

Полтавський університет економіки і торгівлі
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

(підпис)

<<__>> _____ 202__р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ
<<РОЗРОБКА САЙТУ “ ПОРІВНЯННЯ УТИЛІТ ДЛЯ МОНІТОРИНГУ
СТАНУ АПАРАТНИХ ЗАСОБІВ ПК ”>>**

Зі спеціальності 122 <<Комп'ютерні науки>>

Освітня програма <<Комп'ютерні науки>>

Ступеня бакалавр

Виконавець роботи Караблін Роман Вячеславович

_____ <<__>> _____ 202__р.

(підпис)

Науковий керівник к.ф.-м.н., Олексійчук Юрій Федорович

_____ <<__>> _____ 202__р.

(підпис)

Рецензент

ПОЛТАВА 2025

ЗАВДАННЯ І ПЛАН НА КВАЛІФІКАЦІЙНУ РОБОТУ

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Олена Ольховська

<<__>> вересня 2025р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

**на тему << Розробка сайту “ Порівняння утиліт для моніторингу стану
апаратних засобів ПК”>>**

зі спеціальності 122 «Комп’ютерні науки»

освітня програма «Комп’ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Караблін Роман Вячеславович

Затверджена наказом ректора № _____-Н від « » _____ 2025 р.

Термін подання студентом роботи «__» _____ 2025 р.

Вихідні дані до кваліфікаційної роботи:

Зміст пояснювальної записки

ВСТУП

1. ПОСТАНОВКА ЗАДАЧИ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Критерії оцінювання програмних засобів моніторингу

2.2 Огляд популярних утиліт для моніторингу стану ПК

2.3 Порівняльна таблиця характеристик та можливостей утиліт

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття моніторингу апаратного забезпечення

3.2 Основні параметри для моніторингу

3.3 Роль моніторингу у забезпеченні надійної роботи ПК

4. ПРАКТИЧНА ЧАСТИНА

4.1 Вибір платформи та технологій розробки сайту

4.2 Структура та функціонал сайту

4.3 Дизайн сайту та адаптивність

4.4 Реалізація: опис етапів розробки

4.5 Результати тестування сайту

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А. Код програми

Перелік графічного матеріалу: 3-4 аркуші графічного матеріалу, інші необхідні ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	ППП, посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Постанова задачі			
Інформаційний огляд			
Теоретична частина			
Практична частина			

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання <<__>> _____ 202_ р.

Здобувач вищої освіти Караблін Роман Вячеславович

Науковий керівник к.ф.-м.н., Олексійчук Юрій Федорович

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____

(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № ____ віл <<__>> _____ 202_ р.

Секретар ЕК _____
(підпис) (прізвище та ініціали)

Затверджую

Зав. Кафедрою _____

к.ф.-м.н. Олена ОЛЬХОВСЬКА

<<__>> _____ 202_ р.

Погоджено

Науковим керівником _____

<<__>> _____ 202_ р.

План

кваліфікаційної роботи на тему

<< Розробка сайту “ Порівняння утиліт для моніторингу стану апаратних засобів ПК”>>

зі спеціальності 122 Комп’ютерні науки

освітня програма 122 Комп’ютерні науки

ступеня бакалавр

Прізвище, ім'я, по батькові Караблін Роман Вячеславович

ВСТУП

1. ПОСТАНОВКА ЗАДАЧИ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Критерії оцінювання програмних засобів моніторингу

2.2 Огляд популярних утиліт для моніторингу стану ПК

2.3 Порівняльна таблиця характеристик та можливостей утиліт

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття моніторингу апаратного забезпечення

3.2 Основні параметри для моніторингу

3.3 Роль моніторингу у забезпеченні надійної роботи ПК

4. ПРАКТИЧНА ЧАСТИНА

4.1 Вибір платформи та технологій розробки сайту

4.2 Структура та функціонал сайту

4.3 Дизайн сайту та адаптивність

4.4 Реалізація: опис етапів розробки

4.5 Результати тестування сайту

ВИСНОВОК

Здобувач вищої освіти Караблін Роман Вячеславович

<<__>> _____ 202_ р.

РЕФЕРАТ

Записка:с.60,рис.13,табл.2,джерел 17

МОНІТОРИНГ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ, ВЕБРОЗРОБКА, САЙТ-ПОРІВНЯННЯ УТИЛІТ

Об'єкт розробки — інформаційний вебресурс для користувачів персональних комп'ютерів.

Мета роботи — розробка вебсайту для порівняння утиліт моніторингу стану апаратних засобів ПК та надання користувачам структурованої інформації щодо функціональних можливостей різних програм.

Методи дослідження — аналіз і порівняння утиліт моніторингу, сучасні вебтехнології (HTML, CSS, JavaScript), адаптивна верстка на основі flex box та grid, принципи UI/UX-дизайну.

Зроблено аналіз основних параметрів апаратного моніторингу, таких як температура компонентів, навантаження CPU/GPU, обсяг оперативної пам'яті, стан накопичувачів, швидкість обертання вентиляторів кулера тощо. Досліджено функціональні можливості популярних утиліт (HW Monitor, AIDA64, HWiNFO, GPU-Z та інші) та виявлено їхні переваги й недоліки.

Розроблено структуру, інтерфейс і функціонал вебсайту, що включає інформаційні сторінки, карточки утиліт, пошук, порівняльну таблицю. Реалізовано адаптивний дизайн для коректного відображення на ПК, планшетах і мобільних пристроях.

Здійснена програмна реалізація сайту засобами HTML/CSS/JavaScript з подальшим тестуванням на різних роздільних здатностях екрана.

В результаті тестування виявлено, що сайт функціонує стабільно, має інтуїтивно зрозумілий інтерфейс, забезпечує швидке завантаження сторінок і повну відповідність технічному завданню.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК, СИМВОЛІВ, СКОРОЧЕНТ І ТЕРМІНІВ.....	9
ВСТУП.....	10
1. ПОСТАНОВКА ЗАДАЧИ.....	12
2.ІНФОРМАЦІЙНИЙ ОГЛЯД.....	13
2.1 Критерії оцінювання програмних засобів моніторингу.....	13
2.2 Огляд популярних утиліт для моніторингу стану ПК.....	14
2.3 Порівняльна таблиця характеристик та можливостей утиліт.....	18
3. ТЕОРЕТИЧНА ЧАСТИНА.....	20
3.1 Поняття моніторингу апаратного забезпечення.....	20
3.2 Основні параметри для моніторингу.....	21
3.3 Роль моніторингу у забезпеченні надійної роботи ПК.....	24
4. ПРАКТИЧНА ЧАСТИНА.....	28
4.1 Вибір платформи та технологій розробки сайту.....	28
4.2 Структура та функціонал сайту.....	30
4.3 Дизайн сайту та адаптивність.....	35
4.4 Реалізація: опис етапів розробки.....	37
4.5 Результати тестування сайту.....	39
ВИСНОВОК.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
ДОДАТОК А. Код програми.....	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧОК, СИМВОЛІВ, СКОРОЧЕНТ І ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних пояснень, скорочень, символів
ПК	Персональний комп'ютер
ОС	Операційна система
ПЗ	Програмне забезпечення
ЦП	Центральний процесор
НЖМД	Накопичувач на жорсткому магнітному диску
АЦП	Аналого-цифровий перетворювач

ВСТУП

Сучасні комп'ютерні системи відіграють важливу роль у житті користувачів, забезпечуючи виконання широкого спектра завдань – від повсякденного використання до високопродуктивних обчислень. Однак ефективність та стабільність роботи комп'ютерів безпосередньо залежать від стану їхніх апаратних компонентів, таких як процесор, оперативна пам'ять, накопичувачі та система охолодження.

Для контролю та аналізу стану апаратного забезпечення розроблено низку спеціалізованих утиліт, які дозволяють здійснювати моніторинг температури, навантаження, напруги, швидкості обертання вентиляторів тощо. Вибір оптимального програмного інструменту для таких завдань є актуальним питанням, оскільки різні утиліти мають свої переваги та недоліки.

Мета даної кваліфікаційної роботи – розробка веб-сайту, який забезпечуватиме зручне порівняння популярних утиліт для моніторингу апаратних засобів ПК. Даний ресурс допоможе користувачам обирати найбільш підходящі рішення відповідно до їхніх потреб, порівнюючи функціональні можливості, зручність використання, сумісність із операційними системами та інші характеристики.

Об'єкт дослідження – програмні засоби для моніторингу стану апаратного забезпечення персонального комп'ютера.

Предмет дослідження – процес створення веб-ресурсу як засобу інформаційного представлення, аналізу та порівняння утиліт моніторингу ПК.

Методи дослідження – аналіз технічної документації утиліт, метод порівняння, розробка інтерфейсу за допомогою HTML, CSS, JavaScript, тестування функціоналу вебсайту на різних пристроях.

Завдання роботи включають:

- аналіз існуючих утиліт для моніторингу апаратного забезпечення ПК;
- визначення ключових параметрів для порівняння;
- розробку веб-сайту з інтерактивною системою оцінки та вибору утиліт;
- тестування та оцінку ефективності створеного ресурсу.

Основна частина роботи присвячена огляду аналогів, аналізу технологій веброзробки, розробці структури та функціоналу сайту, опису етапів створення, а також практичному тестуванню ресурсу. У результаті реалізовано сайт із зручною навігацією, адаптивним дизайном і функціями, що дозволяють користувачу легко порівняти програмне забезпечення для моніторингу ПК.

Розробка такого веб-сайту сприятиме підвищенню обізнаності користувачів щодо доступних інструментів моніторингу та допоможе їм приймати обґрунтовані рішення при виборі утиліт для підтримки стабільної роботи ПК.

1. ПОСТАНОВКА ЗАДАЧИ

У сучасних комп'ютерних системах важливо постійно контролювати стан апаратного забезпечення для забезпечення стабільної роботи, запобігання перегріву та збоїв, а також для своєчасного виявлення проблем. На ринку існує велика кількість програмних засобів (утиліт), призначених для моніторингу різних параметрів комп'ютера — таких як температура процесора, навантаження на оперативну пам'ять, стан накопичувачів тощо. Однак вибір оптимального інструменту залежить від потреб користувача, сумісності з обладнанням, функціональності утиліти та зручності її використання.

У рамках кваліфікаційної роботи поставлені такі задачі. Провести огляд та аналіз найпопулярніших утиліт для моніторингу апаратних ресурсів персонального комп'ютера, порівняти їх за ключовими критеріями, такими як точність вимірювань, які підтримує компоненти, інтерфейс, системні вимоги та вартість; створити інформаційний веб-сайт, який дозволить користувачам зручно ознайомитися з результатами порівняння, переглянути характеристики утиліт, а також отримати рекомендації щодо їх використання.

Розроблений сайт повинен мати сучасний налаштований інтерфейс, бути зручним у навігації по сайту та дозволяти ефективно представляти результати дослідження у вигляді таблиць, графіків, описів та візуалізацій.

Таким чином, результатом роботи має стати як аналітичне порівняння утиліт, так і вебресурс, який забезпечить користувачам легкий доступ до цієї інформації.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Критерії оцінювання програмних засобів моніторингу

Для об'єктивного порівняння програмних засобів моніторингу апаратного забезпечення комп'ютера необхідно визначити критерії, за якими можна здійснювати оцінювання утиліт. Такий підхід дозволяє сформувати системне уявлення про можливості кожної утиліти, виявити її сильні та слабкі сторони, а також зробити обґрунтований вибір залежно від конкретних потреб користувача.

У рамках цієї роботи були обрані основні критерії оцінювання:

Функціональність. Оцінюється обсяг параметрів, які може відображати утиліта: температура процесора, навантаження на ядра, стан жорстких дисків, напруга, швидкість обертання вентиляторів тощо. Чим більше компонентів підтримується — тим ширші можливості аналізу.

Інтерфейс користувача. Враховується зручність і логічність інтерфейсу, наявність графіків, інформативність, доступність налаштувань, мова інтерфейсу та підтримка темного/світлого режиму.

Системні вимоги та продуктивність. Деякі утиліти можуть споживати багато ресурсів і впливати на продуктивність системи. Важливо оцінити, наскільки програма навантажує процесор або оперативну пам'ять під час роботи.

Точність даних. Надзвичайно важливий критерій, особливо в умовах діагностики чи оверклокінгу. Оцінюється, наскільки дані, які надає утиліта, відповідають реальним значенням або збігаються з іншими перевіреними програмами.

Підтримка сучасного обладнання. Оцінюється, наскільки добре програма розпізнає нові процесори, чипсети, відеокарти, материнські плати тощо. Актуальність оновлень теж враховується.

Можливість експорту та ведення логів. Для глибшого аналізу корисною є функція збереження звітів, логів у вигляді файлів, графіків, скріншотів.

Ціна та ліцензія. Деякі утиліти є повністю безкоштовними, інші — умовно безкоштовні (freeware з обмеженим функціоналом), або ж потребують покупки повної версії. Враховується співвідношення ціни до можливостей.

Ці критерії були використані як основа для складання оглядових таблиць, аналізу кожної утиліти та формування загальних висновків.

2.2 Огляд популярних утиліт для моніторингу стану ПК

Існує велика кількість утиліт для моніторингу апаратного забезпечення персонального комп'ютера, які відрізняються за функціональністю, інтерфейсом, системними вимогами та призначенням. Нижче наведено короткий огляд найпопулярніших з них, які отримали позитивні відгуки як серед звичайних користувачів, так і в ІТ-спільноті.

1. Утиліта HWMonitor

Розробник: CPUID. Легка та швидка програма для перегляду температур, напруг і швидкостей обертання вентиляторів. Має простий текстовий інтерфейс без графіків. Безкоштовна, з доступною Pro-версією. Ідеально підходить для базового моніторингу. На рисунку 2.1 зображений логотип утиліти HWMonitor [1].

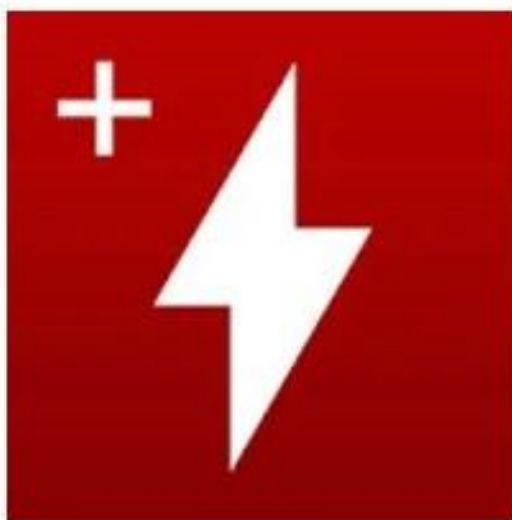


Рис. 2.1 – Логотип утиліти HWMonitor

2. Утиліта HWiNFO

Розробник: REALiX. Дуже розширена утиліта, що показує всю апаратну інформацію та підтримує моніторинг у реальному часі. Має графіки, звіти, виведення на панель або в системний трей. Підтримує експорт логів. Безкоштовна для некомерційного використання. На рисунку 2.2 зображений логотип утиліти HWiNFO [2].



Рис. 2.2 – Логотип утиліти HWiNFO

3. Утиліта AIDA64

Розробник: FinalWire. Платна програма з потужним функціоналом: детальна діагностика, тести продуктивності, підтримка зовнішніх дисплеїв. Підтримує більшість платформ: Windows, Android, iOS. Зручний інтерфейс підтримує звіти, графіки, сенсори. Призначена для професійного використання, але є пробна версія. На рисунку 2.3 зображений логотип утиліти AIDA64 [3].



Рис. 2.3 – Логотип утиліти AIDA64

4. Утиліта Open Hardware Monitor

Розробник: open-source спільнота. Безкоштовна, з відкритим кодом. Проста у використанні, підтримує базові функції моніторингу. Можна налаштувати виведення на робочий стіл. Оновлюється рідше, ніж інші утиліти. На рисунку 2.4 зображений логотип утиліти Open Hardware Monitor [4].



Рис. 2.4 – Логотип утиліти Open Hardware Monitor

5. Утиліта Spessy

Розробник: Piriform .Більш орієнтована на загальну інформацію про систему, ніж на глибокий моніторинг. Має приємний та зручний інтерфейс, працює швидко. Не відображає динамічні графіки. Підходить для швидкого огляду конфігурації. На рисунку 2.5 зображений логотип утиліти Spessy [5].



Рис. 2.5 – Логотип утиліти Spessy

2.3 Порівняльна таблиця характеристик та можливостей утиліт

Таблиця 2.1

Утиліта	Функціональність	Інтерфейс	Точність	Підтримка обладнання	Системні вимоги	Ціна	Додаткові можливості
HWMonitor	Середня	Примітивний	Висока	Добра	Дуже низькі	Безкоштовна / Pro	Мінімальний функціонал
HWiNFO	Висока	Складний	Дуже висока	Широка	Низькі	Безкоштовна	Експорт логів, інтеграція з RivaTuner
AIDA64	Дуже висока	Зручний	Висока	Дуже широка	Помірні	Платна (€ trial)	Тести, звіти, зовнішні дисплеї
Open HW Monitor	Середня	Простий	Середня	Обмежена	Низькі	Безкоштовна	Open-source, гнучкість
Spresu	Низька	Дуже зручний	Середня	Обмежена	Низькі	Безкоштовна / Pro	Загальна інфо, не моніторинг у реальному часі

Короткі висновки по таблиці 2.1:

- Утиліта HWiNFO — найповніший варіант для системного моніторингу, підходить як для досвідчених користувачів, так і для професіоналів.
- Утиліта AIDA64 — потужна, але не безкоштовна програма з професійним ухилом.
- Утиліта HWMonitor та Open Hardware Monitor — прості, легкі рішення для базових завдань.
- Утиліта Sressu — підходить для швидкої діагностики, але не для глибокого моніторингу.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття моніторингу апаратного забезпечення

Моніторинг апаратного забезпечення — це процес безперервного спостереження, збору, обробки та візуалізації даних про технічний стан компонентів комп'ютера у режимі реального часу. Такий контроль дає змогу користувачам та системним адміністраторам оперативно реагувати на будь-які критичні зміни у роботі пристрою, що дозволяє підтримувати стабільність і продуктивність системи на високому рівні.

Моніторинг виконується за допомогою спеціальних програм — утиліт, які взаємодіють з вбудованими сенсорами материнської плати, процесора, графічного адаптера, накопичувачів тощо. Вони здатні не лише виводити поточні показники, а й вести історію змін, будувати графіки та надсилати сповіщення про досягнення критичних значень.

До ключових параметрів, що підлягають моніторингу, належать:

- Температура основних компонентів системи (процесора, відеокарти, чипсета, SSD);
- Швидкість обертання вентиляторів, що впливають на охолодження;
- Напруга на різних лініях живлення, що критично важливо для стабільної роботи обладнання;
- Завантаження центрального (CPU) та графічного процесора (GPU);
- Споживання оперативної пам'яті (RAM);
- Стан жорстких дисків та SSD — температура, кількість увімкнень, залишковий ресурс (S.M.A.R.T. параметри);
- Продуктивність системи загалом — наприклад, через аналіз FPS у реальному часі або затримок обробки команд.

Моніторинг дозволяє:

- Вчасно виявляти та попереджати апаратні несправності;
- Зменшувати ризики перегріву або виходу з ладу окремих компонентів;

- Оптимізувати роботу системи, адаптуючи її до поточних навантажень;
- Підвищити надійність і загальну продуктивність ПК;
- Забезпечити безпечне середовище для оверклокінгу (розгону апаратного забезпечення);
- Продовжити термін служби компонентів шляхом підтримання робочих температур у межах норми;
- Надавати аналітику для прийняття рішень про оновлення або заміну частин системи.

Крім того, моніторинг може бути корисним і в корпоративному середовищі — наприклад, при управлінні парком комп'ютерів в офісі, де потрібна централізована система збору даних для виявлення відхилень у роботі обладнання.

3.2 Основні параметри для моніторингу

У процесі моніторингу апаратного забезпечення комп'ютера особливу увагу слід приділяти параметрам, які мають прямий вплив на стабільність, продуктивність, енергоефективність та безпеку функціонування комп'ютерної системи. Моніторинг цих показників дозволяє не лише виявити наявні проблеми, а й попередити потенційні збої або вихід з ладу окремих компонентів.

1. Температура компонентів

Температурний режим є ключовим параметром, що безпосередньо впливає на довговічність та стабільність роботи апаратного забезпечення.

Ціль моніторингу — попередити перегрів і уникнути тротлінгу (зниження частоти для охолодження).

Основні компоненти для контролю:

- CPU (центральний процесор)
- GPU (відеокарта)
- VRM (модуль регулятора напруги на материнській платі)

- SSD/NVMe накопичувачі

Орієнтовні межі:

- CPU: 30–75 °C (до 90 °C — критично)
- GPU: 30–85 °C (до 95 °C — критично)
- SSD: до 70 °C (бажано до 60 °C для NVMe)

Приклад: під час запуску гри процесор може досягати температури 80 °C. Якщо при цьому кулер не змінює оберти — слід перевірити його налаштування або справність.

2. Завантаження CPU та GPU

Цей параметр дозволяє оцінити навантаження на обчислювальні ресурси та зрозуміти, чи система працює в межах своїх можливостей.

CPU — обробка логіки, системних процесів, фонових ПЗ.

GPU — візуалізація, обробка 3D-графіки, відео.

Типові значення:

- Ідл режим: 1–10 %
- Робоче навантаження: 20–80 %
- 100 % — ознака перевантаження або поганої оптимізації.

Приклад: постійне 100% завантаження CPU під час простою може свідчити про шкідливе ПЗ.

3. Частоти процесора та графічного адаптера

Частота є характеристикою продуктивності: вища частота — більше операцій за одиницю часу.

Динамічне регулювання: сучасні CPU/GPU можуть змінювати частоту залежно від навантаження та температури.

Виявлення нестабільностей: зниження частот може свідчити про перегрів або нестачу живлення.

Приклад: у разі оверклокінгу частоти зростають, але підвищується і температура — це вимагає кращого охолодження.

4. Швидкість обертання вентиляторів

Швидкість обертання вентиляторів визначає ефективність охолодження і впливає на температуру всієї системи.

Одиниці виміру: RPM (обертів за хвилину).

Типи вентиляторів:

- CPU fan
- GPU fan
- System fan

Автоматичне регулювання: залежить від температурних датчиків.

Приклад: при зупиненому CPU fan температура процесора швидко досягає критичних значень.

5. Напруга живлення

Стабільна напруга — критично важливий фактор для стабільної роботи всіх компонентів ПК.

Контрольовані лінії:

- +3.3V, +5V, +12V — для основних компонентів
- Vcore — напруга на процесорі
- Відхилення понад 5–10 % — ознака несправності БЖ (блоку живлення) або електромережі.

Приклад: просідання напруги під навантаженням може викликати мимовільне вимикання системи.

6. Стан накопичувачів (HDD/SSD)

Технологія S.M.A.R.T. дозволяє виявити потенційні збої накопичувачів до фактичної втрати даних.

Параметри контролю:

- Кількість переназначених секторів
- Температура
- Кількість циклів вмикання/вимикання
- Час роботи

Програми:

- CrystalDiskInfo,
- Hard Disk Sentinel

Приклад: зростання числа нестабільних секторів — сигнал до негайного резервного копіювання.

7. Використання оперативної пам'яті (RAM)

Визначає обсяг пам'яті, зайнятої програмами, службами та системою.

Типові межі:

- Idle: 15–30%
- При звичайному навантаженні: 40–70%
- При нестачі пам'яті: спрацьовує свопінг (використання диска як тимчасової пам'яті)

Приклад: при 90–100 % завантаженій RAM комп'ютер починає “гальмувати”.

8. Мережеве навантаження

Деякі утиліти дозволяють моніторити обсяг вхідного/вихідного трафіку в реальному часі.

Корисно для:

- Виявлення надмірної активності фонових програм
- Визначення підозрілої активності (шпигунське ПЗ)
- Діагностики нестабільного інтернет-з'єднання

Приклад: утиліта показує високий вихідний трафік при неактивному браузері — можливе зараження трояном.

3.3 Роль моніторингу у забезпеченні надійної роботи ПК

Моніторинг апаратного забезпечення є невід'ємною частиною підтримки стабільної, безпечної та ефективної роботи комп'ютерної системи. Без постійного контролю параметрів комп'ютера, таких як температура

компонентів, завантаження процесора, графічного адаптера, оперативної пам'яті та накопичувачів, існує ризик виникнення серйозних проблем, які можуть спричинити поломки або втрату даних. Нижче наведено кілька основних аспектів, чому моніторинг є важливим для забезпечення надійної роботи ПК.

1. Запобігання перегріву та пошкодженням обладнання

Перегрів — одна з основних причин поломок в комп'ютерних компонентах, таких як процесор (CPU) і відеокарта (GPU). Висока температура може значно знизити ефективність роботи цих елементів і привести до їх виходу з ладу. Перегрів може бути викликаний низкою факторів: неправильним налаштуванням системи охолодження, накопиченням пилу або зношуванням термопасти. Моніторинг температури дозволяє вчасно виявити надмірне нагрівання і вжити заходів, таких як:

- Очищення вентиляційних отворів та охолоджувальних елементів.
- Заміна термопасти для покращення теплопередачі.
- Регулювання швидкості обертання вентиляторів для оптимізації охолодження.

2. Підвищення продуктивності

Завантаження процесора і графічного адаптера відображає, як активно використовуються ресурси ПК. Моніторинг цих показників дозволяє:

- Виявити програми або процеси, які споживають надмірну кількість ресурсів, що може знижувати продуктивність.
- Зрозуміти, чи потребує система апгрейду, наприклад, у вигляді додавання оперативної пам'яті або заміни компонентів на більш продуктивні.
- Оптимізувати налаштування системи та програмного забезпечення для більш ефективної роботи, особливо в умовах високих навантажень (наприклад, при роботі з великими обсягами даних або під час ігор).

3. Виявлення апаратних несправностей та збоїв

Моніторинг напруги живлення і стану накопичувачів дає змогу своєчасно виявити потенційні проблеми з компонентами:

- Напруга на лініях живлення (3.3V, 5V, 12V) може змінюватися через несправність блоку живлення або неправильне підключення, що може призвести до нестабільної роботи системи.
- За допомогою S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology) можна оцінити стан жорстких дисків і SSD, виявляючи проблеми, такі як зношення або наявність битих секторів. Це дає змогу вчасно перенести дані на резервні носії і уникнути втрати важливої інформації.

4. Забезпечення довговічності обладнання

Моніторинг допомагає виявляти і виправляти дрібні проблеми ще до того, як вони стануть серйозними, що продовжує термін служби компонентів.

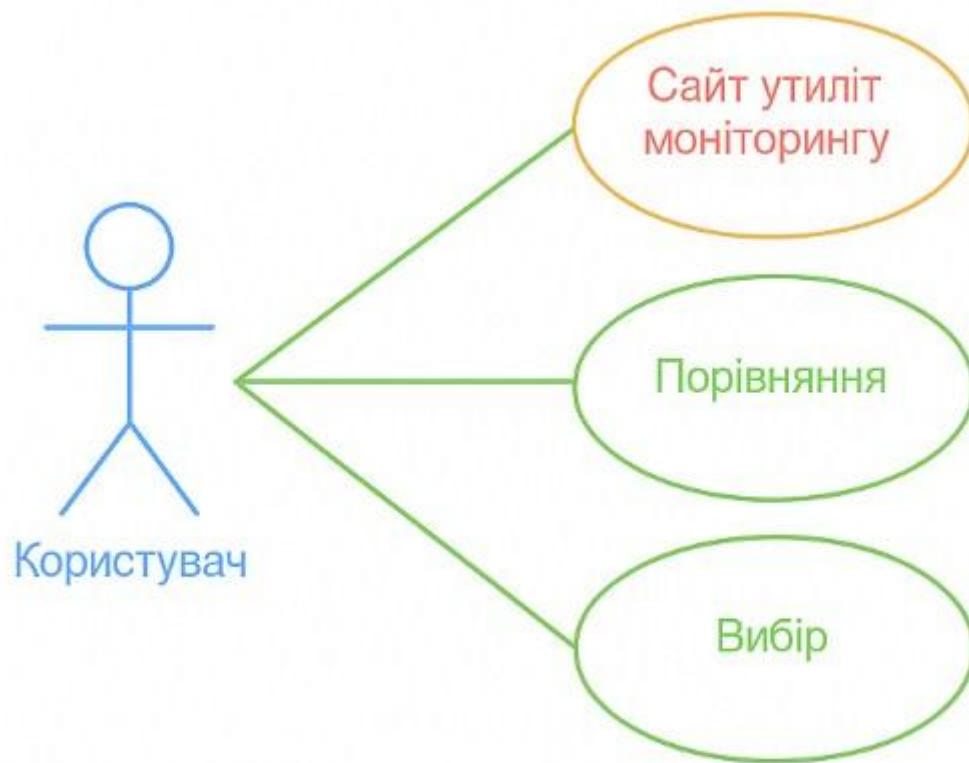
Регулярне спостереження за станом ПК дозволяє:

- Попереджати зношування елементів, таких як вентилятори або накопичувачі.
- Запобігати перевантаженню системи, яке може прискорити знос компонентів.
- Використовувати ефективні заходи для продовження терміну служби апаратного забезпечення.

5. Покращення досвіду користувача

Моніторинг допомагає забезпечити стабільну та ефективну роботу програм і ігор, покращуючи загальний досвід користувача. Наприклад:

- Під час виконання важких обчислювальних задач або гри, моніторинг температури та завантаження ресурсів допомагає запобігти перегріву та зниженню продуктивності, забезпечуючи комфортний користувацький досвід.
- Користувачі можуть отримати повідомлення про необхідність оптимізації системи або заміни компонентів до того, як виникнуть серйозні проблеми.



Діаграма 3.1-- Варіант використання

4. ПРАКТИЧНА ЧАСТИНА

4.1 Вибір платформи та технологій розробки сайту

Для створення інформаційного веб-сайту з порівнянням утиліт для моніторингу апаратного забезпечення ПК було обрано підхід із використанням класичних веб-технологій — HTML, CSS та JavaScript, без залучення складних CMS або фреймворків. Такий вибір зумовлений кількома факторами:

1. HTML

HTML5 використовується як основа структури сайту. Завдяки новим можливостям HTML5 було зручно реалізовано:

- адаптивну верстку;
- структуру сторінок (розділи, заголовки, блоки контенту);
- семантичні елементи (`<header>`, `<section>`, `<article>`, `<footer>`) для покращення SEO та читабельності коду.

2. CSS

CSS3 відповідає за зовнішній вигляд сайту, зокрема:

- кольорову гаму;
- шрифти, інтерфейсні стилі;
- адаптивність через медіа-запити (media queries);
- анімацію елементів, ефекти при наведенні, модальні вікна тощо.

Уся стилізація реалізована у зовнішньому файлі `style.css`, що полегшує редагування й повторне використання стилів.

3. JavaScript

Мова JavaScript була використана для реалізації динамічних функцій сайту:

- інтерактивний пошук по назві утиліти;
- розгортання детальної інформації при натисканні на карточку;
- динамічна побудова або сортування таблиці порівняння;
- модальні вікна для відображення додаткової інформації.

Також планувалося підключення бібліотек (наприклад, jQuery), проте через просту структуру проекту обійшлося використанням нативного JavaScript.

Відмова від CMS та фреймворків керується тим що, система керування контентом (CMS), така як WordPress або Joomla, не була використана з таких причин:

- сайт не потребує складної панелі адміністрування;
- весь контент є статичним та легко редагується без CMS;
- простота реалізації забезпечує менші вимоги до хостингу.

Також не використовувалися фреймворки (наприклад, React, Vue, Angular), оскільки структура сайту не передбачала високої динамічності або SPA (single-page application).

Хостинг

Оскільки сайт має статичну структуру, його можна розмістити на будь-якому безкоштовному або бюджетному хостингу. Серед можливих варіантів:

- GitHub Pages — безкоштовний хостинг для статичних сайтів;
- Netlify або Vercel — також підходять для розгортання невеликих проєктів;
- Localhost — для демонстрації на локальному комп'ютері (наприклад, через Live Server у Visual Studio Code).

Інструменти розробки

- Notepad++ — простий редактор коду з підсвічуванням синтаксису;
- Google Chrome Developer Tools — для відладки та перевірки адаптивності;
- Online HTML/CSS validators — для перевірки на відповідність стандартам W3C.

4.2 Структура та функціонал сайту

Розроблений веб-сайт має інформаційно-аналітичний характер і призначений для ознайомлення користувача з популярними утилітами для моніторингу апаратного забезпечення ПК. Структура сайту є простою, логічною та зручною для користувача, з чітким розподілом інформації за функціональними блоками. Загальна архітектура сайту включає кілька ключових сторінок і елементів інтерфейсу.

Основні сторінки сайту:

1. Головна сторінка

- Вступна інформація про мету сайту.
- Короткий опис функцій моніторингових утиліт.
- Пошукове поле для швидкого знаходження утиліти.
- Список утиліт у вигляді інтерактивних карточок, на яких розміщено назву, кнопки для перегляду деталей.

Зображено на рисунку [4.1].

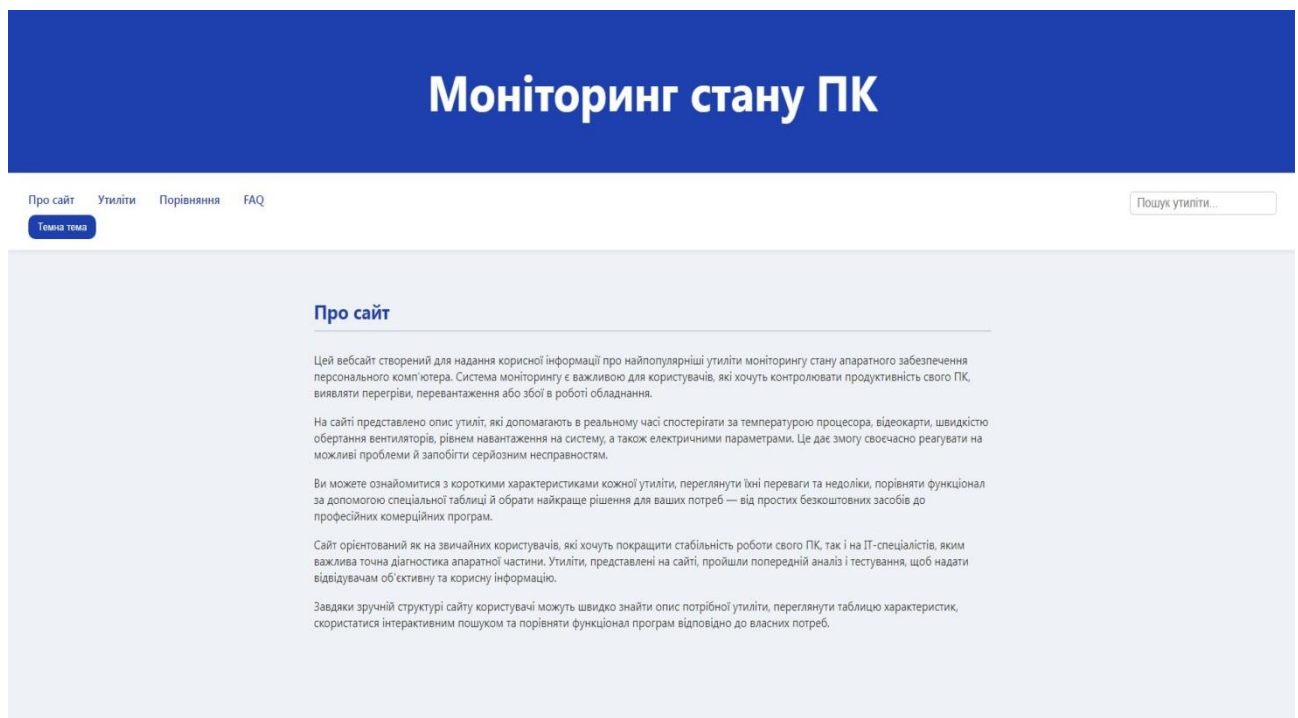


Рис. 4.1 – Головна сторінка сайту

2. Сторінка детального опису утиліт

З’являється при натисканні на карточку утиліти. Зображено на рисунку [4.2].

Містить: розширений опис функціоналу утиліти, її переваги, недоліки, скріншоти (опціонально), платформу підтримки (Windows, macOS, Linux).

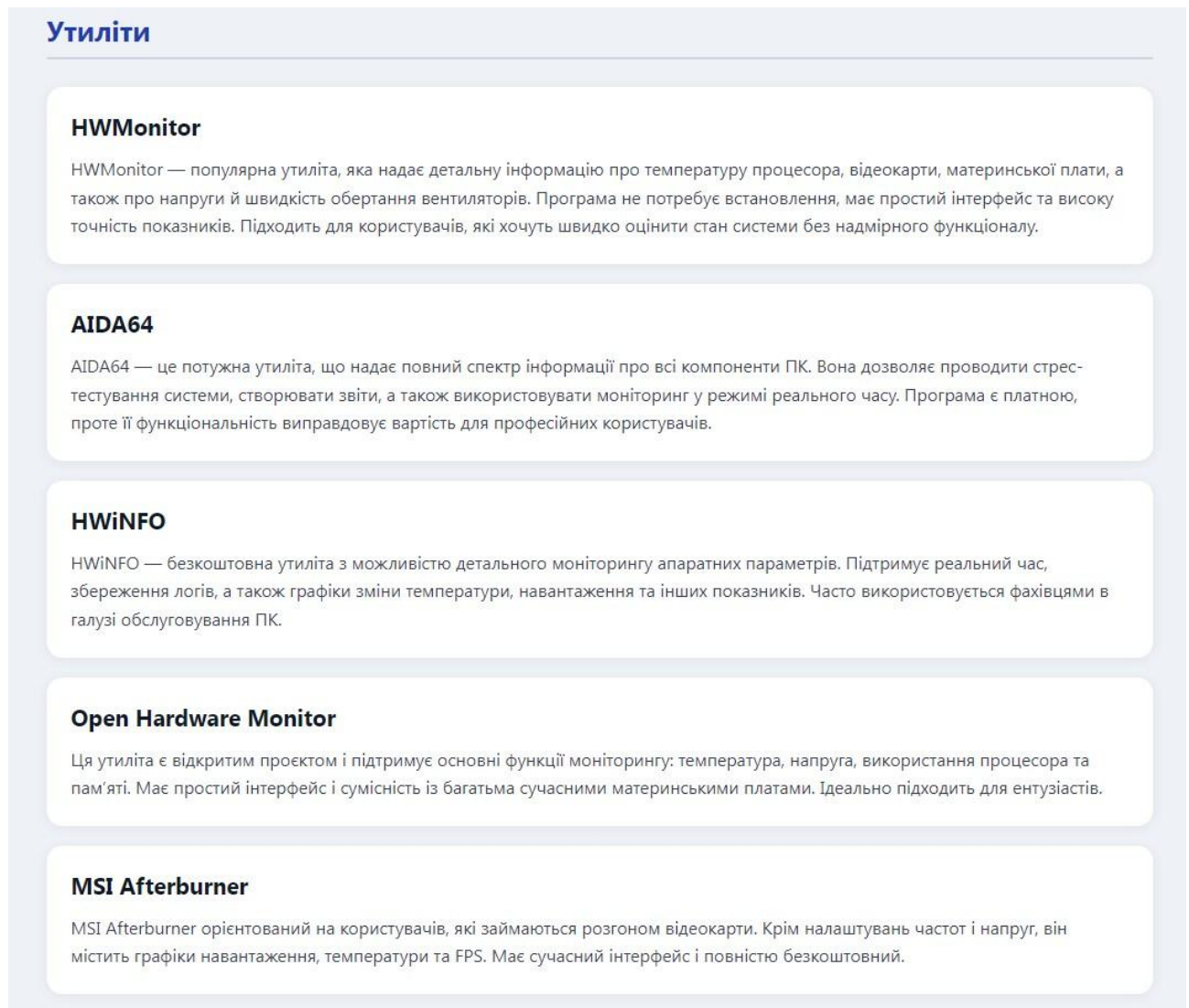


Рис. 4.2 – Детальний опис утиліт

3. Сторінка порівняння

Таблиця, у якій порівнюються утиліти за основними критеріями:

- Підтримувані ОС;
- Тип ліцензії (безкоштовна / платна);
- Підтримка моніторингу температури, напруги, швидкості вентиляторів тощо;

- Інтерфейс (GUI / CLI);
 - Мова інтерфейсу;
 - Рейтинг користувачів (умовний).
- Зображена на рисунку [4.3].

Порівняльна таблиця

НАЗВА	ПРОЦЕСОР	ВІДЕОКАРТА	ТЕМПЕРАТУРА	НАПРУГА	ВЕНТИЛЯТОРИ	ОПЕРАТИВНА ПАМ'ЯТЬ	ІНТЕРФЕЙС
MSI Afterburner	✗	✓ Частота, навантаження, температура	✓	✓	✗	✗	Графіки, вкладки
HWMonitor	✓ Температура, напруга	✓ Температура, частоти	✓	✓	✓ Оберти (RPM)	✓ Частота, обсяг	Таблиця, без графіки
AIDA64	✓ Повна інформація + тести	✓ Вся детальна інформація	✓	✓	✓	✓ Детальна	Багаторівневе меню
Speccy	✓ Основна інформація	✓ Базова інформація	✓	✗	✗	✓	Інтуїтивний, простий
HWiNFO	✓ Дуже детальна + моніторинг	✓ Температури, частоти, навантаження	✓	✓	✓ Всі сенсори	✓ Повна статистика	Таблиці + графіки

Рис. 4.3 – Порівняльна таблиця утиліт

Основні елементи функціоналу:

1. Пошук – Реалізований за допомогою JavaScript по назві утиліт. Працює так при введенні запиту автоматично фільтрує карточки за назвами утиліт на головному вікні зображено на малюнку[4.4] та [4.5].

aida

Рис. 4.4 – Пошукова панель

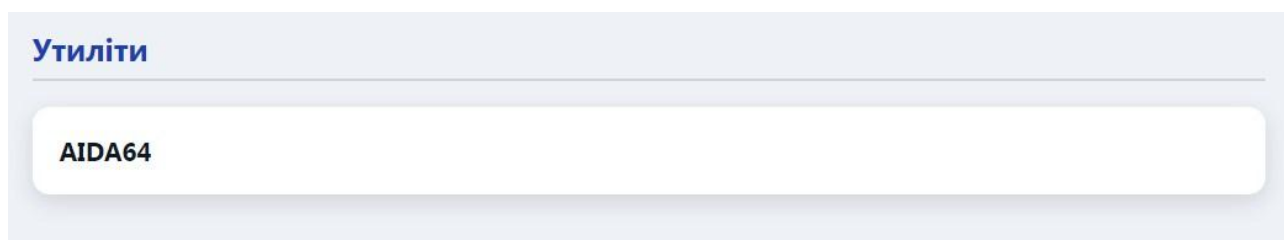


Рис. 4.5 – Результат пошуку

2. Інтерактивні карточки – Кожна карточка містить назву утиліти. При натисканні відкривається модальне вікно з повною інформацією про утиліту зображено на малюнку [4.6] та [4.7].



Рис. 4.6 – Закриті модульні вікна

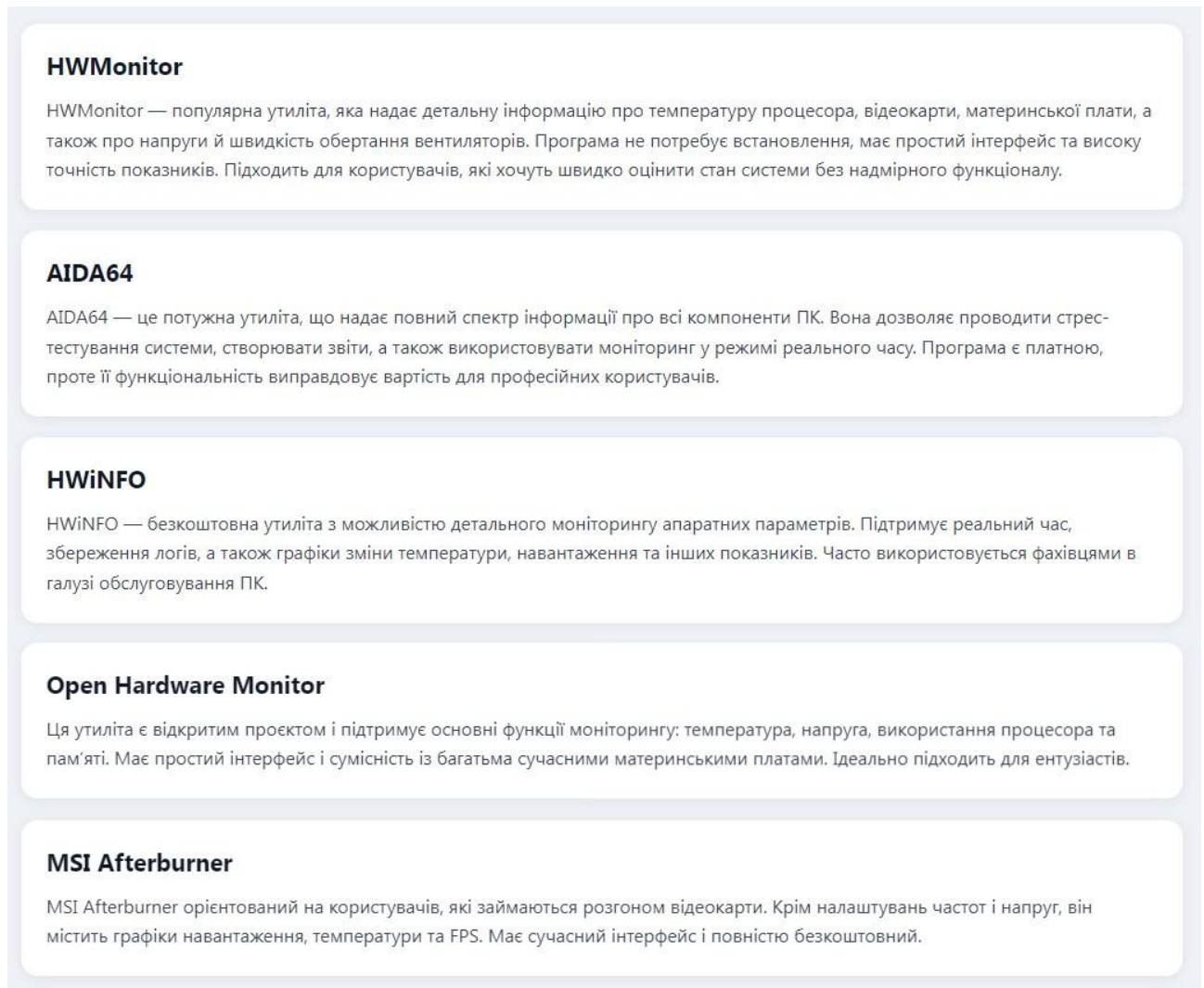


Рис. 4.7 – Відкриті модульні вікна

3. Модальні вікна з'являються поверх сторінки без її перезавантаження. Зручно для ознайомлення з утилітою, не залишаючи головну сторінку.
4. Таблиця порівняння порівнює утиліти за заданими параметрами, також можливе сортування за критеріями.
5. Адаптивний дизайн. Всі елементи сайту коректно відображаються на екранах різного розміру (ПК, планшет, смартфон).

4.3 Дизайн сайту та адаптивність

Дизайн сайту є важливою складовою загального враження користувача від вебресурсу. У рамках розробки сайту, присвяченого порівнянню утиліт для моніторингу апаратного забезпечення, було зроблено акцент на зручність використання, читабельність та сучасний інтерфейс, що поєднує простоту з функціональністю. Крім того, особлива увага була приділена адаптивності інтерфейсу для різних типів пристроїв.

Основні принципи дизайну

1. Мінімалізм і фокус на контенті

Інтерфейс сайту не перевантажений декоративними елементами, що дозволяє користувачу сконцентруватися на головному — інформації про утиліти.

Іконки та логотипи утиліт подані у стислому, але впізнаваному вигляді.

Стиль оформлення витриманий у світло-темній кольоровій палітрі, де основними кольорами є білий, сірий і чорний, що сприяє контрастності та комфортному читанню.

2. Єдина стилістика елементів

Карточки, кнопки, таблиці та модальні вікна мають однаковий стиль із плавними заокругленнями, тінями та анімаціями при наведенні.

Для підвищення візуальної привабливості використовуються ефекти при наведенні на елементи: зміна кольору, підсвічування, анімоване відкриття модальних вікон.

3. Використання сучасних вебтехнологій

Стилізація реалізована за допомогою CSS3 з додаванням анімацій та медіа-запитів.

JavaScript забезпечує інтерактивність: динамічне відкриття інформації, пошук, взаємодію з користувачем.

Шрифт інтерфейсу — безсерифний (наприклад, "Roboto" або "Open Sans"), що покращує читабельність на екранах різного розміру.

Адаптивність сайту

Для сучасного сайту важливо коректно відображатися не лише на десктопах, а й на мобільних пристроях. Тому сайт розроблено із застосуванням принципів адаптивного дизайну, що забезпечує коректну роботу на різних екранах (РС, ноутбук, планшет, смартфон).

Основні аспекти адаптивності:

- Гнучка сітка верстки (flexbox, grid)
Елементи автоматично перебудовуються в залежності від ширини екрану. Наприклад, карточки утиліт на десктопі відображаються в 3 колонки, на планшеті — в 2, на телефоні — в 1.
- Медіа-запити (media queries)
Використовуються для зміни стилів при різних розмірах екранів. Наприклад, змінюється розмір шрифту, ширина блоків, розташування елементів меню.
- Мобільне меню
При вузькому екрані основне меню ховається за іконкою «бургер-меню», що економить простір.
- Адаптивні зображення та шрифти
Зображення масштабуються відповідно до контейнера, а текст залишається добре читабельним навіть на невеликих екранах.

4.4 Реалізація та опис етапів розробки

Розробка сайту проходила кілька послідовних етапів, кожен із яких мав на меті забезпечити якісне, зручне та функціональне веб-рішення для порівняння утиліт моніторингу апаратного забезпечення ПК. Нижче наведено детальний опис кожного етапу.

1. Етап визначення цілей і функціональних вимог сайту.

На першому етапі було сформульовано основну мету проекту — створення інформаційного сайту, що дозволяє користувачу переглядати, порівнювати та отримувати відомості про популярні утиліти для моніторингу стану комп'ютерного обладнання.

Були визначені основні функціональні вимоги:

- представлення інформації про кожну утиліту у вигляді карточок;
- можливість порівняння утиліт;
- пошук по назві утиліти;
- адаптивність сайту;
- простий, інтуїтивний дизайн.

2. Етап вибір технологій.

Для поставлених задач були обрані такі технології:

- HTML5 — для структурування вмісту;
- CSS3 (з використанням Flexbox і Grid) — для візуального оформлення;
- JavaScript — для реалізації динамічної взаємодії (пошук, відкриття інформації);
- VS Code — як основне середовище розробки;
- GitHub Pages — як безкоштовний хостинг для публікації сайту.

3. Етап проектування структури сайту.

Було розроблено схему структури сайту, яка включає:

- головну сторінку з переліком утиліт;
- інтерактивне поле пошуку;
- блок опису функціоналу;
- футер з контактною інформацією та авторством.

4. Етап реалізація основного функціоналу.

На цьому етапі було реалізовано:

- семантичну HTML-структуру;
- стилізацію карточок утиліт та інших блоків;
- JavaScript-скрипти для пошуку та розкриття додаткової інформації при натисканні на карточку;
- стилізацію адаптивного меню та елементів управління для мобільних пристроїв.

5. Етап тестування та налагодження.

Після базової реалізації сайт було протестовано:

- на різних розмірах екранів (десктоп, планшет, смартфон);
- у різних браузерах (Chrome, Firefox, Edge);
- на коректність роботи пошуку, адаптивності, правильність відображення інформації.

Було усунуто помилки верстки, оптимізовано розміри шрифтів, покращено швидкість завантаження сторінок

4.5 Результати тестування сайту

Після завершення основної розробки сайту було проведено тестування з метою перевірки його працездатності, сумісності, зручності використання та відповідності початковим вимогам. Тестування проводилося як вручну, так і за допомогою вбудованих засобів розробника браузера (DevTools). Результати показані в таблиці 4.1.

Таблиця 4.1 Основні аспекти тестування

№	Тестований елемент	Очікуваний результат	Фактичний результат	Статус
1	Завантаження сайту	Сторінка відкривається менш ніж за 2 секунди	1.2 секунди	Успішно
2	Відображення на десктопі	Контент коректно відображається, елементи вирівняні	Відображається правильно	Успішно
3	Відображення на планшеті	Карточки утіліт у 2 колонки, меню адаптивне	Відповідає очікуванням	Успішно
4	Відображення на мобільному пристрої	Елементи автоматично перебудовуються в одну колонку	Відображається коректно	Успішно
5	Пошук утіліти за назвою	При введенні тексту відображаються лише відповідні карточкиколонку	Працює без помилок	Успішно
6	Відкриття додаткової інформації	При натисканні на карточку з'являється опис утіліти	Інформація відкривається	Успішно
7	Адаптивне меню	Меню перетворюється у "бургер" на малих екранах	Працює коректно	Успішно
8	Відображення зображень і шрифтів	Зображення масштабуються, шрифти читабельні	Відображається чітко	Успішно
9	Відсутність помилок у консолі браузера	Не повинно бути критичних помилок або попереджень	Помилки не виявлено	Успішно
10	Взаємодія в різних браузерах	Сайт повинен працювати в Chrome, Firefox, Edge	Працює у всіх перевірених	Успішно

ВИСНОВОК

У ході виконання кваліфікаційної роботи було досягнуто поставленої мети — створено вебсайт, який дозволяє користувачам отримувати структуровану інформацію про найпоширеніші утиліти для моніторингу стану апаратного забезпечення персонального комп'ютера.

На основі проведеного теоретичного аналізу були визначені основні параметри моніторингу, такі як температура компонентів, навантаження CPU та GPU, стан оперативної пам'яті, накопичувачів та напруг живлення. Проведено порівняльний огляд популярних програм для моніторингу — AIDA64, HWMonitor, HWiNFO, Open Hardware Monitor та інших, визначено їх функціональні можливості, переваги та недоліки.

У практичній частині розроблено структуру сайту, обрано оптимальні технології для реалізації (HTML, CSS, JavaScript), створено адаптивний дизайн для коректного відображення на різних пристроях (ПК, планшет, смартфон). Вебресурс містить інформаційні сторінки, таблиці, інфокартки утиліт, зручне меню навігації, а також реалізовано базові механізми взаємодії з користувачем.

Проведене тестування показало, що сайт працює стабільно, швидко завантажується, коректно відображається на різних типах пристроїв та забезпечує користувача усією необхідною інформацією для вибору утиліти моніторингу.

Отже, створений сайт є ефективним інструментом для ознайомлення з можливостями сучасного програмного забезпечення у сфері моніторингу апаратного забезпечення ПК та може бути використаний як довідковий ресурс як для звичайних користувачів, так і для спеціалістів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глинський Я.М., Романюк В.В. Основи Web-програмування. — Львів: Видавництво Львівської політехніки, 2020. — 256 с.
2. Негрей М.Ф., Стасюк О.В. Основи HTML та CSS. — Тернопіль: ТНТУ, 2019. — 142 с.
3. Мельник А.М. JavaScript для початківців. — Харків: Ранок, 2021. — 176 с.
4. Вільямс Р. Основи веб-дизайну. — Київ: Діалектика, 2022. — 288 с.
5. AIDA64 [Електронний ресурс]. — Режим доступу: <https://www.aida64.com>
6. HWiNFO — Professional System Information and Diagnostics [Електронний ресурс]. — Режим доступу: <https://www.hwinfo.com>
7. HWMonitor — CPUID [Електронний ресурс]. — Режим доступу: <https://www.cpuid.com/softwares/hwmonitor.html>
8. Open Hardware Monitor [Електронний ресурс]. — Режим доступу: <https://openhardwaremonitor.org>
9. Спеccy — System Information [Електронний ресурс]. — Режим доступу: <https://www.ccleaner.com/speccy>
10. MDN Web Docs: HTML, CSS, JavaScript [Електронний ресурс]. — Режим доступу: <https://developer.mozilla.org>
11. Bootstrap — The most popular HTML, CSS, and JS library [Електронний ресурс]. — Режим доступу: <https://getbootstrap.com>
12. W3Schools — Web Development Tutorials [Електронний ресурс]. — Режим доступу: <https://www.w3schools.com>
13. Ляшенко В.І. Веб-дизайн: навчальний посібник. — Київ: КНЕУ, 2020. — 198 с.
14. Столярчук О.І. Основи проектування вебсайтів. — Вінниця: ВНТУ, 2019. — 110 с.
15. Responsive Web Design Basics — Google Developers [Електронний ресурс]. — Режим доступу: <https://web.dev/responsive-web-design-basics>

- 16.Сучасні засоби моніторингу апаратного забезпечення ПК / Науковий вісник НУ «Львівська політехніка». — 2021. — №3. — С. 47–52.
- 17.ISO/IEC 9126 — Software Engineering — Product Quality. — International Organization for Standardization, 2001.

ДОДАТОК А. Код сайту

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Моніторинг стану ПК</title>
  <style>
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  margin: 0;
  padding: 0;
  background: #eef1f5;
  color: #333;
}

header {
  background: #1e40af;
  color: white;
  padding: 20px 30px;
  text-align: center;
  font-size: 2rem;
  font-weight: 600;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.15);
}

nav {
  background: #ffffff;
  padding: 14px 30px;
  display: flex;
```

```
flex-wrap: wrap;
justify-content: space-between;
align-items: center;
box-shadow: 0 2px 6px rgba(0, 0, 0, 0.08);
}
```

```
nav a {
  margin-right: 20px;
  text-decoration: none;
  color: #1e40af;
  font-weight: 600;
  transition: color 0.25s ease;
}
```

```
nav a:hover {
  color: #3b82f6;
}
```

```
#searchInput {
  padding: 8px 14px;
  border-radius: 10px;
  border: 1px solid #d1d5db;
  font-size: 1rem;
  transition: box-shadow 0.3s ease;
}
```

```
#searchInput:focus {
  outline: none;
  box-shadow: 0 0 0 2px #3b82f6;
}
```

```
.container {  
  padding: 40px 20px;  
  max-width: 1000px;  
  margin: auto;  
}
```

```
h2 {  
  color: #1e40af;  
  border-bottom: 2px solid #cbd5e1;  
  padding-bottom: 8px;  
  margin-bottom: 25px;  
  font-size: 1.6rem;  
}
```

```
.card {  
  background: white;  
  padding: 22px;  
  margin-bottom: 18px;  
  border-radius: 14px;  
  box-shadow: 0 2px 12px rgba(0, 0, 0, 0.06);  
  transition: transform 0.2s ease, box-shadow 0.2s ease;  
  cursor: pointer;  
}
```

```
.card:hover {  
  transform: translateY(-4px);  
  box-shadow: 0 6px 20px rgba(0, 0, 0, 0.12);  
}
```

```
.card-title {  
  font-weight: 700;  
  font-size: 1.3rem;  
  color: #111827;  
}
```

```
.card-text {  
  display: none;  
  margin-top: 12px;  
  color: #374151;  
  line-height: 1.6;  
  animation: fadeIn 0.3s ease-in-out;  
}
```

```
@keyframes fadeIn {  
  from { opacity: 0; }  
  to { opacity: 1; }  
}
```

```
.table {  
  width: 100%;  
  border-collapse: collapse;  
  background: white;  
  border-radius: 12px;  
  overflow: hidden;  
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);  
}
```

```
.table th {  
  background: #1e40af;
```

```
color: white;
padding: 14px;
text-transform: uppercase;
font-size: 0.95rem;
}
```

```
.table td {
padding: 14px;
text-align: center;
border-bottom: 1px solid #e5e7eb;
}
```

```
.table tbody tr:nth-child(even) {
background: #f9fafb;
}
```

```
footer {
background-color: #1f2937;
color: white;
text-align: center;
padding: 24px;
margin-top: 60px;
font-size: 1rem;
}
```

```
#backToTop {
display: none;
position: fixed;
bottom: 30px;
right: 30px;
```

```
background: #1e40af;
color: white;
border: none;
border-radius: 50%;
width: 48px;
height: 48px;
font-size: 22px;
cursor: pointer;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.25);
transition: background 0.3s ease, transform 0.3s ease;
}
```

```
#backToTop:hover {
  background: #3b82f6;
  transform: scale(1.1);
}
```

```
.dark-theme {
  background: #111827;
  color: #e5e7eb;
}
```

```
.dark-theme header {
  background: #0f172a;
  color: #f9fafb;
}
```

```
.dark-theme nav {
  background: #1f2937;
  color: #e5e7eb;
}
```

```
.dark-theme nav a {  
  color: #93c5fd;  
}
```

```
.dark-theme nav a:hover {  
  color: #60a5fa;  
}
```

```
.dark-theme .card {  
  background: #1e293b;  
  color: #e5e7eb;  
  box-shadow: 0 2px 12px rgba(255, 255, 255, 0.05);  
}
```

```
.dark-theme .card-text {  
  color: #cbd5e1;  
}
```

```
.dark-theme .table {  
  background: #1e293b;  
  color: #e5e7eb;  
}
```

```
.dark-theme .table th {  
  background: #334155;  
}
```

```
.dark-theme .table tbody tr:nth-child(even) {  
  background: #273449;
```

```
}
```

```
.dark-theme footer {
  background-color: #0f172a;
  color: #f9fafb;
}
```

```
.dark-theme #backToTop {
  background: #334155;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Моніторинг стану ПК</h1>
```

```
</header>
```

```
<nav>
```

```
<div style="display: flex; align-items: center; flex-wrap: wrap; width: 100%;
justify-content: space-between;">
```

```
<div style="display: flex; flex-wrap: wrap; gap: 15px;">
```

```
<a href="#about">Про сайт</a>
```

```
<a href="#utilities">Утиліти</a>
```

```
<a href="#compare">Порівняння</a>
```

```
<a href="#faq">FAQ</a>
```

```
</div>
```

```
<input
```

```
type="text"
```

```

    id="searchInput"
    placeholder="Пошук утиліти..."
    oninput="filterCards()"
    style="padding: 5px 10px; border-radius: 5px; border: 1px solid #ccc; font-size:
1rem; margin-top: 10px; margin-left: auto; max-width: 200px;"
  />
</div>
<button id="themeToggle" style="padding: 8px 14px; border-radius: 10px; border:
none; background: #1e40af; color: white; cursor: pointer;">
  Темна тема
</button>
</nav>

```

```
<div class="container" id="about">
```

```
<h2>Про сайт</h2>
```

```
<p>
```

Цей вебсайт створений для надання корисної інформації про найпопулярніші утиліти моніторингу стану апаратного забезпечення персонального комп'ютера.

Система моніторингу є важливою для користувачів, які хочуть контролювати продуктивність свого ПК, виявляти перегрів, перевантаження або збої в роботі обладнання.

```
</p>
```

```
<p>
```

На сайті представлено опис утиліт, які допомагають в реальному часі спостерігати за температурою процесора, відеокарти, швидкістю обертання вентиляторів,

рівнем навантаження на систему, а також електричними параметрами. Це дає змогу своєчасно реагувати на можливі проблеми й запобігти серйозним несправностям.

</p>

<p>

Ви можете ознайомитися з короткими характеристиками кожної утиліти, переглянути їхні переваги та недоліки, порівняти функціонал за допомогою спеціальної таблиці

й обрати найкраще рішення для ваших потреб — від простих безкоштовних засобів до професійних комерційних програм.

</p>

<p>

Сайт орієнтований як на звичайних користувачів, які хочуть покращити стабільність роботи свого ПК, так і на ІТ-спеціалістів, яким важлива точна діагностика апаратної частини.

Утиліти, представлені на сайті, пройшли попередній аналіз і тестування, щоб надати відвідувачам об'єктивну та корисну інформацію.

</p>

<p>

Завдяки зручній структурі сайту користувачі можуть швидко знайти опис потрібної утиліти, переглянути таблицю характеристик, скористатися інтерактивним пошуком та порівняти функціонал програм відповідно до власних потреб.

</p>

</div>

<div class="container" id="utilities">

<h2>Утиліти</h2>

<div class="card" onclick="toggleInfo(this)">

<div class="card-title">HWMonitor</div>

<div class="card-text">HWMonitor — популярна утиліта, яка надає детальну інформацію про температуру процесора, відеокарти, материнської плати, а

також про напруги й швидкість обертання вентиляторів. Програма не потребує встановлення, має простий інтерфейс та високу точність показників. Підходить для користувачів, які хочуть швидко оцінити стан системи без надмірного функціоналу.

AIDA64 — це потужна утиліта, що надає повний спектр інформації про всі компоненти ПК. Вона дозволяє проводити стрес-тестування системи, створювати звіти, а також використовувати моніторинг у режимі реального часу. Програма є платною, проте її функціональність виправдовує вартість для професійних користувачів.

HWiNFO — безкоштовна утиліта з можливістю детального моніторингу апаратних параметрів. Підтримує реальний час, збереження логів, а також графіки зміни температури, навантаження та інших показників. Часто використовується фахівцями в галузі обслуговування ПК.

Ця утиліта є відкритим проєктом і підтримує основні функції моніторингу: температура, напруга, використання процесора та пам'яті. Має простий інтерфейс і сумісність із багатьма сучасними материнськими платами. Ідеально підходить для ентузіастів.

```
</div>
```

```
<div class="card" onclick="toggleInfo(this)">
```

```
<div class="card-title">MSI Afterburner</div>
```

```
<div class="card-text">MSI Afterburner орієнтований на користувачів, які  
займаються розгоном відеокарти. Крім налаштувань частот і напруг, він містить  
графіки навантаження, температури та FPS. Має сучасний інтерфейс і повністю  
безкоштовний.</div>
```

```
</div>
```

```
</div>
```

```
<div class="container" id="compare">
```

```
<h2>Порівняльна таблиця</h2>
```

```
<input type="text" id="tableSearch" placeholder="□ Фільтр по назві або  
можливостях..." style="margin-bottom:15px; padding:8px; width:100%; border-  
radius:8px; border:1px solid #ccc; font-size:1rem;" />
```

```
<table class="table" id="utilityTable">
```

```
<thead>
```

```
<tr>
```

```
<th onclick="sortTable(0)">Назва ⬆️</th>
```

```
<th onclick="sortTable(1)">Процесор ⬆️</th>
```

```
<th onclick="sortTable(2)">Відеокарта ⬆️</th>
```

```
<th onclick="sortTable(3)">Температура ⬆️</th>
```

```
<th onclick="sortTable(4)">Напруга ⬆️</th>
```

```
<th onclick="sortTable(5)">Вентилятори ⬆️</th>
```

```
<th onclick="sortTable(6)">Оперативна пам'ять ⬆️</th>
```

```
<th onclick="sortTable(7)">Інтерфейс ⬆️</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr title="MSI Afterburner: деталі про відеокарту, температуру,  
навантаження.">
```

```
<td>MSI Afterburner</td>
```

```
<td>✗ </td>
```

```
<td>✓ Частота, навантаження, температура</td>
```

```
<td>✓ </td>
```

```
<td>✓ </td>
```

```
<td>✗ </td>
```

```
<td>✗ </td>
```

```
<td>Графіки, вкладки</td>
```

```
</tr>
```

```
<tr title="HWMonitor: моніторинг температур, вентиляторів, напруг.">
```

```
<td>HWMonitor</td>
```

```
<td>✓ Температура, напруга</td>
```

```
<td>✓ Температура, частоти</td>
```

```
<td>✓ </td>
```

```
<td>✓ </td>
```

```
<td>✓ Оберти (RPM)</td>
```

```
<td>✓ Частота, обсяг</td>
```

```
<td>Таблиця, без графіки</td>
```

```
</tr>
```

```
<tr title="AIDA64: комплексна утиліта з тестами та детальним  
моніторингом.">
```

```
<td>AIDA64</td>
```

```
<td>✓ Повна інформація + тести</td>
```

```
<td>✓ Вся детальна інформація</td>
```

```
<td>✓ </td>
```

```

<td>✓</td>

<td>✓</td>

<td>✓ Детальна</td>

<td>Багаторівневе меню</td>
</tr>

<tr title="Spessy: базова утиліта з простим інтерфейсом.">
  <td>Spessy</td>
  <td>✓ Основна інформація</td>
  <td>✓ Базова інформація</td>
  <td>✓</td>
  <td>✗</td>
  <td>✗</td>
  <td>✓</td>
  <td>Інтуїтивний, простий</td>
</tr>

<tr title="HWiNFO: професійна утиліта з повною статистикою.">
  <td>HWiNFO</td>
  <td>✓ Дуже детальна + моніторинг</td>
  <td>✓ Температури, частоти, навантаження</td>
  <td>✓</td>
  <td>✓</td>
  <td>✓ Всі сенсори</td>
  <td>✓ Повна статистика</td>
  <td>Таблиці + графіки</td>
</tr>
</tbody>
</table>
</div>

```

```

<div class="container" id="faq">
  <h2>Часті запитання (FAQ)</h2>
  <div class="card" onclick="toggleInfo(this)">
    <div class="card-title">Яку утиліту обрати для базового моніторингу ПК?</div>
    <div class="card-text">Для базового контролю температури та навантаження підійде HWMonitor або Open Hardware Monitor.</div>
  </div>
  <div class="card" onclick="toggleInfo(this)">
    <div class="card-title">Чи є безкоштовні утиліти з повним функціоналом?</div>
    <div class="card-text">Так, наприклад, HWiNFO має потужний функціонал і не потребує ліцензії для домашнього використання.</div>
  </div>
</div>

<footer style="background-color:#343a40; color:white; text-align:center; padding:15px; margin-top:40px;">
  &copy; 2025 Порівняння утиліт моніторингу ПК | Розроблено в межах кваліфікаційної роботи
</footer>

<button id="backToTop">↑</button>

<script>
function toggleInfo(card) {
  const text = card.querySelector('.card-text');
  text.style.display = text.style.display === 'block' ? 'none' : 'block';
}

```

```

const backToTop = document.getElementById("backToTop");
window.addEventListener("scroll", () => {
  backToTop.style.display = window.scrollY > 300 ? "block" : "none";
});
backToTop.addEventListener("click", () => {
  window.scrollTo({ top: 0, behavior: "smooth" });
});

function filterCards() {
  const input = document.getElementById('searchInput').value.toLowerCase();
  const cards = document.querySelectorAll('.card');
  cards.forEach(card => {
    const title = card.querySelector('.card-title').textContent.toLowerCase();
    card.style.display = title.includes(input) ? 'block' : 'none';
  });
}

const themeToggle = document.getElementById("themeToggle");
const body = document.body;

// Завантажити збережену тему
if (localStorage.getItem("theme") === "dark") {
  body.classList.add("dark-theme");
  themeToggle.textContent = "☐ Світла тема";
}

themeToggle.addEventListener("click", () => {
  body.classList.toggle("dark-theme");
  const isDark = body.classList.contains("dark-theme");
  themeToggle.textContent = isDark ? "☐ Світла тема" : "☐ Темна тема";
  localStorage.setItem("theme", isDark ? "dark" : "light");
});

```

```

});
// □ Пошук по таблиці
document.getElementById("tableSearch").addEventListener("input", function () {
  const filter = this.value.toLowerCase();
  const rows = document.querySelectorAll("#utilityTable tbody tr");
  rows.forEach(row => {
    row.style.display = [...row.cells].some(cell =>
      cell.textContent.toLowerCase().includes(filter)
    ) ? "" : "none";
  });
});

// ↑ Сортювання
function sortTable(columnIndex) {
  const table = document.getElementById("utilityTable");
  const rows = Array.from(table.rows).slice(1);
  let ascending = table.dataset.sortOrder !== "asc";

  rows.sort((a, b) => {
    const textA = a.cells[columnIndex].textContent.trim();
    const textB = b.cells[columnIndex].textContent.trim();
    return ascending ? textA.localeCompare(textB) : textB.localeCompare(textA);
  });

  rows.forEach(row => table.tBodies[0].appendChild(row));
  table.dataset.sortOrder = ascending ? "asc" : "desc";
}
</script>

</body>

```

</html>