвРКІ 022082.22.02.67 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

до API активується локальний механізм аналізу, який базується на наборі інженерних правил. Розглянуті правила враховують температурні пороги для центрального та графічного процесорів, мінімальні рекомендовані значення оперативної пам'яті, тип накопичувача та інші параметри системи. У результаті незалежно від умов роботи користувач отримує завершений набір рекомендацій, що підвищує практичність застосування [60-64].

Розроблений засіб підтримує кілька сценаріїв практичного використання. До них належить первина діагностика комп'ютера, перевірка температурного режиму, а також сканування системи до можливості модернізації апаратних компонентів, а завдяки збереженню профілю можна виконати порівняльний аналіз системи до та після технічного обслуговування, чи покращення. Крім того, застосунок може використовуватися у навчальних цілях для демонстрації принципів віддаленого моніторингу апаратних параметрів комп'ютера.

Серед основних властивостей реалізованого засобу можна виділити інформативність, оперативність отримання та сканування даних, зручність використання, а також відмовостійкість. Інформативність забезпечується поєднанням показників із фіксованим інтервалом опитування, графічної візуалізації та ШІ-рекомендацій. Оперативність досягається шляхом сканування температурних показників із фіксованим інтервалом опитування. Відмовостійкість системи забезпечується наявністю резервних механізмів отримання даних і додаткового режиму формування рекомендацій. Зручність використання досягається завдяки зрозумілому інтерфейсу та можливості ручного доповнення або коригування параметрів системи.

Але як і всі інші програмно-технічний засіб має певні обмеження застосування. Точність температурних даних залежить від доступності апаратних сенсорів і драйверів операційної системи. Деякі платформи можуть обмежувати доступ до окремих датчиків, що впливає на якість отриманих даних. Крім того, сформовані системою рекомендації мають сумнівний характер і потребують додаткової перевірки.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, реалізований програмно-технічний засіб HELPERPC є системою моніторингу та аналітичної підтримки користувача ПК. Завдяки поєднанню функції сканування, аналізу показників, візуалізації результатів та інтелектуального формування рекомендацій, забезпечується комплексна оцінка технічного стану комп'ютера та сприяє прийняттю обґрунтованих рішень щодо оптимізації або покращень. Модульна архітектура застосунку і безпечна модель взаємодії процесів та гібридний підхід до формування рекомендацій роблять цей додаток приємним як для повсякденного використання, так і для застосування у межах навчання.

3.2 Програмна реалізація ключових модулів

Програмна реалізація програмно-технічного засобу HELPERPC побудована за модульним принципом, відповідно до якого кожен компонент системи виконує чітко поставлену функцію в межах процесу обробки даних. Такий процес можна показати у вигляді послідовних дій: отримання даних → обробка та нормалізація → візуалізація → формування рекомендацій. Використання такого типу архітектури дозволяє підвищити надійність застосунку, спростити супровід, а також забезпечити можливість подальшого розширення функціональності без суттєвих змін у структурі всієї системи.

Базова реалізація включає низку ключових файлів, кожен з яких відповідає за окремий функціональний блок системи. До основних компонентів належать такі модулі: `index.js`, що реалізує головний процес застосунку; `preload.js`, який забезпечує безпечний зв'язок між інтерфейсом користувача та системною логікою; `scanner.js`, призначений для отримання апаратних характеристик комп'ютера та температур; `ai-advisor.js`, що формує рекомендації щодо оптимізації системи; `main.js`, який реалізує прикладну логіку інтерфейсу користувача; а також `index.html`, що визначає структуру графічного інтерфейсу.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2.1 Модуль ініціалізації та організації міжпроцесної взаємодії

Файл `index.js` реалізує головний процес середовища Electron і виступає центральною точкою керування застосунком. Основним призначенням цього модуля є створення головного вікна застосунку, налаштування параметрів його роботи та організація взаємодії між інтерфейсом користувача і системними модулями.

Під час запуску застосунку створюється вікно у якому завантажується інтерфейс користувача. Для підвищення рівня безпеки застосовуються спеціальні параметри, зокрема ізоляція контексту та заборона прямої інтеграції Node.js у браузерне середовище. Такий підхід дозволяє відокремити інтерфейс користувача від системних функцій операційної системи та зменшує ризик виконання стороннього коду.

Крім створення головного вікна, виконується реєстрація обробників міжпроцесної взаємодії. Саме через ці обробники інтерфейс користувача взаємодіє з логікою системи. У застосунку реалізовано декілька каналів обміну повідомленнями. Канал `scan-system`, використовується для запуску сканування апаратних характеристик системи. Канал `scan-thermal`, призначений для отримання поточних температур, також канал `ai` використовується для формування рекомендацій на основі профілю системи.

Для реалізації безпечної взаємодії між графічним інтерфейсом та головним процесом використовується модуль `preload.js`, який виступає проміжним мостом між шарами системи. Основою його реалізації є механізм що дозволяє експортувати обмежений набір функцій у глобальний простір браузерного середовища. У результаті виконання цього скрипта у вікні браузера формується об'єкт який надає інтерфейсу доступ до функцій сканування і рекомендацій. Такі функції виступають проксі-методами, які передають запити до головного процесу через механізми IPC.

Завдяки такій архітектурі інтерфейс користувача не отримує прямого доступу до API, що відповідає принципу мінімальних привілеїв та значно підвищує безпеку виконання застосунку.

3.2.2 Модуль апаратного сканування та термального моніторингу

Основний механізм отримання технічної інформації про комп'ютерну систему реалізовано у файлі scanner.js. Для доступу до апаратних характеристик використовується бібліотека systeminformation, яка надає можливість отримувати дані про операційну систему, центральний процесор, оперативну пам'ять, графічні адаптери та накопичувачі.

Під час виконання процедури сканування модуль паралельно отримує низку системних параметрів. До них належать характеристики операційної система, модель і температура центрального процесора, параметри оперативної пам'яті, конфігурація встановлених модулів RAM, характеристики графічних адаптерів та конфігурація накопичувачів даних.

Після отримання первинних даних виконується їх нормалізація. На цьому етапі здійснюється перевірка коректності форматів, приведення значень до стандартів. Зокрема температурні значення очищуються від текстових символів, результати перевіряються на фізично можливий діапазон температур, а також коми у числових значеннях замінюються на крапки.

Окремим етапом є визначення графічної підсистеми. Для цього аналізується весь масив контролерів, які містять інформацію про всі доступні графічні адаптери. На основі даних про виробника та модель формується підпис відеокарти, який є зрозумілим для користувача.

На завершальному етапі формується повний профіль комп'ютера. У ньому параметри оперативної пам'яті, накопичувачів та відеокарти переводяться у гігабайти, також рахується кількість наявних модулів пам'яті та створюється JSON-об'єкт, який використовується іншими модулями системи [71-72].

					КвРКІ 022082.22.02.67 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

У межах цього ж модуля реалізовано функцію сканування температур, яка відповідає за оперативне їх зчитування. На відміну від повного зчитування системи, ця функція отримує лише обмежений набір параметрів, необхідних для побудови температурних графіків у реальному часі. Вона повертає температуру центрального процесора і графічного процесора, також часову мітку та службову інформацію у якій одиниці виконувалося вимірювання.

Особливістю реалізації є використання резервного механізму отримання температури процесора. Якщо при використанні стандартної бібліотеки не повертається валідний результат, застосовується альтернативний метод читання сенсорів через системні інтерфейси Windows. Це підвищує надійність програмно-технічного засобу, для критично важливих даних.

3.2.3 Модуль інтелектуального аналізу та формування рекомендацій

Файл ai-advisor.js реалізує механізм аналізу отриманих параметрів системи та формування рекомендації користувача. Модуль використовує гібридний підхід, що поєднує локальний аналіз та використання повноцінного ШІ.

На першому етапі формується компактний набір вхідних даних, який включає основні характеристики комп'ютера, поточні температурні показники та статичні параметри попередніх сканувань. Після цього використовується механізм локального аналізу, що базується на наборі різних інженерних правил. Вони включають аналіз температурних порогів процесора та відеокарти, перевірку обсягу оперативної пам'яті і її достатності, визначення кількості модулів RAM та аналіз типу накопичувача.

Наприклад, якщо система виявляє, що обсяг пам'яті є меншим за рекомендовані значення, формується рекомендація щодо модернізації. Аналогічно, у випадку використання одного модуля пам'яті система пропонує перехід до двоканального режиму роботи, із метою підвищення продуктивності.

Після виконання локального аналізу виконується звернення до штучного інтелекту через Groq API. У запиті використовується мовна модуль, яка аналізує комп'ютера та генерує структурований список рекомендацій.

У випадку недоступності мережі або виникнення помилки під час виклику система автоматично повертається до локального механізму формування рекомендацій. Після отримання результатів виконується їх нормалізація: видаляються дублікати, усуваються порожні елементи та обмежується максимальна кількість рекомендацій.

3.2.4 Модуль інтерфейсу користувача та представлення даних

Прикладна логіка роботи інтерфейсу реалізована у файлі main.js . У цьому модулі обробляються дії користувача, виконуються запити до системних функцій та здійснюється оновлення компонентів інтерфейсу.

У модулі реалізовано систему керування станом застосунку, яка включає профіль користувача його комп'ютера, історію температурних вимірювань та службові прапори стану виконання запитів. Під час запуску програмно-технічного засобу виконується автоматичне сканування системи, після чого отримані параметри відображаються у відповідних інформаційних картках інтерфейсу.

Для забезпечення безперервного моніторингу температур використовується механізм періодичного опитування сенсорів. Через задані інтервали часу викликається функція отримання даних, після чого вони додаються до історії вимірювань. На основі накопиченої історії обчислюються середні та максимальні температури, що використовуються як для візуалізації, так і для формування рекомендацій.

Графічна візуалізація реалізована за допомогою елемента canvas, у якому будується система координат та відображається температурна крива. Останє виміряне значення позначається окремим маркером. Зо дозволяє користувачу легко і швидко оцінити поточний стан системи.

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

Структура графічного інтерфейсу визначається у файлі index.html, де інтерфейс поділено на кілька блоків. Перший блок призначений для відображення результатів сканування ПК, другий реалізує температурний моніторинг, а третій відображає рекомендації щодо оптимізації системи. Крім головних функцій, інтерфейс містить службове меню роботи з профілями системи, яке дозволяє зберігати результати сканування, завантажувати попередні профілі та очищати збережені дані.

3.3 Режими роботи програмно-технічного засобу

Розроблений програмно-технічний засіб HELPERPC функціонує як багаторежимна система, що забезпечує послідовний перехід від первинного отримання апаратної інформації до безперервного віддаленого моніторингу технічного стану ПК та формування практичних рекомендацій щодо оптимізації. Організація режимів роботи побудована таким чином, щоб користувач міг виконати повний цикл аналізу системи від автоматичного зчитування параметрів конфігурації до отримання аналітичних висновків про стан обладнання. Кожен режим має власне функціональне призначення, набір вхідних даних та результати виконання, які використовуються наступними компонентами системи. Завдяки цьому забезпечується узгоджена взаємодія між модулями програмного засобу та послідовна обробка інформації на всіх етапах роботи застосунку. Реалізована структура дозволяє поєднати функції збору, аналізу, відображення та інтерпретації системних параметрів у межах єдиного програмного середовища. Зовнішній вигляд програми зображено на Рис.3.1:

					КвРКІ 022082.22.02.67 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

У ході виконання цього режиму здійснюється паралельне зчитування інформації про операційну систему, центральний процесор, температуру процесора, обсяг та конфігурації оперативної пам'яті, графічні адаптери та накопичувачі. Отримані значення проходять етап нормалізація, під час якого перевіряється коректність форматів і виконується переведення отриманих параметрів до стандартних одиниць вимірювання. Після цього формуються похідні характеристики, зокрема загальний обсяг оперативної пам'яті, кількість модулів, обсяг відеопам'яті та опис доступних накопичувачів. Сформований профіль системи передається до інтерфейсу користувача та відображається у потрібних інформаційних картках. У разі відсутності використовуються контрольовані резервні значення, або введені користувачем за допомогою полів вводу, що запобігає аварійному завершенню роботи застосунку. Дану частину зображено на Рис.3.3:



Рисунок 3.3 – Результат сканування системи

Після завершення автоматичного сканування користувач може перейти до режиму ручного редагування та уточнення параметрів конфігурації. Цей режим призначений для корекції даних у випадках, коли автоматичне визначення

параметрів виконано не повністю або коли користувач бажає змоделювати іншу конфігурацію для подальшого аналізу. У режимі редагування відкривається форма ручного введення параметрів, у якій користувач може змінювати різні характеристики, від процесора, графічного адаптера, обсягу оперативної пам'яті, типу накопичувача, до температурних показників та інших доступних параметрів системи. Дану частину зображено на Рис. 3.4:

The screenshot shows a web interface for system configuration. At the top left is the logo 'HELPERPC' and at the top right is 'SYSTEM THERMAL HELP LOGIN'. Below these are two summary boxes: one for hardware (Sticks: 2/4, SSD: NVMe WD PC SN740 SDDQNQD-512G-1014 477 GB, Screen: 1536x864) and one for system info (Browser: Electron, Device: PC, Platform: Windows). A central 'Enter Manually' section contains several input fields with example values: Processor (Intel i7-12700K), Graphics Card (RTX 4090), RAM (GB) (32), SSD (Samsung 990 Pro 1TB), CPU Frequency (GHz) (3.5), VRAM (GB) (6), CPU Temperature (65), GPU Temperature (70), RAM Sticks (1-4) (2), and Device Type (Laptop). A green 'Save' button is located at the bottom center.

Рисунок 3.4 – Ручне введення даних системи

Після збереження змін нові значення інтегруються з автоматично отриманими значеннями та формують повний звіт за допомогою них. Оновлена інформація зберігається у локальному сховищі браузера, після чого відбувається повне оновлення потрібних карток та рекомендацій на основі нових даних. Також передбачено механізм валідації та синхронізації збережених даних, що дозволяє уникнути конфліктів між внесеними користувачем та автоматично зібраними значеннями. Під час інтеграції виконується перевірка цілісності структури даних, а також їх актуальності, щоб у звіті відображалась лише коректна та релевантна інформація. Це забезпечує стабільність формування підсумкових результатів і підвищує точність подальшого аналізу, зокрема при побудові рекомендацій та

графічного відображення стану системи. Збереження даних сканування зображено на Рис. 3.5:



Рисунок 3.5 – Збереження даних сканування

Наступним функціональним елементом є режим безперервного термального моніторингу, який забезпечує контроль температурних показників процесора та графічного адаптера у реальному часі. Після успішного сканування або завантаження профілю системи активується механізм періодичного опитування температурних сенсорів із фіксованим інтервалом часу. Під час кожного циклу опитування температурних сенсорів із певним інтервалом часу. Під час кожного циклу опитування система опитує актуальні температурні значення через відповідний IPC-виклик, після чого ці дані додаються до історії вимірювань. Історія температур використовується для обчислення поточних, середніх та максимальних значень температури, а також для побудови графіків температурної динаміки. Графічна візуалізація реалізується за допомогою canvas, що дозволяє відображати зміну температур у часовому інтервалі та оперативно реагувати на наявні перегріву системи. Навіть у випадку відсутності доступу до сенсорних даних застосунок залишається працездатним завдяки використанню

резервних температурних значень. Частину застосунку із графіками зображено на Рис. 3.6:

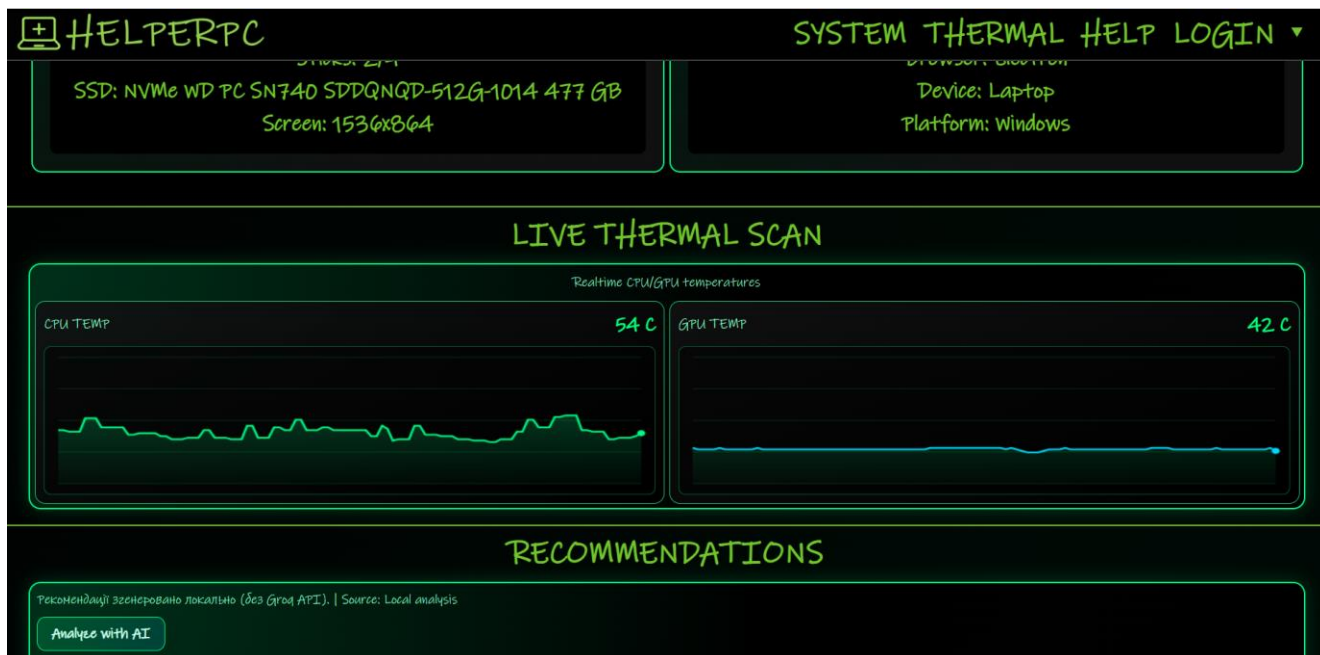


Рисунок 3.6 – Термальні графіки

Окремим функціональним режимом роботи системи є режим аналітики та формування рекомендацій, який забезпечує інтерпретацію технічного стану комп'ютера. Для виконання аналізу формується пакет даних, що включає апаратні характеристики системи, поточні температурні показники та статичні параметри температурної історії. Отриманий набір даних параметрів передається до модуля формування рекомендації, де спочатку застосовується локальний механізм аналізу, що базується на наборі інженерних правил. Вказані правила враховують температурні пороки процесора та графічного адаптера, достатність обсягу оперативної пам'яті, кількість модулів RAM та тип накопичувача. Після виконання локального аналізу, за наявності доступу до мережі, система може виконати додатковий запит до зовнішнього сервісу штучного інтелекту через Groq API. Отримана відповідь перевіряється на коректність і використовується для формування структурованого списку рекомендацій. У випадку недоступності зовнішнього сервісу застосунок автоматично повертається до локального

					КвРКІ 022082.22.02.67 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

системні інтерфейси операційної системи, а також використовується локальний механізм формування рекомендацій замість хмарного аналізу. Завдяки цьому навіть у разі виникнення помилок або обмежень доступу до сенсорних даних застосунок продовжує надавати користувачу корисну інформацію про стан системи.

Таким чином реалізовані режими роботи формують цілісну операційну модель програмно-технічного засобу HELPERPC, яка формує автоматичне отримання даних про апаратну конфігурацію, можливість ручного коригування параметрів, безперервний віддалений моніторинг температурних показників та інтелектуальний аналіз технічного стану системи. Такий підхід забезпечує баланс між автоматизацією процесів, контролем з боку користувача та надійністю роботи програмного засобу, що робить його придатним для практичного використання під час діагностики та оптимізації персональних комп'ютерів.

3.4 Таблиці із результатами

Метою експериментальної перевірки є підтвердження працездатності алгоритмів автоматичного визначення конфігурації персонального комп'ютера, модуля моніторингу температурних показників, а також системи формування рекомендацій щодо оптимізації роботи апаратних компонентів. Окрему увагу приділено перевірці коректності взаємодії між модулями системи, стабільності отримання даних у реальному часі та точності обробки зібраної інформації, що безпосередньо впливає на якість подальших аналітичних висновків.

Для забезпечення об'єктивності дослідження тестування проводилося на декількох комп'ютерних системах різних класів, включаючи офісний персональний комп'ютер, високопродуктивну ігрову систему та сучасний ноутбук. Такий підхід дозволяє оцінити роботу програмного засобу в умовах різних апаратних архітектур та типових сценаріїв виконання, а також перевірити адаптивність алгоритмів до різних рівнів навантаження, конфігурацій охолодження та характеристик компонентів.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Результати автоматичного сканування конфігурації ПК

№	Параметр	ПК-1 (базовий)	ПК-2 (ігровий)	ПК-3 (ноутбук)
1	ОС	Windows 10 Pro	Windows 11 Pro	Windows 11 Home
2	CPU	Intel Core i3-12100f	AMD Ryzen 5 5600G	Intel Core i5-13420h
3	Потоки CPU	8	12	12
4	Частота CPU, GHz	3.3	4.4	2.1
5	GPU (дискретна)	GTX1050TI	RTX 4060	Nvidia RTX3050(laptop)
6	GPU (інтегрована)	-	Radeon UHD	Intel UHD
7	VRAM, GB	4	8	6
8	RAM, GB	16	64	32
9	К-сть планок RAM	2	2	2
10	SSD	SATA 512 GB	NVMe 1 TB	NVMe 477 GB

Таблиця 3.1 демонструє результати автоматичного визначення конфігурації персональних комп'ютерів різних типів за допомогою програмно-технічного засобу. Отримані дані підтверджують, що система конкретно зчитує як базові характеристики програмного середовища, так і детальні апаратні параметри, включаючи тип процесора, кількість обчислювальних ядер, тактову частоту, конфігурацію оперативної пам'яті та параметри графічної підсистеми. У процесі тестування було використано три різні типи пристроїв: офісний персональний комп'ютер, високопродуктивну ігрову систему та сучасний ноутбук, що дозволило перевірити роботу алгоритмів у різних умовах апаратної архітектури.

Особливу увагу привертає коректне визначення як дискретних, так і інтегрованих графічних адаптерів, а також обсягу відеопам'яті, що є важливим фактором для подальшого аналізу продуктивності системи. Крім того, програмно-технічний засіб правильно визначає кількість встановлених модулів оперативної пам'яті, що дозволяє оцінити режим її роботи та потенційні можливості

оперативної пам'яті модуль рекомендації пропонує варіанти модернізації системи, що свідчить про адекватність логіки аналізу апаратних параметрів.

Окрему увагу під тестування було виділено перевірці роботи системи у випадку відсутності підключення до віддаленого ШІ-сервісу. У таких ситуаціях програмно-технічний засіб автоматично переходить до використання локального алгоритму формування рекомендацій, що забезпечує безперервність функціонування. Таким чином, результати тестування підтверджують відмовостійкість реалізованої структури та її здатність стабільно працювати навіть у разі виникнення зовнішніх технічних обмежень.

Таблиця 3.4. – Функціональне тестування режимів роботи

№	Функція	Тестова дія	Очікуваний результат	Фактичний результат
1	Scan PC	Натиснути кнопку сканування	Заповнення всіх карток	Виконано
2	Edit/Save	Внести ручні дані та зберегти	Оновлення карток і рекомендацій	Виконано
3	Live Thermal	Спостереження 2-3 цикли	Оновлення кожні 5 с	Виконано
4	Analyze with AI	Запуск аналізу	6 рекомендацій + summary	Виконано
5	Save/Load	Зберегти і завантажити профіль	Відновлення стану	Виконано
6	Clear	Очистити дані	Скидання інтерфейсу	Виконано

Функціональне тестування, результати якого наведені у таблиці 3.4, було спрямоване на перевірку коректності роботи основних модулів програмного засобу. У процесі тестування було перевірено ключові сценарії взаємодії користувача з системою, включаючи автоматичне сканування конфігурації

комп'ютера, ручне редагування параметрів, запуск модуля аналізу та генерації рекомендацій, а також функції збереження та відновлення профілів.

Отримані результати свідчать про стабільну роботу програмного забезпечення та відсутність критичних помилок під час виконання операцій. Особливо важливо, щоб процес автоматичного сканування конфігурації коректно заповнював інформаційні картки апаратних параметрів, що програмно-технічний засіб і виконує. Модуль термального моніторингу демонструє стабільне оновлення даних із заданим часовим інтервалом, що дозволяє користувачу спостерігати зміну температурних показників у режимі реального часу.

Таблиця 3.5. – Перевірка механізмів і надійності

Ситуація	Очікувана поведінка системи	Фактична поведінка
Немає валідної CPU temperature із SI	Спроба через WMI	Виконано
Відсутня температура GPU	Підстановка резервного значення	Виконано
Недоступний Groq API	Перехід до локальних рекомендацій	Виконано
Немає інтернету	Робота локальної логіки без зупинки UI	Виконано

Результати перевірки механізмів відмовостійкості, приведені у таблиці 3.5, демонструють здатність програмного засобу стабільно працювати навіть у випадку виникнення технічних проблем. Під час тестування було змодельовано кілька типових проблемних сценаріїв, таких як відсутність коректних температурних даних від сенсорів, недоступність зовнішнього ШІ або інтернет-з'єднання.

У випадку неможливості отримання показників центрального процесора через стандартну бібліотеку системної інформації система автоматично використовує альтернативні джерела даних операційної систем. Якщо температурні показники графічного адаптера недоступні, інтерфейс відображає

резервне значення або відповідну позначку відсутності даних, що запобігає некоректному відображенню інформації.

Також було підтверджено, що у випадку недоступності зовнішнього сервісу штучного інтелекту програмно-технічний засіб переходить до використання локального модуля генерації рекомендацій. Такий підхід дозволяє уникнути повної втрати функціональності системи та забезпечує надійність та стабільність роботи інтерфейсу. Отримані результати свідчать про наявність продуманої архітектури обробки помилок та підтверджують надійність розробленого програмного рішення.

3.5 Тестування програмно-технічного засобу

Оцінка продуктивності програмно-технічного засобу HELPERPC виконується із метою визначення ефективності його роботи в умовах реального навантаження. Даний етап тестування дозволяє оцінити, наскільки швидко застосунок виконує сканування апаратної конфігурації, оновлює показники температур, а також як він використовує системні ресурси, зокрема оперативну пам'ять і процесорний час [78-80].

Окремо враховується час відповіді від зовнішнього ШІ-сервісу, оскільки затримки на цьому етапі безпосередньо впливають на загальну швидкодію системи. Таким чином результати дозволяються зробити висновок щодо придатності застосунку до використання в режимі реального часу без суттєвого навантаження на систему. Додатково оцінюється стабільність функціонування програмного засобу під час тривалої роботи та повторного виконання процедур сканування. Це дозволяє визначити рівень надійності застосунку, ефективність використання системних ресурсів і коректність роботи модулів у різних режимах навантаження.

Таблиця 3.6. – Оцінка продуктивності HELPERPC

					КвРКІ 022082.22.02.67 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

Параметр	HELPERPC	Еталон	Відхилення
CPU	Ryzen 5 5600G	CPU-Z	0
RAM	32 GB	HWiNFO	0
GPU	RTX 4060	GPU-Z	0
SSD	Kingston NVMe2 1 TB	AIDA64	збіг
CPU Temp	51 °C	49 °C	+2 °C
GPU Temp	54 °C	55 °C	-1 °C

Таблиця 3.9 – Тестовий ПК №3

Параметр	HELPERPC	Еталон	Відхилення
CPU	I5-13420h	CPU-Z	0
RAM	32 GB	HWiNFO	0
GPU	RTX350 TI (Laptop)	GPU-Z	0
SSD	WD SN530 NVMe 512 GB	AIDA64	збіг
CPU Temp	67 °C	64 °C	+3 °C
GPU Temp	61 °C	60 °C	+1 °C

Порівняльний аналіз з еталонними утилітами підтвердив високу точність отриманих результатів. Незначні відхилення знаходяться в допустимих межах і пояснюються використанням різних джерел сенсорних даних та методів їх зчитування. Це свідчить про коректність реалізації алгоритмів нормалізації та обробки системної інформації. Також встановлено, що програмно-технічний засіб має прийнятні показники продуктивності та не створює надто суттєвого навантаження на систему користувача. Реалізований механізм відмовостійкості забезпечує стабільну роботу застосунку навіть у разі недоступності ШІ-сервісу.

3.6 Висновок до третього розділу

					КвРКІ 022082.22.02.67 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

У третьому розділі було розглянуто програмну реалізацію розробленого програмно-технічного засобу HELPERPC а особливості його архітектурної побудови. Детально описано структуру системи, принципи взаємодія між модулями, а також механізми отримання, обробки та подальшого аналізу системної інформації. Особливу увагу надано організації міжпроцесної взаємодії у середовищі Electron, що забезпечує безпечну та стабільну комунікацію між графічним інтерфейсом та системними компонентами. Реалізація такої взаємодії дозволила забезпечити чітке розмежування відповідальності між окремими частинами системи та знизити залежність між її основними модулями.

У межах розділу проаналізовано ключові функціональні модулі системи, зокрема модулів ініціалізації застосунку, апаратного сканування, термального моніторингу, інтелектуального аналізу параметрів комп'ютера та формування рекомендацій щодо оптимізації роботи програмно-технічного засобу. Окремо розглянуто механізми нормалізації отриманих даних, формування узагальненого профілю комп'ютера та організації безперервного моніторингу температурних показників апаратних компонентів. Також описано структуру графічного інтерфейсу користувача, принципи відображення системної інформації та способи візуалізації результатів аналізу. Узгоджена взаємодія між зазначеними модулями забезпечується через єдину логіку обробки даних, що дозволяє підтримувати послідовність виконання операцій та цілісність інформаційного потоку. Також описано структуру графічного інтерфейсу користувача, принципи відображення системної інформації та способи візуалізації результатів аналізу. Застосовані підходи до організації модулів і подання даних забезпечують логічну узгодженість між отриманою інформацією та її інтерпретацією в інтерфейсі користувача.

Запропонована структура взаємодії модулів забезпечує узгоджений та послідовний потік даних на всіх етапах обробки інформації від первинного збору системних параметрів до формування кінцевих рекомендацій. Кожен модуль

					КвРКІ 022082.22.02.67 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

виконує чітко визначену функцію що дозволяє уникнути дублювання логіки та підвищує структурованість програмного коду. Такий підхід сприяє зниженню залежностей між компонентами системи та спрощує їх подальше тестування й супровід. Окремі етапи обробки даних можуть функціонувати незалежно що підвищує гнучкість архітектури. Завдяки цьому забезпечується стабільність роботи застосунку навіть при зміні або розширенні окремих модулів. Отримана модель взаємодії компонентів формує основу для подальшого масштабування системи.

Додатково слід відзначити, що реалізована архітектура HELPERPC має модульний характер, що забезпечує її масштабованість та можливість подальшого розширення функціональності без суттєвих змін у вже існуючій структурі. Завдяки цьому нові компоненти, зокрема додаткові аналітичні алгоритми або розширені модулі моніторингу, можуть інтегруватися у систему з мінімальними витратами на адаптацію. Такий підхід підвищує гнучкість програмного засобу та дозволяє використовувати його як основу для подальших дослідницьких і прикладних розробок у сфері діагностики та оптимізації комп'ютерних систем.

У результаті виконаної роботи сформовано цілісну програмну архітектуру застосунку, яка забезпечує автоматичне отримання апаратних характеристик комп'ютера, аналіз технічного стану системи та генерації рекомендацій для користувача щодо покращення її роботи. Реалізовані механізми взаємодії між програмними модулями, використання безпечної моделі доступу до системних функцій, а також застосування гібридного підходу до аналізу даних сприяють підвищенню надійності, стабільності та функціональності програмного засобу. Отримані результати підтверджують практичну придатність розробленого програмно-технічного засобу для використання під час діагностики, моніторингу та оптимізації роботи ПК.

					КвРКІ 022082.22.02.67 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У рамках виконання дипломної роботи було проведено комплексне дослідження та реалізацію програмно-технічного засобу HELPERPC, призначеного для оцінки технічного стану персонального комп'ютера, віддаленого моніторингу їх апаратних ресурсів та формування практичних рекомендацій щодо оптимізації й модернізації системи. Дослідження включало аналіз предметної області, оцінку існуючих методів діагностики ПК, проектування архітектури програмної системи, розробку та тестування модулів системи, а також перевірку працездатності програмно-технічного засобу в умовах експлуатації.

На етапі аналізу предметної області було виявлено ключові проблеми сучасних підходів, а саме: обмежена автоматизація збору параметрів, недостатня кількість даних із сенсорів, відсутність зручного інтерфейсу для візуалізації та прогнозування технічного стану. Визначено основні вимоги до нового програмного засобу: багаторежимність роботи, підтримка автоматичного та ручного введення даних, безпечна взаємодія інтерфейсу з системними компонентами, можливість безперервного моніторингу та формування рекомендацій на основі сучасних алгоритмів аналізу.

1. Проаналізовано сучасні програмні засоби для діагностики та моніторингу персональних комп'ютерів і визначено їх основні функціональні можливості. У результаті аналізу виявлено особливості існуючих рішень, їх переваги та обмеження, а також визначено напрями вдосконалення функцій моніторингу й аналізу системних параметрів.

2. Досліджено методи отримання апаратної інформації з операційної системи та апаратних інтерфейсів. Розглянуто механізми використання WMI, SMBIOS, ACPI та інших системних джерел для отримання характеристик комп'ютерних компонентів і температурних показників.

					КвРКІ 022082.22.02.67 ПЗ	Арк. 74
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Розроблено архітектуру програмного засобу для збору, обробки та нормалізації даних про апаратне забезпечення ПК. Реалізована структура системи побудована за модульним принципом, що забезпечує розподіл функцій між окремими компонентами та спрощує подальше розширення функціональності.

4. Реалізовано модуль збору системної інформації та модуль аналізу параметрів продуктивності й температурного стану компонентів. Забезпечено автоматичне отримання даних про процесор, оперативну пам'ять, графічний адаптер і накопичувачі, а також виконання моніторингу температурних показників системи.

5. Розроблено алгоритми оцінювання стану системи та формування рекомендацій щодо оптимізації та модернізації ПК. Алгоритми поєднують локальний rule-based аналіз та використання можливостей штучного інтелекту для формування узагальнених рекомендацій користувачу.

6. Проведено тестування програмного засобу з використанням схожих утиліт та виконано аналіз точності отриманих результатів. У процесі тестування здійснено порівняння характеристик і температурних показників із результатами сторонніх програмних засобів моніторингу.

7. Оцінено якість роботи програмно-технічного засобу та сформовано таблиці з результатами. Отримані результати підтвердили працездатність розробленої системи, коректність функціонування основних модулів і придатність застосунку для моніторингу та аналізу стану персонального комп'ютера.

У процесі проєктування та реалізації системи була розроблена архітектура, яка забезпечує чіткий поділ функцій між модулями, безпечну міжпроцесну взаємодію та гнучкість у масштабуванні. Основні модулі системи реалізують ініціалізацію застосунку, безпечну взаємодію між інтерфейсом і системною логікою, а також апаратне сканування та нормалізацію отриманих даних. Використання середовища Electron з увімкненими механізмами ізоляції контексту

та обмеження доступу до Node.js гарантує безпеку роботи застосунку та захист від потенційного шкідливого коду.

Реалізовані функціональні режими системи дозволяють користувачу виконувати повний цикл взаємодії, що охоплює автоматичне сканування параметрів комп'ютера, ручне редагування профілю системи, безперервний віддалений моніторинг даних процесора та відеокарти, а також формування рекомендацій на основі локального аналізу й алгоритмів штучного інтелекту. Додатково передбачено керування профілями користувача із можливістю збереження, завантаження та скидання стану системи, що підвищує практичну придатність та зручність експлуатації. Система відрізняється високою відмовостійкістю завдяки реалізації резервних механізмів отримання даних у випадку недоступності зовнішніх сервісів.

У межах програмної реалізації було забезпечено високий рівень інтеграції між модулями, використовуючи стандартизований формат обміну на основі JSON. Такий підхід дозволяє зберігати цілісність інформації, зменшує навантаження та забезпечує можливість легкості подальшої модернізації та тестування системи. Особлива увага приділена нормалізації даних, що забезпечує коректне відображення апаратних параметрів, температурних показників та рекомендацій в інтерфейсі користувача.

Проведені тестування та перевірка показали, що система працює стабільно і адекватно реагує на відсутність деяких даних, безпечно обробляє запити до зовнішніх сервісів і забезпечує коректне формування рекомендацій. Інтерфейс користувача відрізняється високою зручністю і наочністю, а модуль візуалізації даних дозволяє легко і швидко оцінити систему на помилки та прийняти обґрунтовані рішення щодо її оптимізації.

Реалізована архітектура є технічно обґрунтованою, функціонально завершеною, безпечною та практично придатною для використання в реальних умовах. Розроблений програмно-технічний засіб HELPERPC може бути рекомендований для користувачів, що прагнуть забезпечити якісну роботу ПК.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Acer Inc. NitroSense програмне забезпечення для моніторингу ноутбуків Acer Nitro. 2025. URL: <https://www.acer.com/> (дата звернення: 12.02.2026)
2. CPU Z утиліта для відображення характеристик процесора та пам'яті. 2025. URL: <https://www.cpubid.com/software/cpu-z.html> (дата звернення: 10.02.2026)
3. MSI Afterburner програмне забезпечення для моніторингу та розгону GPU. 2025. URL: <https://www.msi.com/Landing/afterburner/graphics-cards> (дата звернення: 18.02.2026)
4. Intel Corporation. Intel® 64 and IA-32 Architectures Software Developer's Manual: System Programming Guide. 2024. URL: <https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html> (дата звернення: 05.02.2026)
5. Intel Corporation. Intel® Thermal Design Guidelines for Processors. 2023. URL: <https://www.intel.com/content/www/us/en/design/thermal-solutions.html> (дата звернення: 06.02.2026)
6. AMD. Processor Programming Reference (PPR) for AMD Family Processors- Power and Thermal Management. 2023. URL: <https://www.amd.com/en/support/tech-docs> (дата звернення: 07.02.2026)
7. Що таке і як використовувати NVIDIA GeForce Experience. Notatka. URL: <https://notatka.com.ua/instrukcii/shcho-take-i-iak-vikoristovuvati-nvidia-geforce-experience> (дата звернення: 01.03.2026)
8. Google. Chrome DevTools Documentation - Analyze runtime performance. 2025. URL: <https://developer.chrome.com/docs/devtools/performance/> (дата звернення: 02.03.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 77
Зм.	Арк.	№ докум.	Підпис	Дата		

9. Node.js Foundation. Node.js Process API Documentation (CPU usage, system monitoring). 2025. URL: <https://nodejs.org/api/process.html> (дата звернення: 26.02.2026)

10. WHATWG. High Resolution Time and Performance API Specification. 2024. URL: <https://w3c.github.io/hr-time/> (дата звернення: 27.02.2026)

11. Кондратюк І. О. Застосунок для моніторингу комп'ютерної системи : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти / Харків. нац. ун-т радіоелектроніки, 2025. URL: <https://openarchive.nure.ua/handle/document/32747> (дата звернення: 01.03.2026)

12. NVIDIA Corporation. NVIDIA GeForce Experience. 2025. URL: <https://www.nvidia.com/en-us/geforce/geforce-experience/> (дата звернення: 26.02.2026)

13. UserBenchmark Ltd. UserBenchmark сервіс тестування продуктивності комп'ютерних компонентів. 2025. URL: <https://www.userbenchmark.com/> (дата звернення: 03.03.2026)

14. UL Solutions. 3DMark програмне забезпечення для тестування графічної продуктивності. 2025. URL: <https://benchmarks.ul.com/3dmark> (дата звернення: 07.03.2026)

15. Open Hardware Monitor. Open Hardware Monitor моніторинг температур та напруги. 2025. URL: <https://openhardwaremonitor.org/> (дата звернення: 08.03.2026)

16. W3C. Performance Timeline Specification. 2024. URL: <https://w3c.github.io/performance-timeline/> (дата звернення: 10.03.2026)

17. Electron. Debugging and Profiling Electron Applications in VSCode. 2025. URL: <https://www.electronjs.org/docs/latest/tutorial/debugging-vscode> (дата звернення: 12.03.2026)

18. Grafana Labs. Grafana Documentation - Visualization and Monitoring Platform. 2025. URL: <https://grafana.com/docs/grafana/latest/> (дата звернення: 13.03.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 78
Зм.	Арк.	№ докум.	Підпис	Дата		

19. Парамуд Я. С., Янчинський А. В. Комп'ютерна система оповіщення критичних значень мікроклімату в приміщенні. Львів : Національний університет «Львівська політехніка», 2021. URL: <https://ena.lpnu.ua/handle/ntb/57954> (дата звернення: 14.03.2026)

20. Бочкар'юв О. Ю. Автономна децентралізована система моніторингу комп'ютерної мережі на основі програмних агентів. Львів : Національний університет «Львівська політехніка», 2023. URL: <https://ena.lpnu.ua/handle/ntb/111636> (дата звернення: 15.03.2026)

21. Гриценко О. Метод оцінювання інформаційних моделей спостережуваних процесів у комп'ютерної мережі. Львів : Національний університет «Львівська політехніка», 2014. URL: <https://ena.lpnu.ua/handle/ntb/24663> (дата звернення: 16.03.2026)

22. Боженко В. І. Відтворення теплового зображення на моніторі персонального комп'ютера. Львів : Національний університет «Львівська політехніка», 2009. URL: <https://ena.lpnu.ua/handle/ntb/6049> (дата звернення: 17.03.2026)

23. Погребенник В., Огуряєва О. Принципи побудови комп'ютерних систем опрацювання даних екологічного моніторингу. Львів : Національний університет «Львівська політехніка», 2008. URL: <https://ena.lpnu.ua/handle/ntb/1286> (дата звернення: 17.03.2026)

24. Гадзало О. Я. Інформаційна система аналізу даних комп'ютерних комплектуючих. Львів : Національний університет «Львівська політехніка», 2022. URL: <https://ena.lpnu.ua/items/d030a082-93e9-42c2-86bb-42ec1bc9ecab> (дата звернення: 13.03.2026)

25. Квурт Л. С., Піць В. А. Засоби оцінювання екологічності комп'ютера. Львів : Національний університет «Львівська політехніка», 2009. URL: <https://ena.lpnu.ua/items/adb975f2-5b6b-4feb-b79b-38e8e5d65024> (дата звернення: 14.03.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 79
Зм.	Арк.	№ докум.	Підпис	Дата		

26. Мруць В. М. Система моніторингу атмосферного повітря в місті Львів. Львів : Національний університет «Львівська політехніка», 2023. URL: <https://ena.lpnu.ua/handle/ntb/113740> (дата звернення: 19.03.2026)

27. Intel Corporation. Intel Processor Diagnostic Tool - утиліта для тестування процесорів Intel. 2025. URL: <https://www.intel.com/content/www/us/en/download/15951/intel-processor-diagnostic-tool.html> (дата звернення: 17.03.2026)

28. Piriform. Speccy - утиліта для аналізу апаратного забезпечення комп'ютера. 2026. URL: <https://www.ccleaner.com/speccy> (дата звернення: 20.03.2026)

29. Piriform. CCleaner - утиліта для очищення та оптимізації системи. 2026. URL: <https://www.ccleaner.com/ccleaner> (дата звернення: 20.03.2026)

30. FinalWire Ltd. AIDA64 - утиліта для діагностики, тестування та моніторингу комп'ютерної системи. 2026. URL: <https://www.aida64.com/> (дата звернення: 20.03.2026)

31. AMD. AMD Ryzen Master Utility - програмне забезпечення для моніторингу та налаштування процесорів Ryzen. 2025. URL: <https://www.amd.com/en/technologies/ryzen-master> (дата звернення: 21.03.2026)

32. TechPowerUp. GPU-Z утиліта для моніторингу параметрів відеокарти. 2025. URL: <https://www.techpowerup.com/gpuz/> (дата звернення: 22.03.2026)

33. PassMark Software. PerformanceTest - програмне забезпечення для тестування продуктивності ПК. 2025. URL: <https://www.passmark.com/products/performancetest/> (Дата звернення: 23.03.2026)

34. Crystal Dew World. CrystalDiskInfo - утиліта моніторингу стану накопичувачів. 2025. URL: <https://crystalmark.info/en/software/crystaldiskinfo/> (дата звернення: 23.03.2026)

35. HWMonitor - програмне забезпечення для моніторингу температури та напруг компонентів ПК. 2025. URL: <https://www.cpuid.com/softwares/hwmonitor.html> (дата звернення: 24.03.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 80
Зм.	Арк.	№ докум.	Підпис	Дата		

36. HWiNFO - утиліта для аналізу апаратної конфігурації комп'ютера. 2025. URL: <https://www.hwinfo.com/> (дата звернення: 25.03.2026)
37. Blog IT. Розгортання Electron-додатків для різних платформ. 2023. URL: <https://blog-it.com.ua/rozgortannya-electron-dodatkov-dlya-riznyh-platform/> (дата звернення: 26.03.2026)
38. SiSoftware. Sandra Lite - програмний комплекс для діагностики та аналізу комп'ютерних систем. 2025. URL: <https://www.sisoftware.co.uk/> (дата звернення: 27.03.2026)
39. Mozilla Developer Network. JavaScript Performance Optimization Techniques. 2024. URL: <https://developer.mozilla.org/en-US/docs/Web/Performance> (дата звернення: 28.03.2026)
40. Google Developers. Web Performance Optimization Guide. 2024. URL: <https://web.dev/performance/> (дата звернення: 29.03.2026)
41. React Documentation. Optimizing Performance in React Applications. 2024. URL: <https://react.dev/learn/render-and-commit> (дата звернення: 30.03.2026)
42. GeeksforGeeks. Introduction to ElectronJS - створення кросплатформних настільних застосунків на основі JavaScript, HTML та CSS. 2026. URL: <https://www.geeksforgeeks.org/javascript/introduction-to-electronjs/> (дата звернення: 01.04.2026)
43. W3C. Web Performance Working Group Standards. 2024. URL: <https://www.w3.org/webperf/> (дата звернення: 02.04.2026)
44. Заковоротний О. Ю., Сапальський О. А. Вирішення проблем продуктивності за допомогою асинхронних методів в JavaScript. *Системи управління навігації та зв'язку*. 2025. № 1. URL: <https://repository.kpi.kharkov.ua/items/fdf77170-e7b8-4720-a41a-9791cd5a266f> (дата звернення: 03.04.2026)
45. Ecma International. ECMAScript Language Specification (ECMA-262) - офіційна специфікація мови JavaScript. 2026. URL: <https://tc39.es/ecma262/> (дата звернення: 04.04.2026)

					КвРКІ 022082.22.02.67 ПЗ	Арк. 81
Зм.	Арк.	№ докум.	Підпис	Дата		

46. Ярош О.Л., Бабаков Р. М. Методи оптимізації продуктивності вебзастосунків. 2021. № 2. URL: <https://jait.donnu.edu.ua/article/view/13969> (дата звернення: 04.04.2026)

47. Антонюк А. І., Павловський В. І. Покращення продуктивності та реакції вебсторінок завдяки оптимізації використання GPU в JavaScript. Київ : КПІ ім. Ігоря Сікорського, 2023. URL: <https://ela.kpi.ua/handle/123456789/65479> (дата звернення: 05.04.2026)

48. Приліпа А. О., Філатова Г. Є. Методи оптимізації JavaScript для підвищення продуктивності вебзастосунків. Харків : НТУ ХПІ, 2024. URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/85094> (дата звернення: 06.04.2026)

49. Козуб В. Ю., Семенов М. А., Осадчук С. І. Аналіз впливу WebAssembly на продуктивність складних вебзастосунків. 2025. № 3. С. URL: <https://doi.org/10.35546/kntu2078-4481.2025.3.2.28> (дата звернення: 07.04.2026)

50. Панченко В. І., Любенко А. І. Моніторинг навантаження компонентів персонального комп'ютера : наукова доповідь / Нац. техн. ун-т "Харків. політех. інститут", 2023. URL: <https://repository.kpi.kharkov.ua/entities/publication/4bf7ce14-ed8f-465a-a374-5588e6fdcdb7> (дата звернення: 08.04.2026)

51. Electron. Міжпроцесне спілкування (IPC) у Electron: основні можливості та приклади використання. URL: <https://www.electronjs.org/docs/latest/api/ipc-main> (дата звернення: 09.04.2026)

52. Electron. Рекомендації щодо безпеки при роботі з preload та contextIsolation. URL: <https://www.electronjs.org/docs/latest/tutorial/security> (дата звернення: 10.04.2026)

53. W3Schools. JavaScript Tutorial - навчальні матеріали з мови програмування JavaScript для створення вебзастосунків. 2024. URL: <https://www.w3schools.com/Js/default.asp> (дата звернення: 11.04.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 82
Зм.	Арк.	№ докум.	Підпис	Дата		

54. Використання бібліотеки systeminformation для отримання HW/OS даних у Node.js. URL: <https://github.com/sebhildebrandt/systeminformation> (дата звернення: 12.04.2026)

55. System Monitoring Project. System-monitoring - Node.js Library for Retrieving CPU, Memory and Temperature Metrics. 2024. URL: <https://system-monitoring.js.org/> (дата звернення: 13.04.2026)

56. W3Schools. JavaScript Graphics - використання JavaScript для створення графіки та візуалізації даних у веббраузері. 2024. URL: https://www.w3schools.com/Js/js_graphics.asp (дата звернення: 14.04.2026)

57. W3Schools. JavaScript Reference - довідник методів, об'єктів та властивостей мови JavaScript. 2024. URL: <https://www.w3schools.com/jsref/default.asp> (дата звернення: 15.04.2026)

58. UserBenchmark. UserBenchmark - сервіс для порівняння продуктивності та аналізу характеристик комп'ютерних компонентів. 2025. URL: <https://userbenchmark.org/uk/> (дата звернення: 16.04.2026)

59. freeCodeCamp. JavaScript Algorithms and Data Structures - навчальний курс з програмування мовою JavaScript. 2024. URL: <https://www.freecodecamp.org/learn/javascript-v9/> (дата звернення: 17.04.2026)

60. Філоненко К. С. Веб-застосунок для інтернет-магазину комп'ютерної техніки з використанням React та Node.js : індивідуальний проєкт / КПП ім. Ігоря Сікорського, 2022. URL: <https://ela.kpi.ua/handle/123456789/57797> (дата звернення: 18.04.2026)

61. Марченко О. В., Шевченко О. О. Аналіз сучасних методів діагностики технічного стану персональних комп'ютерів. Харків : Харківський національний університет радіоелектроніки. URL: <https://openarchive.nure.ua/entities/publication/990a0190-80a3-4e22-bf8e-343c9b85341e/full> (дата звернення: 19.04.2026)

62. Методичні вказівки до виконання лабораторних робіт з дисципліни «Охорона праці». Луцьк : Волинський національний університет імені Лесі

					КвРКІ 022082.22.02.67 ПЗ	Арк. 83
Зм.	Арк.	№ докум.	Підпис	Дата		

Українки. URL: https://evnuir.vnu.edu.ua/bitstream/123456789/21253/1/labor_prakt.pdf (дата звернення: 20.04.2026)

63. Mustaffa M. Md., Shibghatullah A. S., Hasan Basari A. S., Hussin B. Diagnosing computer hardware failures using expert system (rule-based technique). New York : Foundation of Computer Science, International Journal of Computer Applications, 2021. URL: <https://ijcaonline.net/archives/volume183/issue11/31969-2021921411/> (дата звернення: 21.04.2026)

64. Al-Ajlan A., Zedan H. Intelligent Computer Hardware Fault Diagnosis System. Amman : The International Arab Journal of Information Technology, 2017. URL: <https://iajit.org/PDF/vol%2014,%20no.%205%20sep/8408.pdf> (дата звернення: 22.04.2026)

65. Groq API Cookbook. Practical examples of using Groq API. URL: <https://deepwiki.com/groq/groq-api-cookbook> (дата звернення: 23.04.2026)

66. GroqFlow. Framework for building AI workflows using Groq. URL: <https://github.com/groq/groqflow> (дата звернення: 24.04.2026)

67. Groq: Intuitively and Exhaustively Explained. Towards Data Science. URL: <https://towardsdatascience.com/groq-intuitively-and-exhaustively-explained-01e3fcd727ab/> (дата звернення: 25.04.2026)

68. Groq API Reference. Documentation of API methods for interacting with Groq models. URL: <https://console.groq.com/docs/api-reference#chat-create> (дата звернення: 26.04.2026)

69. Electron. Cross-platform desktop applications with JavaScript, HTML and CSS. URL: <https://www.electronjs.org/> (дата звернення: 27.04.2026)

70. Electron HTML CSS JS Tutorial. Tutorialpedia. URL: <https://www.tutorialpedia.org/blog/electron-html-css-js/> (дата звернення: 28.04.2026)

71. Chau D. Desktop Application Development Using Electron Framework. Finland : Theseus, 2021. URL: https://www.theseus.fi/bitstream/handle/10024/353557/Chau_Dao.pdf (дата звернення: 29.04.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 84
Зм.	Арк.	№ докум.	Підпис	Дата		

72. Influence of JavaScript on Website Development and User Experience. International Journal for Research in Applied Science and Engineering Technology. URL: <https://www.ijraset.com/research-paper/influence-of-javascript-on-website-development-and-user-experience> (дата звернення: 30.04.2026)

73. Park T. PubCSS: Formatting Academic Publications in HTML and CSS. URL: <https://thomaspark.co/2015/01/pubcss-formatting-academic-publications-in-html-css> (дата звернення: 05.05.2026)

74. Coyier C. Designing for Long-Form Articles. CSS-Tricks. URL: <https://css-tricks.com/designing-for-long-form-articles/> (дата звернення: 07.05.2026)

75. HTML5: A New Standard for the Web. URL: https://www.academia.edu/24118014/HTML5_A_New_Standard_for_the_Web (дата звернення: 08.05.2026)

76. Kumar R., Sharma P. Web Application Development Using HTML, CSS and JavaScript. Journal of Novel Research in Information and Data Science. URL: <https://tjjer.org/jnrid/papers/JNRID2308001.pdf> (дата звернення: 09.05.2026)

77. HTML in Visual Studio Code. Documentation for working with HTML in Visual Studio Code. URL: <https://code.visualstudio.com/Docs/languages/html> (дата звернення: 09.05.2026)

78. How to Analyze Computer Systems: A Comprehensive Guide. Binadox. URL: <https://www.binadox.com/blog/how-to-analyze-computer-systems-a-comprehensive-guide/> (дата звернення: 10.05.2026)

79. PCPartPicker. System builder and compatibility checker for computer components. URL: <https://pcpartpicker.com/> (дата звернення: 11.05.2026)

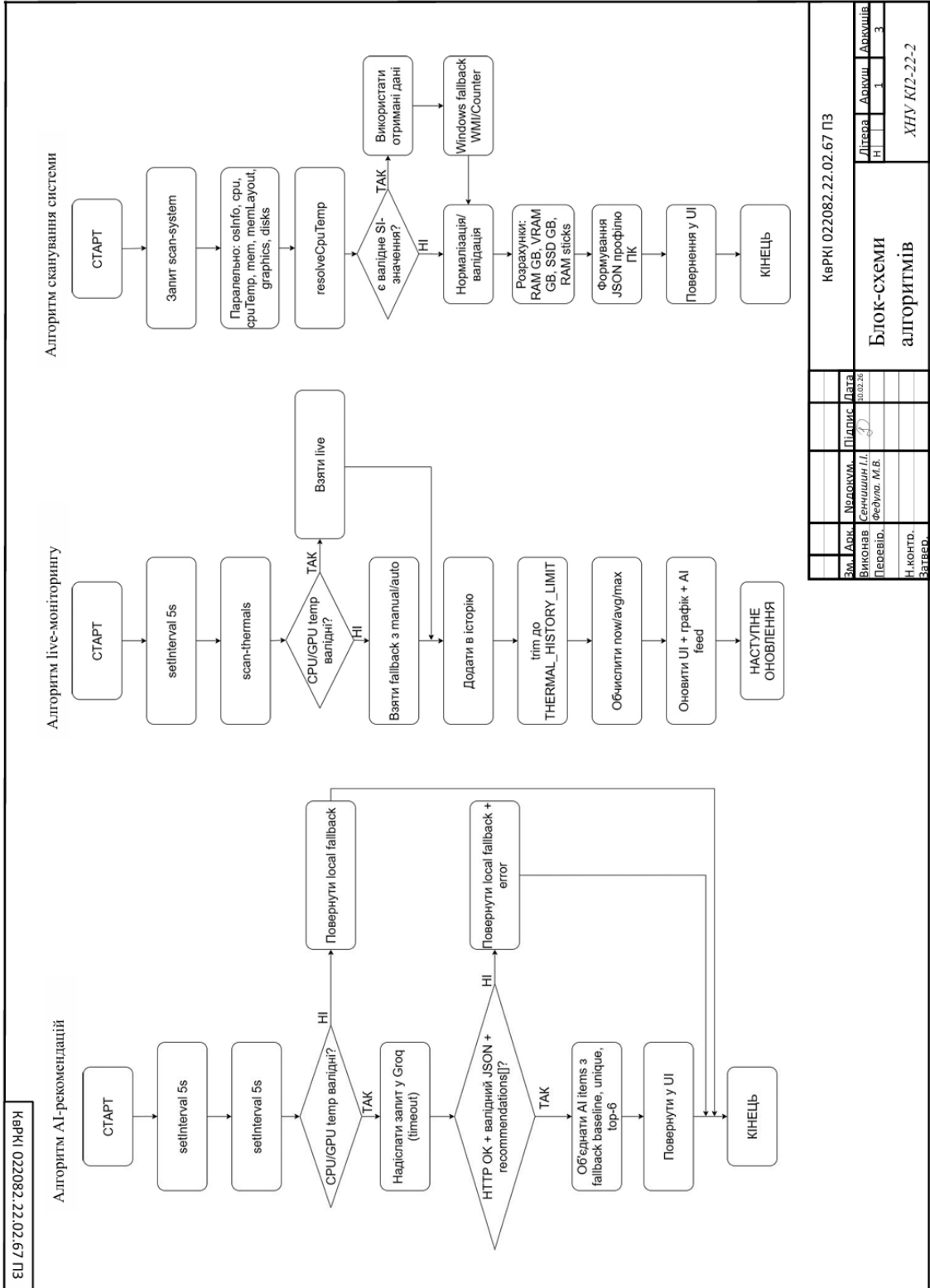
80. BuildMyPC. Online platform for building and comparing PC configurations. URL: <https://buildmypc.net/> (дата звернення: 12.05.2026)

					КВРКІ 022082.22.02.67 ПЗ	Арк. 85
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А

(обов'язковий)

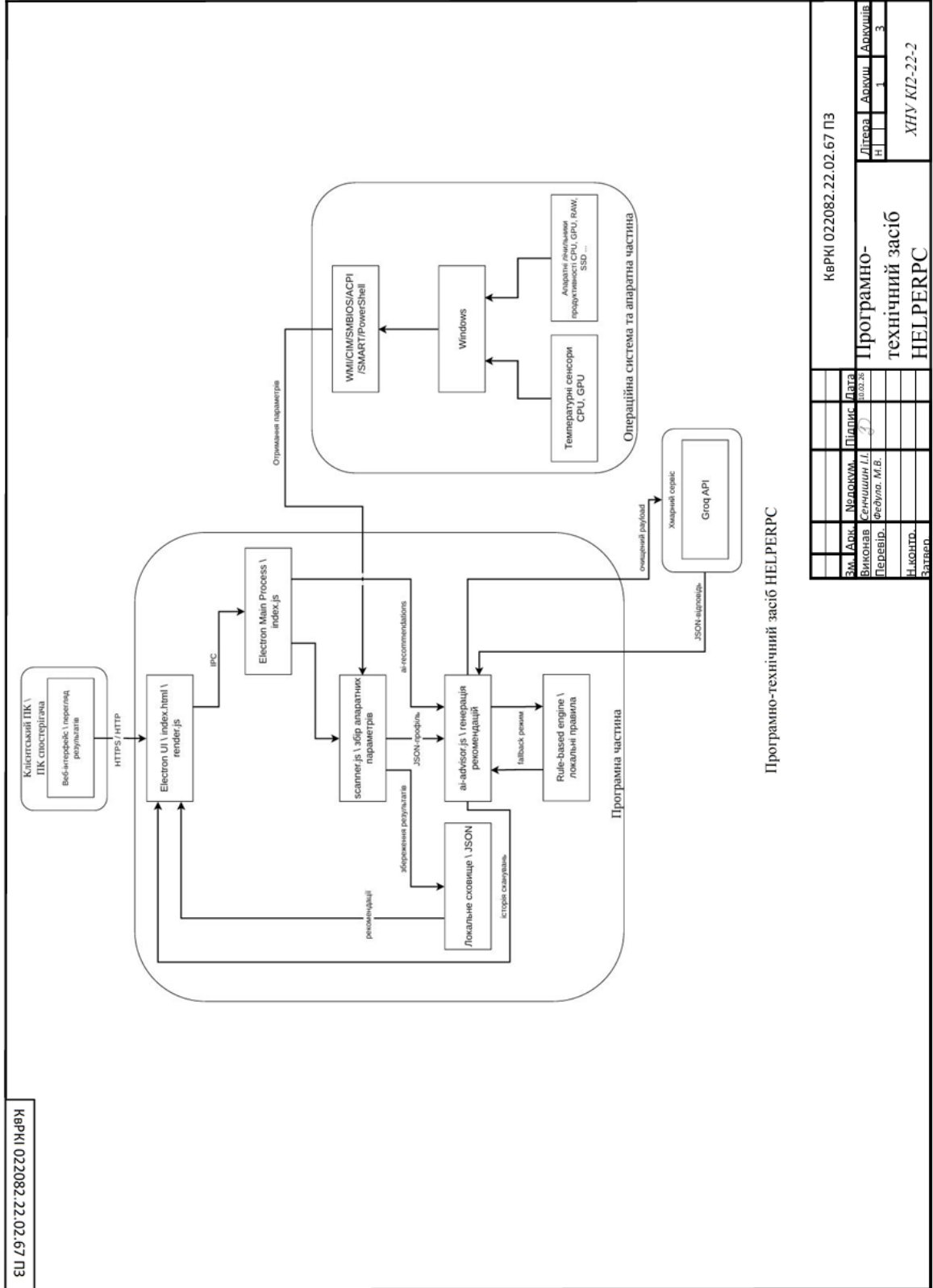
Тестування застосунку на різних ПК



ДОДАТОК В

(обов'язковий)

Схема програмно-технічного застосунку "HELPERPC"



Програмно-технічний засіб HELPERPC

КерПКІ 022082.22.02.67 ПЗ			
Зм. Друк	Недоліки	Підпис	Дата
Виконав	Семчишин І.І.	30.02.20	
Підтверд.	Федюла М.Б.		
Н.Стор.			
Б.Тверд.			
Програмно-технічний засіб HELPERPC		Літера	Аркуш
		Н	1
		А	3
ХМУ КІ2-22-2			

ДОДАТОК Г

(обов'язковий)

Код підключення ШІ до проекту і вид надання відповідей

Ai-advisor.js:

```
const DEFAULT_MODEL = process.env.GROQ_MODEL || "llama-3.1-8b-instant";
const GROQ_URL = "https://api.groq.com/openai/v1/chat/completions";
const GROQ_TIMEOUT_MS = Number.parseInt(process.env.GROQ_TIMEOUT_MS || "6000", 10);
function asNumber(value) {
  const n = Number.parseFloat(String(value ?? "").replace(",", "."));
  return Number.isFinite(n) ? n : null;
}
function pushIf(items, condition, text) {
  if (condition) items.push(text);
}
function looksLikeLaptop(value) {
  return /laptop|notebook/i.test(String(value || ""));
}
function cpuExampleByVendor(cpuName) {
  const cpu = String(cpuName || "");
  if (/amd|ryzen/i.test(cpu)) {
    return "CPU приклади (перевір socket/BIOS): Ryzen 5 5600, Ryzen 7 5700X, Ryzen 7 5800X3D.";
  }
  if (/intel|core/i.test(cpu)) {
    return "CPU приклади (перевір socket/BIOS): Core i5-12400F, Core i5-13400F, Core i7-12700F.";
  }
  return "CPU приклади: Core i5-12400F або Ryzen 5 5600 (після перевірки сумісності плати).";
}
function buildFallbackRecommendations(input) {
  const data = input || {};
  const cpuTemp = asNumber(data.cpuTemp);
  const gpuTemp = asNumber(data.gpuTemp);
  const ram = asNumber(data.ram);
  const ramSlots = asNumber(data.ramSlots);
  const ssd = String(data.ssd || "").toLowerCase();
  const isLaptop = looksLikeLaptop(data.device);
  const items = [];
  pushIf(
    items,
    cpuTemp !== null && cpuTemp >= 90,
    "CPU температура критична. Спочатку обслуговування охолодження: чистка, нова термопаста, перевірка тиску кулера."
  );
  pushIf(
```

```

    items,
    gpuTemp !== null && gpuTemp >= 88,
    "GPU температура висока. Перевір вентилятори, продуй радіатор, за
потреби зроби repaste/repad відеокарти."
);
pushIf(
    items,
    cpuTemp !== null && cpuTemp >= 80 && cpuTemp < 90,
    "CPU температура підвищена. Оптимізуй криву вентиляторів і airflow
корпусу перед апгрейдом компонентів."
);
pushIf(
    items,
    ram !== null && ram < 16,
    "ОЗП нижче 16 GB. Приклади: 2x8 GB DDR4-3200 або 2x16 GB DDR4-
3200/3600 (перевір тип DDR і частоти плати)."
);
pushIf(
    items,
    ram !== null && ram >= 16 && ramSlots === 1,
    "Рекомендовано двоканал: краще 2 однакові планки RAM замість однієї
для вищої пропускної здатності."
);
pushIf(
    items,
    ssd && !/nvme|m\.2/.test(ssd),
    "Схоже, що диск не NVMe. Приклади SSD: WD Black SN770, Samsung
980/990 EVO, Kingston NV2 (перевір M.2 слот)."
);

if (!isLaptop) {
    pushIf(
        items,
        ram !== null && ram >= 16 && (cpuTemp === null || cpuTemp < 85),
        cpuExampleByVendor(data.cpu)
    );
    pushIf(
        items,
        true,
        "GPU приклади (перевір БЖ і розміри): RTX 4060/4060 Ti, RX
7600/7700 XT."
    );
} else {
    items.push("Для ноутбука найчастіше практичні апгрейди: RAM і SSD;
CPU/GPU зазвичай не змінюються окремо.");
}
if (items.length === 0) {
    items.push("Система виглядає стабільно: критичних причин для
негайної заміни компонентів не виявлено.");
    items.push("Можна відкласти апгрейд, залишити профілактичну чистку
та періодичний контроль температур.");
}

```

```

    const baseline = [
      "Перевір сумісність перед купівлею: socket CPU, тип RAM (DDR4/DDR5), наявність M.2, потужність БЖ.",
      "Якщо GPU/CPU гріються, спочатку обслуговування охолодження, потім апгрейд компонентів.",
      "Для ігор у 1080p пріоритет зазвичай: GPU > CPU > RAM > SSD.",
      "Для роботи/монтажу пріоритет частіше: RAM та SSD, потім CPU.",
      "Онови BIOS/драйвери після апгрейду для стабільності та коректної роботи пам'яті.",
      "Якщо бюджет обмежений, почни з найвужчого місця: RAM <16 GB, старий SATA SSD або гарячий кулер."
    ];
    const normalizedItems = ensureSix(items, baseline);
    return {
      source: "local-fallback",
      summary: "Рекомендації згенеровано локально (без Groq API).",
      items: normalizedItems
    };
  }
}
function compactInput(payload) {
  const merged = payload?.merged || {};
  const thermal = payload?.thermal || {};
  return {
    os: merged.os || "Unknown",
    cpu: merged.cpu || "Unknown",
    gpu: merged.gpu || "Unknown",
    ram: merged.ram ?? "-",
    ramSlots: merged.ramSlots ?? "-",
    ssd: merged.ssd || "-",
    cpuTemp: thermal.cpuNow ?? merged.cpuTemp ?? null,
    gpuTemp: thermal.gpuNow ?? merged.gpuTemp ?? null,
    cpuAvg: thermal.cpuAvg ?? null,
    gpuAvg: thermal.gpuAvg ?? null,
    cpuMax: thermal.cpuMax ?? null,
    gpuMax: thermal.gpuMax ?? null
  };
}
function safeJsonParse(raw) {
  if (!raw) return null;
  try {
    return JSON.parse(raw);
  } catch {
    return null;
  }
}
function uniqueNonEmpty(items) {
  const out = [];
  const seen = new Set();
  for (const item of items || []) {
    if (typeof item !== "string") continue;
    const text = item.trim();
    if (!text) continue;
  }
}

```

```

    const key = text.toLowerCase();
    if (seen.has(key)) continue;
    seen.add(key);
    out.push(text);
  }
  return out;
}
function ensureSix(primaryItems, fallbackItems) {
  const merged = uniqueNonEmpty([...(primaryItems || []), ...(fallbackItems || [])]);
  return merged.slice(0, 6);
}
async function askGroq(compact) {
  const key = process.env.GROQ_API_KEY;
  if (!key) return null;
  const systemPrompt = [
    "Ти PC hardware advisor.",
    "Дай короткі практичні рекомендації українською.",
    "Працюй лише з наданими метриками.",
    "Формат відповіді: JSON об'єкт з полями summary (string) і recommendations (string[]).",
    "Наводь конкретні приклади компонентів (CPU/GPU/RAM/SSD) як варіанти апгрейду.",
    "Поверни рівно 6 рекомендацій у полі recommendations.",
    "Кожна рекомендація має бути практичною і короткою; додавай примітку перевірити сумісність (socket/PSU/DDR/M.2).",
    "Для ноутбуків фокусуйся на RAM/SSD і не радь заміну десктопних CPU/GPU."
  ].join(" ");
  const userPrompt = `Scan data:\n${JSON.stringify(compact, null, 2)}`;
  const controller = new AbortController();
  const timeoutId = setTimeout(() => controller.abort(), GROQ_TIMEOUT_MS);
  let res;
  try {
    res = await fetch(GROQ_URL, {
      method: "POST",
      headers: {
        Authorization: `Bearer ${key}`,
        "Content-Type": "application/json"
      },
      body: JSON.stringify({
        model: DEFAULT_MODEL,
        temperature: 0.2,
        max_tokens: 450,
        response_format: { type: "json_object" },
        messages: [
          { role: "system", content: systemPrompt },
          { role: "user", content: userPrompt }
        ]
      })
    ),

```

```

    signal: controller.signal
  });
} finally {
  clearTimeout(timeoutId);
}
if (!res.ok) {
  const text = await res.text();
  throw new Error(`Groq API error ${res.status}: ${text.slice(0,
300)}`);
}
const json = await res.json();
const content = json?.choices?.[0]?.message?.content;
const parsed = safeJsonParse(content);
const recs = Array.isArray(parsed?.recommendations)
  ? parsed.recommendations.filter((item) => typeof item === "string"
&& item.trim())
  : [];
if (recs.length === 0) {
  throw new Error("Groq response does not contain recommendations
array.");
}
return {
  source: "groq",
  model: DEFAULT_MODEL,
  summary: typeof parsed?.summary === "string" ? parsed.summary :
"AI рекомендації на базі поточного скану.",
  items: recs.slice(0, 6)
};
}
async function getRecommendations(payload) {
  const compact = compactInput(payload);
  const fallback = buildFallbackRecommendations(compact);
  try {
    const ai = await askGroq(compact);
    if (ai) {
      return {
        ...ai,
        items: ensureSix(ai.items, fallback.items)
      };
    }
    return fallback;
  } catch (error) {
    return {
      ...fallback,
      error: error?.message || "Unknown error"
    };
  }
}
}

module.exports = { getRecommendations };

```

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 10%**

ID: 272413 Назва: БКР Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера Додано в БД: 2026-05-27 Автора: Ігор СЕНЧИШИН Керівники: Микола ФЕДУЛА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	114131	837	3189 (3%)	39 (5%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Ігор СЕНЧИШИН

Співавтор:

Назва: Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера

Експерт: Микола ФЕДУЛА

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 3.35%

Коефіцієнт подібності 2: 0%

Мікропробіли: 3

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-27 06:32:32.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-05-27

Дата



Доцент Андрій Нічепорук

експерт

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Сенчишин Ігор Ігорович

Тема: Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 94

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка веб-системи для аналізу технічних параметрів та формування конфігурації персонального комп'ютера.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено дослідження предметної області, пов'язаної з аналізом технічних параметрів персонального комп'ютера та системи моніторингу апаратних ресурсів. Розглянуто основні підходи до збору, обробки та відображення даних про стан комп'ютерної системи, зокрема показники завантаження процесора, використання оперативної пам'яті, температурні параметри компонентів, а також стан накопичувачів. Виконано аналіз існуючих програмних засобів моніторингу та діагностики комп'ютерних систем. На основі проведеного аналізу сформульовано постановку задачі розроблення веб-систем, призначених для аналізу технічних характеристик комп'ютера та формування оптимальної конфігурації апаратних компонентів.

У другому розділі кваліфікаційної роботи здійснено проектування програмної архітектури веб-систем. Розглянуто принципи побудови і використання сучасних веб-технологій, а також засобів розробки на базі JavaScript. Виконано формалізований опис структури системи, визначено основні функціональні модулі

та їх взаємодію. Розроблено алгоритми збору технічних параметрів комп'ютера, їх обробки та подальшого аналізу. Також створено структуру бази даних та логіку обробки інформації для формування рекомендації щодо конфігурацій персонального комп'ютера. Описано взаємодію між компонентами системи, а також принципи відображення результатів аналізу користувачу.

У третьому розділі кваліфікаційної роботи виконано програмну реалізацію розробленої веб-системи. Реалізовано програмні модулі збору інформації про апаратні ресурси комп'ютера з використанням сучасних бібліотек та засобів. Також розроблено користувацький інтерфейс системи з використанням веб-технологій та середовища Electron для створення настільного застосунку. Реалізовано механізми аналізу отриманих параметрів, відображення температурних показників процесора, використання оперативної пам'яті, стану і розміру накопичувачів та інших характеристик системи. Проведено тестування програмного засобу та перевірку коректності роботи під час збору та аналізу технічних параметрів комп'ютера.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага щодо виконання додатку.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

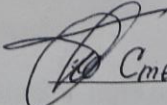
8. Інші зауваження: _____

9. Оцінка дипломної роботи: відмінно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

*к.т.н., доцент кафедри фармакологічної та медичної
ма інтелектуальних технологій*

“ 01 ” *08* 2026 р.

 Стецюк В. І. (підпис)

Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Сенчишин Ігор Ігорович

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-2

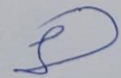
ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Сенчишин Ігор Ігорович на захист кваліфікаційної роботи
(прізвище, ім'я, по батькові)

за спеціальністю 123 - Комп'ютерна інженерія

На тему: Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера

Кваліфікаційна робота, рецензія і довідка про перевірку на академічні запозичення додаються.

В.О. Декан факультету



Сергій Меселеко
(ім'я, прізвище)

ДОВІДКА УСПІШНОСТІ

Сенчишин І.І. з 2022 по 2025 роки повністю виконав навчальний план спеціальності з таким розподілом оцінок за:

національною шкалою: відмінно 55,56 %, добре 37,04 %, задовільно 7,41 %.

шкалою ЄКТС: А 55,56 %, В 14,81 %, С 22,22 %, D 3,70 %, E 3,70 %.

Методист факультету

Метод
(підпис)

Метод Косенко
(ім'я, прізвище)

ВИСНОВОК КЕРІВНИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Сенчишин Ігор Ігорович виконав бакалаврську кваліфікаційну роботу на високому науково-технічному рівні. Розроблено програмно-технічний засіб для моніторингу та конфігурації ресурсів комп'ютера із застосуванням сучасних технологій, в м.ч. елементів штучного інтелекту. Результати реалізовані у формі діючого програмно-техн. засобу на інтерн.

Оцінка кваліфікаційної роботи відмінно А (95,00)

Керівник кваліфікаційної роботи

Ф.И.
(підпис)

Микола Федун
(ім'я, прізвище)

" 01 " 06 2026 р.

ВИСНОВОК КАФЕДРИ ПРО КВАЛІФІКАЦІЙНУ РОБОТУ

Кваліфікаційну роботу розглянуто. Студент Сенчишин І.І. допускається до її захисту на екзаменаційній комісії.

Завідувач кафедри

О.Павлов
(підпис, ім'я, прізвище)

КІС
(назва)

" 01 " 06 2026 р.

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Програмно-технічний засіб для віддаленого моніторингу та конфігурації ресурсів комп'ютера

Автор Ігор СЕНЧИШИН

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Микола ФЕДУЛА

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:


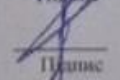
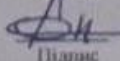
- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
 - 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
 - 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
 - 4) значна частина знайденого плагіату відноситься до списку використаних джерел
- Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3,35% і адресується до 20 першоджерел; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.05.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис

Підпис

Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Микола ФЕДУЛА
Ім'я, ПРІЗВИЩЕ