

|  |    |
|--|----|
| ВСТУП .....  | 1  |
| 1. Аналіз предметної області .....                 | 3  |
| 1.1. Огляд предметної області .....                | 3  |
| 1.2. Постановка задачі .....                       | 4  |
| 2. Проектування бази даних .....                   | 7  |
| 2.1. Інфологічне та даталогічне проектування ..... | 7  |
| 2.1.1. Інфологічне проектування .....              | 7  |
| 2.1.2. Даталогічне проектування .....              | 8  |
| 2.2. Проектування серверної частини .....          | 8  |
| 2.2.1. Схема і об'єкти бази даних .....            | 8  |
| 2.2.2. ERD структура .....                         | 9  |
| 3. ОПИС КЛІЄНТСЬКОГО ДОДАТКУ .....                 | 13 |
| ВИСНОВОК .....                                     | 16 |

## **ВСТУП**

На сучасному етапі стрімкого розвитку інформаційних технологій, роль відеоадаптерів (графічних процесорів, GPU) набуває все більшої значущості. Ці складні електронні пристрої є серцем будь-якої системи, де візуалізація даних, складні обчислення та обробка графіки є критично важливими. Їхня незамінність проявляється у широкому спектрі застосувань: від забезпечення високої продуктивності та реалістичності в ігровій індустрії та віртуальній реальності, до прискорення професійних завдань у графічному дизайні, 3D-моделюванні та відеомонтажі. Останні роки особливо підкреслили їхню роль у наукових обчисленнях, криптографії та, що найважливіше, у сфері штучного інтелекту та машинного навчання, де GPU забезпечують паралельну обробку величезних обсягів даних, необхідних для тренування нейронних мереж.

Проте, незважаючи на критичну важливість відеоадаптерів, процес їхнього вибору є значним викликом як для пересічного користувача, так і для фахівця. Ринок переповнений величезною кількістю моделей від різних виробників, кожна з яких має унікальний набір технічних характеристик (обсяг і тип відеопам'яті, ширина шини, тактові частоти, вимоги до живлення) та динамічну цінову політику. Без структурованого підходу, пошук оптимального рішення, яке б відповідало конкретним потребам і бюджету, стає надзвичайно складним і ресурсозатратним завданням, часто призводячи до неоптимальних рішень.

Метою даної курсової роботи є розробка сучасної, ефективної та інтуїтивно зрозумілої інформаційної системи для комплексного зберігання, аналізу та пошуку даних про відеоадаптери. Створювана система буде спрямована на подолання зазначених труднощів, надаючи користувачам потужний інструментарій для:

- Швидкого та точного отримання інформації про тисячі моделей GPU.
- Детального аналізу їхніх технічних характеристик та порівняння параметрів.

- Відстеження динаміки цін для прийняття обґрунтованих рішень щодо покупки.
- Формування аналітичних звітів та рекомендацій, що спростить процес вибору та допоможе краще розуміти ринкові тенденції.

Для реалізації цього амбітного проекту обрано надійний та перевірений стек технологій. В якості системи керування базами даних (СКБД) використовується PostgreSQL, яка відома своєю високою надійністю, продуктивністю, підтримкою складних запитів та масштабованістю, що є критично важливим для великих обсягів даних. Серверна частина системи розроблена на Python з використанням фреймворку FastAPI, що забезпечує високу швидкість розробки, асинхронність обробки запитів та автоматичну генерацію документації API. Клієнтська частина буде створена за допомогою JavaScript, забезпечуючи динамічний та інтерактивний користувацький інтерфейс. Розробка здійснюється в VS Code з використанням відповідних розширень, таких як FastAPI та SQLTools, для оптимізації процесу кодування та взаємодії з базою даних. Такий підхід дозволить створити повноцінне веб-додаток, здатне ефективно вирішувати поставлені задачі.

## **1. Аналіз предметної області**

### **1.1. Огляд предметної області**

Цей розділ закладає фундамент для розуміння того, з чим буде працювати система. Він пояснює, наскільки складними та важливими є сучасні відеоадаптери (графічні процесори, GPU). Вони є ключовими елементами в багатьох сферах – від комп'ютерної графіки та ігор до наукових обчислень та машинного навчання.

Звіт виділяє основні характеристики відеоадаптерів, які є критично важливими для їхнього опису та класифікації:

- **Виробник:** Перелічуються ключові гравці на ринку GPU, такі як NVIDIA, AMD та Intel, а також спеціалізовані бренди, що випускають готові відеокарти (ASUS, MSI, Gigabyte). Це важливо, оскільки виробник часто визначає екосистему та доступні технології.

- **Модель:** Наводяться конкретні приклади моделей від кожного виробника, що демонструє різноманіття пропозицій (наприклад, NVIDIA RTX 4090, AMD RX 7900 XTX).

- **Технічні параметри:** Це серце опису відеоадаптера. До них належать:
  - **Обсяг відеопам'яті:** Вказує на кількість даних, які GPU може зберігати та швидко обробляти. Діапазон від 2 ГБ до 48 ГБ показує величезну різницю в можливостях.

- **Тип пам'яті:** (GDDR5, GDDR6, GDDR6X, HBM2) – цей параметр прямо впливає на швидкість обміну даними між GPU та пам'яттю, що є критичним для продуктивності.

- **Ширина шини пам'яті:** (64-384 біт) – ще один ключовий показник пропускної здатності пам'яті, який прямо впливає на те, наскільки швидко GPU може отримувати доступ до даних.

- **Тактова частота ядра:** (MHz/GHz) – визначає швидкість виконання обчислювальних операцій самим графічним процесором.

- **Максимальна роздільна здатність:** Вказує на те, які стандарти відображення (4K, 8K, VR) підтримує відеоадаптер.

- **Вимоги до живлення (TDP у ватах):** Важливий параметр для користувачів при виборі блоку живлення та оцінці енергоспоживання.

- **Ціноутворення:** Цей аспект підкреслює значну варіативність цін (\$200 до \$2000+), яка залежить від технічних характеристик, дати виходу, ринкового попиту та наявності передових технологій, таких як трасування променів.

## 1.2. Постановка задачі

Цей розділ чітко визначає, що саме повинна робити розроблювана інформаційна система.

Мета проекту: Створення системи для комплексного управління даними про відеоадаптери, яка дозволить ефективно шукати, аналізувати та візуалізувати інформацію. Це означає, що система не просто зберігатиме дані, але й надаватиме інструменти для їхнього осмислення.

Функціональні вимоги до системи: Детально описують можливості, які має надавати система:

- Управління даними:
  - Повне зберігання інформації про всі характеристики GPU.
  - Підтримка історії змін цін, що є дуже цінною функцією для відстеження ринкових тенденцій.
  - Можливість додавання нових моделей та виробників, що забезпечує гнучкість системи.
- Пошукові можливості:
  - Фільтрація за основними параметрами (виробник, ціновий діапазон) для швидкого пошуку.
  - Розширений пошук за технічними характеристиками, що дозволяє користувачам знаходити моделі за конкретними вимогами (діапазон пам'яті, тип пам'яті, тактова частота, живлення, рік випуску).
- Аналітичні функції:
  - Формування звітів: Наприклад, "ТОП-10 найпродуктивніших моделей", аналіз цінової політики за виробниками, порівняння характеристик обраних моделей. Це перетворює сирі дані на корисну інформацію.
  - Візуалізація даних: Графіки зміни цін та діаграми розподілу характеристик зроблять аналіз більш наочним та зрозумілим.
- Інтерфейс користувача:
  - Інтуїтивно зрозуміле табличне представлення даних для легкості сприйняття.
  - Можливість сортування за будь-якою колонкою.

- Форми для додавання/редагування даних з валідацією введення для забезпечення цілісності даних.
- Експорт даних у зручних форматах (CSV, PDF) для подальшого використання.
- Додаткові функції:
  - Система рекомендацій на основі введених параметрів, що може бути дуже корисним для користувачів, які не є експертами.
  - Можливість порівняння обраних моделей для зручного вибору.
  - Інтеграція з API виробників для автоматичного оновлення даних – це амбітна, але дуже цінна функція для підтримки актуальності.

Нефункціональні вимоги: Описують якість та обмеження системи:

- Багатокористувацький режим: Система повинна підтримувати одночасну роботу декількох користувачів.
- Швидкість відгуку (<1с): Основні операції повинні виконуватися дуже швидко.
- Захист даних: Авторизація та резервне копіювання є критичними для безпеки.
- Адаптивний інтерфейс: Система повинна коректно відображатися на різних пристроях.

Актуальність системи: Підсумовуються переваги створення такої системи: спрощення вибору GPU, аналіз ринкових тенденцій, порівняння характеристик та економія часу.

## 2. Проектування бази даних

### 2.1. Інфологічне та даталогічне проектування

Цей розділ описує, як інформація з предметної області буде організована та зберігатися в базі даних.

#### 2.1.1. Інфологічне проектування

Тут представлена концептуальна модель бази даних, яка є абстрактним описом сутностей та зв'язків між ними, незалежно від конкретної СУБД. Визначено чотири основні сутності:

- Виробники (manufacturers): Зберігає базову інформацію про компанії.
  - id (унікальний ідентифікатор)
  - name (назва виробника)
- Моделі відеокарт (gpu\_models): Описує конкретні моделі, пов'язані з виробниками.
  - id (унікальний ідентифікатор)
  - Назва моделі
  - Посилання на виробника (зовнішній ключ до manufacturers)
  - Рік випуску

- Технічні характеристики (specifications): Деталізує параметри кожної моделі.
  - Обсяг пам'яті
  - Тип пам'яті
  - Ширина шини
  - Базова тактова частота
  - Максимальна роздільна здатність
  - Вимоги до живлення
- Ціни (prices): Зберігає історію цін, пов'язану з моделями.
  - Поточний ідентифікатор (мається на увазі id для запису ціни)
  - Посилання на модель (зовнішній ключ до `gpu_models`)
  - Вартість
  - Дата оновлення ціни

#### 2.1.2. Даталогічне проектування

Це вже фізична реалізація бази даних, тобто, як ці сутності будуть представлені у вигляді таблиць з конкретними даними. Наведено приклади даних, які можуть міститися в кожній таблиці:

- Таблиця `manufacturers`: Містить 14 записів, наприклад, NVIDIA, AMD, Intel, ASUS.
- Таблиця `gpu_models`: Містить 14 записів про моделі, такі як GeForce RTX 4080 (2022) та Radeon RX 7900 XTX (2022), з вказівкою зв'язку з виробником (`manufacturer_id`).
- Таблиця `specifications`: Описує детальні параметри, наприклад, обсяг пам'яті від 8192 МБ до 24576 МБ, типи пам'яті GDDR6/GDDR6X, ширина шини 128-384 біт. Важливо, що ця таблиця, ймовірно, буде пов'язана з `gpu_models` через зовнішній ключ, щоб кожна модель мала свої специфікації.
- Таблиця `prices`: Містить історію цін з діапазоном від \$199.99 до \$1799.99 та датами оновлення від 2022 до 2025 року.

### 2.2. Проектування серверної частини

#### 2.2.1. Схема і об'єкти бази даних



Цей підрозділ узагальнює основні принципи проектування бази даних:

- Чітка структура з чотирьох взаємопов'язаних таблиць.
- Наявність первинних ключів (id) у кожній таблиці для забезпечення унікальності записів.
- Використання відповідних типів даних для кожного поля (наприклад, текстовий для назв, числовий для обсягу пам'яті, дата для дати оновлення).
- Встановлення зв'язків між таблицями через зовнішні ключі, що забезпечує цілісність даних (наприклад, `manufacturer_id` в `gpu_models` посилається на `id` в `manufacturers`).
- Встановлення обмежень (CHECK) для числових значень, що допомагає запобігти введенню недійсних даних (наприклад, негативного обсягу пам'яті).

### 2.2.2. ERD структура

ERD (Entity-Relationship Diagram) структура — це графічне представлення зв'язків між сутностями в базі даних. У звіті не наведено саму діаграму, але з опису зрозуміло, що вона повинна відображати наступні зв'язки:

- "Один-до-багатьох" між `manufacturers` та `gpu_models`: Один виробник може випускати багато моделей відеокарт.
- "Один-до-одного" або "один-до-багатьох" між `gpu_models` та `specifications`: Залежно від того, як задумано зберігання специфікацій. Якщо одна модель має один набір специфікацій, це "один-до-одного". Якщо специфікації можуть змінюватися з часом для однієї моделі (наприклад, різні версії однієї моделі), то це може бути "один-до-багатьох", де кожна версія специфікації пов'язана з моделлю. З наведеного опису "детальний опис параметрів кожної моделі" це більше схоже на "один-до-одного".
- "Один-до-багатьох" між `gpu_models` та `prices`: Одна модель відеокарти може мати багато записів про ціну, оскільки ціни змінюються з часом.

Загальна структура: Діаграма складається з чотирьох таблиць, які представляють основні сутності інформаційної системи для управління даними

про відеоадаптери. Кожна таблиця має назву, список стовпців (атрибутів) та позначення ключів і типів даних. Стрілки показують зв'язки між таблицями.

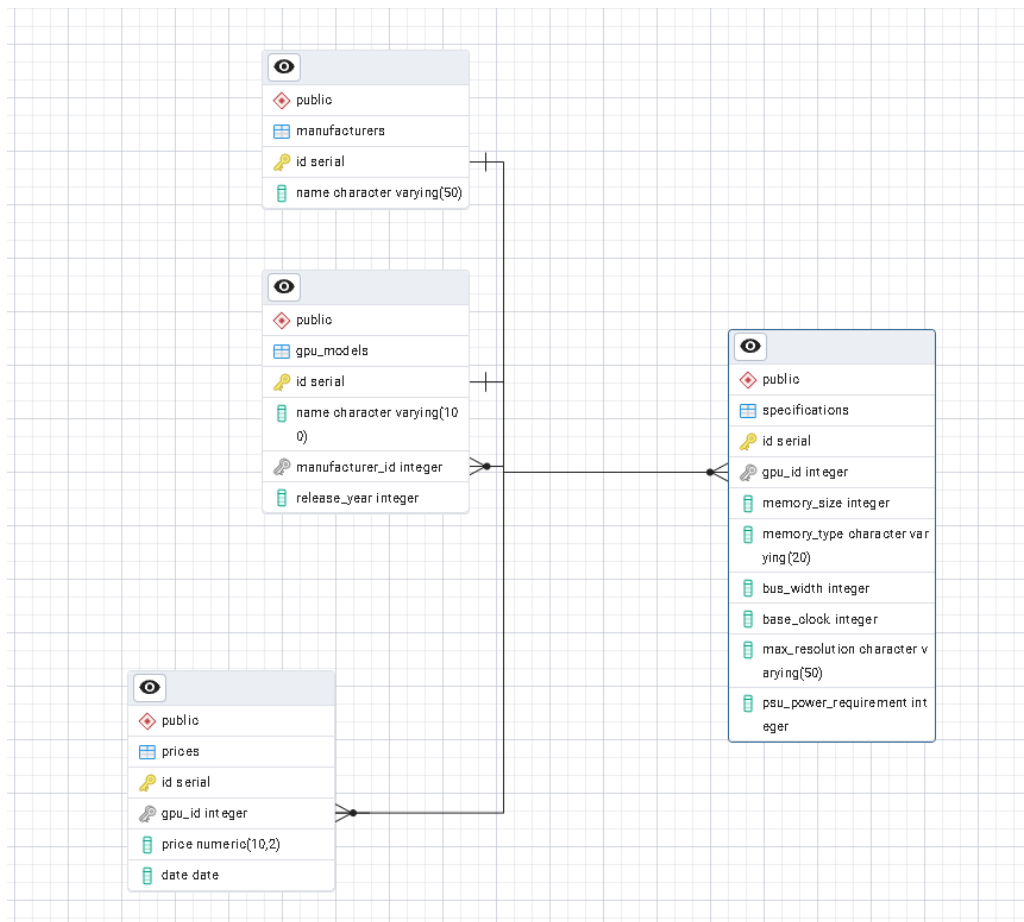


Рис. 1. ERD for database

Опис кожної таблиці (сутності):

1. manufacturers (Виробники)

a. Призначення: Зберігає інформацію про компанії-виробників графічних процесорів.

b. Стовпці (Атрибути):

i. id (serial): Первинний ключ, унікальний ідентифікатор для кожного виробника. Тип serial зазвичай означає ціле число, яке автоматично збільшується.

ii. name (character varying(50)): Назва виробника. Тип character varying(50) означає текстовий рядок змінної довжини до 50 символів.

c. Ключі: id позначений як первинний ключ (символ золотого ключа).

## 2. `gpu_models` (Моделі GPU)

a. Призначення: Зберігає інформацію про конкретні моделі відеокарт.

b. Стовпці (Атрибути):

i. `id (serial)`: Первинний ключ, унікальний ідентифікатор для кожної моделі GPU.

ii. `name (character varying(100))`: Назва моделі (наприклад, "RTX 4090", "RX 7900 XTX"). Тип `character varying(100)` означає текстовий рядок змінної довжини до 100 символів.

iii. `manufacturer_id (integer)`: Зовнішній ключ, який посилається на `id` таблиці `manufacturers`. Це пов'язує модель з її виробником.

iv. `release_year (integer)`: Рік випуску моделі. Тип `integer` означає ціле число.

c. Ключі: `id` позначений як первинний ключ. `manufacturer_id` не позначений як зовнішній ключ явно на цій діаграмі, але зв'язок це показує.

## 3. `specifications` (Специфікації)

a. Призначення: Зберігає детальні технічні характеристики для кожної моделі GPU.

b. Стовпці (Атрибути):

i. `id (serial)`: Первинний ключ, унікальний ідентифікатор для кожного набору специфікацій (теоретично, якщо модель може мати різні специфікації).

ii. `gpu_id (integer)`: Зовнішній ключ, який посилається на `id` таблиці `gpu_models`. Це пов'язує специфікації з конкретною моделлю GPU.

iii. `memory_size (integer)`: Обсяг відеопам'яті в МБ.

iv. `memory_type (character varying(20))`: Тип пам'яті (наприклад, "GDDR6X", "HBM2").

v. `bus_width (integer)`: Ширина шини пам'яті в бітах.

vi. `base_clock (integer)`: Базова тактова частота ядра в МГц.

vii. `max_resolution (character varying(50))`: Максимальна підтримувана роздільна здатність (наприклад, "8K", "4K").

viii. `psu_power_requirement` (integer): Вимоги до живлення блоку живлення (TDP) у ватах.

с. Ключі: `id` позначений як первинний ключ. `gpu_id` не позначений як зовнішній ключ явно, але зв'язок це показує.

4. `prices` (Ціни)

a. Призначення: Зберігає історію цін для кожної моделі GPU.

b. Стовпці (Атрибути):

i. `id` (serial): Первинний ключ, унікальний ідентифікатор для кожного запису ціни.

ii. `gpu_id` (integer): Зовнішній ключ, який посилається на `id` таблиці `gpu_models`. Це пов'язує запис ціни з конкретною моделлю GPU.

iii. `price` (numeric(10,2)): Вартість відеокарти. Тип `numeric(10,2)` означає число з 10 цифрами всього, з яких 2 після десяткової крапки (тобто до 99999999.99).

iv. `date` (date): Дата оновлення або фіксації ціни. Тип `date` означає дату без часу.

с. Ключі: `id` позначений як первинний ключ. `gpu_id` не позначений як зовнішній ключ явно, але зв'язок це показує.

Зв'язки між таблицями (Relationship):

Зв'язки представлені лініями зі стрілками або позначеннями.

- `manufacturers <-- "Один-до-багатьох" --> gpu_models`:

- Лінія з позначкою "плюс" (+) на стороні `gpu_models` означає, що один виробник може мати багато моделей GPU, але кожна модель GPU належить лише одному виробнику. Це стандартний зв'язок "один-до-багатьох". Зовнішній ключ `manufacturer_id` в `gpu_models` реалізує цей зв'язок.

- `gpu_models <-- "Один-до-багатьох" --> specifications`:

- Лінія з позначкою "плюс" (+) на стороні `specifications` означає, що одна модель GPU може мати багато наборів специфікацій. Це дозволяє зберігати різні версії специфікацій для однієї моделі або, наприклад, відстежувати зміни в специфікаціях (хоча для цього зазвичай додають поле "дата" в `specifications`). В

контексті "один-до-одного" зв'язку (як це зазвичай буває, коли модель має один набір специфікацій), ця діаграма передбачає можливість множинних специфікацій для однієї моделі. Зовнішній ключ `gpu_id` в `specifications` реалізує цей зв'язок.

- `gpu_models <-- "Один-до-багатьох" --> prices:`
  - Лінія з позначкою "плюс" (+) на стороні `prices` означає, що одна модель GPU може мати багато записів про ціни. Це абсолютно логічно, оскільки ціни на відеокарти змінюються з часом, і система має відстежувати цю історію. Зовнішній ключ `gpu_id` в `prices` реалізує цей зв'язок.

### 3. ОПИС КЛІЄНТСЬКОГО ДОДАТКУ

Призначення: Веб-додаток для керування базою даних відеокарт, призначений для адміністраторів комп'ютерних магазинів, геймерів та ІТ-ентузіастів.

Основні функціональні можливості:

1. Робота з базою даних відеокарт:
  - a. Перегляд повного списку відеокарт з усіма характеристиками
  - b. Додавання нових відеокарт до бази даних
  - c. Редагування існуючих записів
  - d. Видалення відеокарт з бази даних
2. Розширена система пошуку:

- a. Пошук за виробником (наприклад, NVIDIA, AMD, Intel)
- b. Пошук за моделлю відеокарти (наприклад, RTX 4090, RX 7800 XT)
- c. Інтуїтивний пошук з автозаповненням
- 3. Потужна система сортування:
  - a. Сортування за ціною (від дешевих до дорогих і навпаки)
  - b. Сортування за потужністю блоку живлення
  - c. Сортування за максимальною роздільною здатністю
  - d. Сортування за базовою частотою
  - e. Сортування за шириною шини пам'яті
  - f. Сортування за об'ємом пам'яті
  - g. Сортування за роком випуску
- 4. Детальний перегляд характеристик:
  - a. Відображення всіх технічних характеристик відеокарт:
    - i. Виробник та модель
    - ii. Рік випуску
    - iii. Об'єм та тип пам'яті
    - iv. Ширина шини пам'яті
    - v. Базова частота
    - vi. Максимальна роздільна здатність
    - vii. Вимоги до блоку живлення
    - viii. Актуальна ціна
- 5. Зручний інтерфейс адміністрування:
  - a. Модальні вікна для додавання/редагування записів
  - b. Підтвердження дій (особливо при видаленні)
  - c. Валідація введених даних
  - d. Адаптивний дизайн для різних пристроїв
- 6. Технічні особливості реалізації:
  - a. Frontend: HTML5, CSS3, JavaScript (чистий JS без бібліотек)
  - b. Backend: FastAPI (Python)
  - c. База даних: PostgreSQL

- d. Архітектура: REST API
- e. Асинхронні запити до сервера
- f. Реалізація CRUD (Create, Read, Update, Delete)

Переваги додатку:

- 1. Зручність використання: Інтуїтивно зрозумілий інтерфейс на українській мові.
- 2. Гнучкість: Можливість швидко знаходити потрібні відеокарти за будь-якими параметрами.
- 3. Актуальність: Можливість оновлювати ціни та характеристики.
- 4. Мобільність: Адаптивний дизайн дозволяє використовувати на різних пристроях.
- 5. Безпека: Підтвердження критичних дій (видалення, зміна даних).

Сценарії використання:

- 1. Для менеджера магазину:
  - a. Швидке додавання нових моделей відеокарт
  - b. Оновлення цін
  - c. Підбір асортименту за характеристиками
- 2. Для покупця:
  - a. Порівняння характеристик різних моделей
  - b. Підбір відеокарти за бюджетом
  - c. Перевірка технічних вимог
- 3. Для аналітика:
  - a. Аналіз ринкових тенденцій
  - b. Порівняння продуктивності різних років випуску
  - c. Визначення оптимального співвідношення ціна/якість

Перспективи розвитку:

- 1. Додавання функції порівняння відеокарт
- 2. Інтеграція з API інтернет-магазинів для автоматичного оновлення цін
- 3. Додавання графіків та статистики

4. Розширена система фільтрів
5. Можливість експорту даних у різних форматах

## **ВИСНОВОК**

У рамках даної курсової роботи було успішно реалізовано важливу мету – розробку **сучасної інформаційної системи для комплексного управління даними про відеоадаптери**. Ця система, від початку до кінця спроектована з урахуванням актуальних потреб користувачів ринку GPU, є наочним підтвердженням можливості ефективного поєднання функціональності, зручності та надійності.

**Ключові досягнення проекту включають:**

**1. Глибокий аналіз предметної області:** Було ретельно вивчено світ сучасних відеоадаптерів, ідентифіковано їхні ключові характеристики, виробників, моделі та фактори, що впливають на ціноутворення. Це дозволило сформулювати комплексне розуміння даних, з якими працює система.

**2. Чітка постановка задачі та вимог:** На основі аналізу були сформульовані детальні функціональні та нефункціональні вимоги до системи. Це забезпечило прозорість цілей проекту та визначило шлях до їх реалізації.

**3. Надійне проектування бази даних:** Інфологічне та даталогічне проектування призвело до створення структурованої та нормалізованої схеми бази даних на базі **PostgreSQL**. Визначені сутності (manufacturers, gpu\_models, specifications, prices) та їхні взаємозв'язки забезпечують цілісне



зберігання інформації, включаючи історію цін, що є критично важливою для аналізу ринкових тенденцій.

**4. Ефективна архітектура рішення:** Застосування сучасного стеку технологій – **Python (FastAPI)** для серверної частини та **JavaScript** для клієнтської – дозволило створити гнучку, масштабовану та високопродуктивну систему з архітектурою **REST API**. Асинхронні запити та реалізація CRUD-операцій забезпечують швидкий відгук та безперебійну роботу.

**5. Розширений функціонал клієнтського додатку:** Розроблений веб-додаток є не просто сховищем даних, а потужним інструментом з:

а. **Комплексним управлінням даними:** Можливість додавання, редагування та видалення записів про відеокарти.

б. **Просунутою системою пошуку та сортування:** Фільтрація за виробником, моделлю, та всіма основними технічними характеристиками (ціна, потужність БЖ, роздільна здатність, частота, ширина шини, об'єм пам'яті, рік випуску).

в. **Детальним переглядом характеристик:** Зручне відображення всієї інформації про обрану модель.

г. **Інтуїтивно зрозумілим інтерфейсом адміністрування:** Модальні вікна, валідація даних та підтвердження дій значно покращують користувацький досвід.

д. **Адаптивним дизайном:** Забезпечує коректне відображення та функціональність на різних пристроях.

Даний додаток є **повноцінною системою управління базою даних відеокарт**, яка поєднує в собі зручний інтерфейс, потужні інструменти пошуку та сортування, а також всі необхідні функції для роботи з даними. Його цінність полягає в здатності **значно спростити процес вибору та аналізу відеоадаптерів** для широкого кола користувачів – від менеджерів комп'ютерних магазинів, яким потрібно швидко оновлювати асортимент та ціни, до геймерів та

ІТ-ентузіастів, які прагнуть знайти оптимальне рішення за своїми критеріями, а також аналітиків, що досліджують ринкові тенденції.

**Перспективи подальшого розвитку** системи є значними та передбачають: інтеграцію функції порівняння відеокарт для ще більш наочного вибору, підключення до зовнішніх API інтернет-магазинів для автоматичного оновлення цін, додавання графіків та статистики для поглибленого аналізу даних, розширення системи фільтрів та можливість експорту даних у різноманітних форматах. Ці доповнення дозволять системі адаптуватися до мінливих ринкових умов та задовольнити ще ширший спектр потреб користувачів.

Таким чином, розроблена інформаційна система не тільки відповідає всім поставленим завданням, а й закладає міцний фундамент для майбутніх розширень, підтверджуючи свою актуальність та потенціал для широкого практичного застосування.