МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп`ютерної інженерії та кібербезпеки

**КУРСОВА РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об’єктно-орієнтоване програмування»

на тему:

**«Інформаційно-пошукова система: сучасні моделі відеокарт»**

студента І курсу групи КБ-21-2

спеціальності 125 «Кібербезпека»

\_\_Новіцького Віктора Сергійовича\_\_\_\_\_\_\_\_

(прізвище, ім’я та по-батькові)

Керівник ст. викладач кафедри КН\_\_\_\_\_\_\_

, Левківський В.Л.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Л. Левківський \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Марчук .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Граф \_

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра комп`ютерної інженерії та кібербезпеки

Освітній рівень: бакалавр

Спеціальність 125 «Кібезбезпека»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри КІ та КБ

\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Єфіменко

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

Новіцькому Віктору Сергійовичу

1. Тема роботи: «Розробка інформаційно-пошукової системи: сучасні моделі відеокарт»,

керівник роботи: старший викладач кафедри комп’ютерних наук Левківський Віталій Леонідович.

1. Строк подання студентом: “ 13 ” липня 2022р.
2. Вихідні дані до роботи: Розробити інформаційно-пошукову систему: сучасні моделі відеокарт.
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Посилання на репозиторій: \_<https://gitlab.com/kb212_nvs/term-paper>\_\_\_\_\_\_\_\_\_\_

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 15 ” квітня 2022 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Постановка задачі | 17.04.2022 |  |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 20.04.2022 |  |
| 3 | Формулювання технічного завдання | 02.05.2022 |  |
| 4 | Опрацювання літературних джерел | 03.06.2022 |  |
| 5 | Проектування структури | 15.06.2022 |  |
| 6 | Написання програмного коду | 30.06.2022 |  |
| 7 | Відлагодження | 05.07.2022 |  |
| 8 | Написання пояснювальної записки | 10.07.2022 |  |
| 9 | Захист |  |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Новіцький В.С.

(підпис) (прізвище та ініціали)

**Керівник роботи**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Левківський В.Л.

(підпис) (прізвище та ініціали)

**РЕФЕРАТ**

Завданням на курсову роботу було створення інформаційно-пошукової системи для пошуку сучасних модель відеокарт.

Пояснювальна записка до курсової роботи на тему «Інформаційно-пошукова система: сучасні моделі відеокарт» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 34 сторінках друкованого тексту.

Пояснювальна записка має 24 сторінок додатків. Список використаних джерел містить 12 найменувань і займає 2 сторінки. В роботі наведено 30 рисунків. Загальний обсяг роботи – 58 сторінок.

Ключові слова: C#, SQL, WINDOWS FORMS, ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, ВІДЕОКАРТА, БАЗА ДАНИХ.

**ЗМІСТ**

ВСТУП 6

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДИ, ТА ЗАСОБИ ВИРІШЕННЯ 8

1.1 Аналіз задачі, засобів та методів її вирішення 8

1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи 9

Висновки до першого розділу 11

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 12

2.1 Проектування загального алгоритму роботи програми 12

2.2 Розробка функціональних алгоритмів роботи програми 13

2.3 Розробка програмного забезпечення 19

Висновки до другого розділу 20

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 23

3.1 Опис роботи з програмним додатком 23

3.2 Тестування роботи програмного забезпечення 29

Висновки до третього розділу 31

ВИСНОВКИ 32

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 33

ДОДАТКИ 35

# ВСТУП

**Актуальність теми:** пошук інформації є однією з основних складових людської діяльності, з ним ми зіштовхуємося щодня: вивчаючи театральну афішу, щоб вибрати цікавий спектакль, підбираючи в розкладі поїздів зручну електричку, перегортаючи телефонну книгу... Людині, яка в силу своєї професії чи захоплень часто зіштовхується з підбором і пошуком якої-небудь тематичної інформації, рано чи пізно (зі зростанням її обсягу) приходиться застосовувати деякі принципи систематизації і класифікації наявних даних, що забезпечують більш зручний і ефективний пошук. Так, у бібліотеках складають картотеку: відомості про книгу за визначеною схемою записуються на картку, туди ж міститься шифр - кілька букв і цифр, по яких можна визначити місце розташування книги (сховище, стелаж, полицю); картки розставляються в алфавітному чи тематичному порядку. Застосування інформаційно-пошукових систем дає більш широкі можливості для роботи з великими масивами інформації.

**Мета дослідження:** вдосконалення навичок програмування мовою С# та створення програмного забезпечення на тему “Інформаційно-пошукова система: сучасні моделі відеокарт”. Встановлена мета: повинна забезпечити, реалізувати наступні функції:

− Виведення даних на екран в зручному вигляді;

− Дані мають зберігатися у базі даних.

− Пошук даних (за всіма параметрами).

− Редагування даних. Перед редагуванням знайти потрібний запис.

− Видалення даних. Перед видаленням знайти потрібний запис.

− Фільтрація даних (за 3-а параметрами);

− Додавання нових записів.

− Порівняння даних. Перед порівнянням відмітити потрібні записи

− Програма має містити реєстрацію та авторизацію та здійснювати перехід між різними формами.

**Об’єкт дослідження:** технологія роботи з формами, класами, базою даних.

**Предмет дослідження:** робота з класами та наслідуванням, базою даних, методами класів, конструкторами класів, формами, функціями та бібліотеками класів.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДИ, ТА ЗАСОБИ ВИРІШЕННЯ

## Аналіз задачі, засобів та методів її вирішення

Поставлено завдання створити інформаційно-пошукову систему на тему “Сучасні модель відеокарт”. Для зберігання інформації будемо використовувати баз даних, а саме MySql для створення бази відеокарт та користувачів. Для виконання цього завдання програма має містити та вміти виконувати такі функції:

− Зчитувати інфомацію з бази даних;

Для реалізації буде створено метод, який прийматиме дані.

− Заносити зміни в базу даних;

Для реалізації під час роботи будемо заносити корективи в базу даних.

Програма має вміти виконувати такі функції з базою даних:

− Виведення даних;

Для реалізації будемо використовувати елемент Windows Forms DataGridView.

− Пошук даних;

Для реалізації буде створено метод, який шукатиме схожу інформацію в таблиці.

− Фільтрація даних;

Для реалізації буде створено метод, який фільтруватиме таблицю за параметрами, які визначить користувач.

−Порівняння даних;

Для реалізації буде створено метод, який відбиратиме обрані користувачем записи і показуватиме тільки їх.

− Редагування даних. Перед редагуванням знайти потрібний запис;

Для реалізації буде створено метод, який даватиме користувачеві редагувати конкретний запис.

− Видалення даних. Перед видаленням знайти потрібний запис;

Для реалізації буде створено метод, який видалятиме обраний користувачем запис.

− Додавання запису;

Для реалізації буде створено метод, який додаватиме новий запис користувача.

− Авторизування і реєстрація користувача;

Це потрібно для індентифікації користувача і надання йому можливостей: звичайний гість переглядати записи і доступ до пошуку адмін матиме змогу редагувати, видаляти та додавати записи, а зареєстрований юзер фільтрувати, бачити більш детальну інформацію та порівнювати записи.

− Зручне переміщення між формами;

Для реалізації буде створено взаємозв`язок між усіма формами.

## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи

Аналогічними системами пошуку користуються інтернет магазини. Такі інформаційно-пошукові системи зазвичай використовуються на сайтах і переглядаються користувачем з браузеру.

1) Один із найвідоміших прикладів є система пошуку маг інтернет-маркету з продажу відеокарт Rozetka, автором якої є Владислав Чечьоткін (<https://rozetka.com.ua/ua/>).

Вона написана за допомогою Owox, яку створила українська компанія. Ця система має доволі широкий спектр можливостей: фільтрування, пошук, показ більш детальної інформації про конкретний продукт, сортування і занесення продукту до корзини.

Перевагою цієї системи є кількість функціоналу, а мінусом робота через браузер.

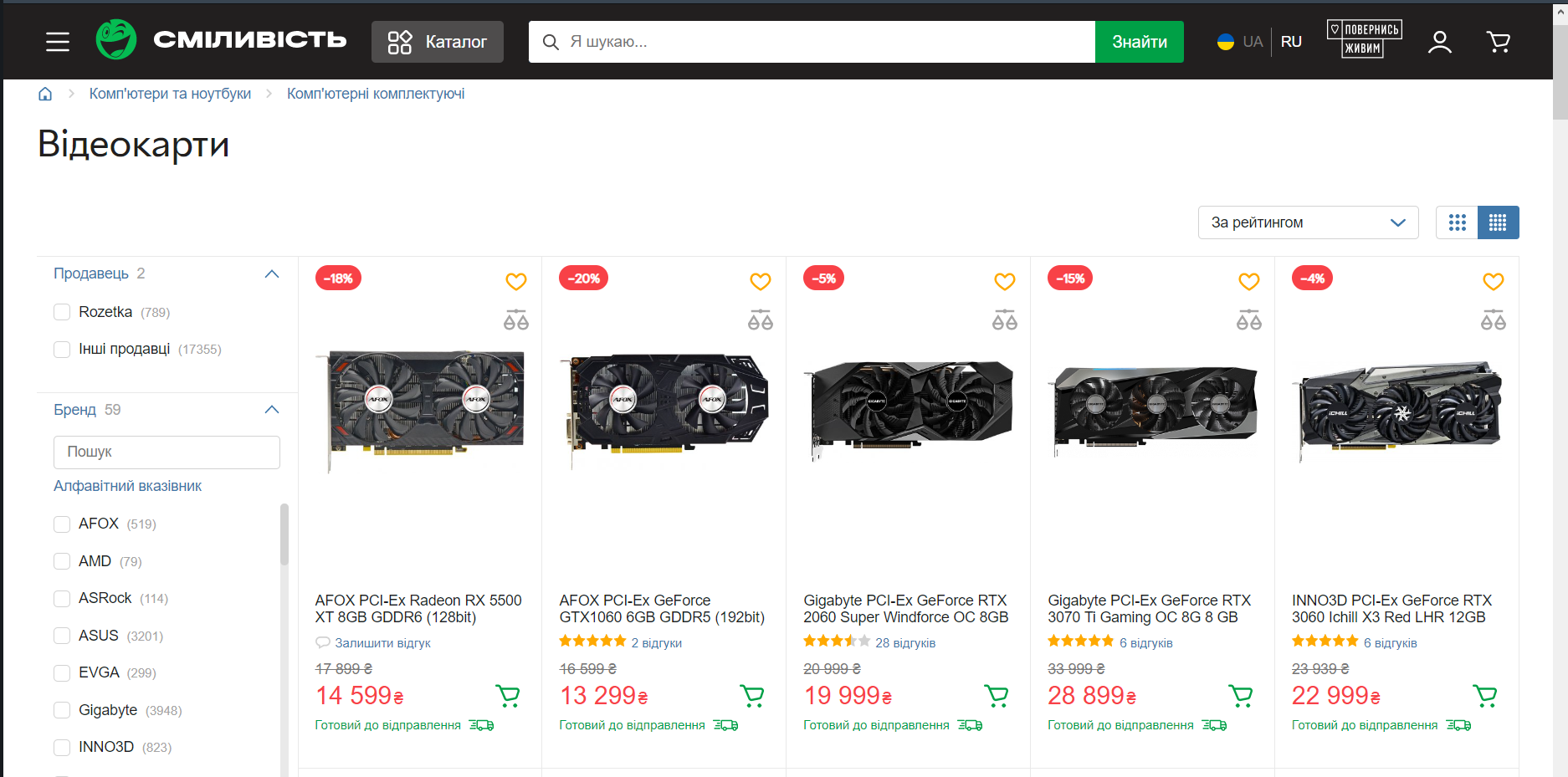


Рис. 1.1. Інтерфейс системи пошуку – “Rozetka”

2) Іншим аналогом можна вважати інтернет систему пошуку для купівлі відеокарт e.katalog, автором якого є Руслан Новіков (<https://ek.ua/ua/>).

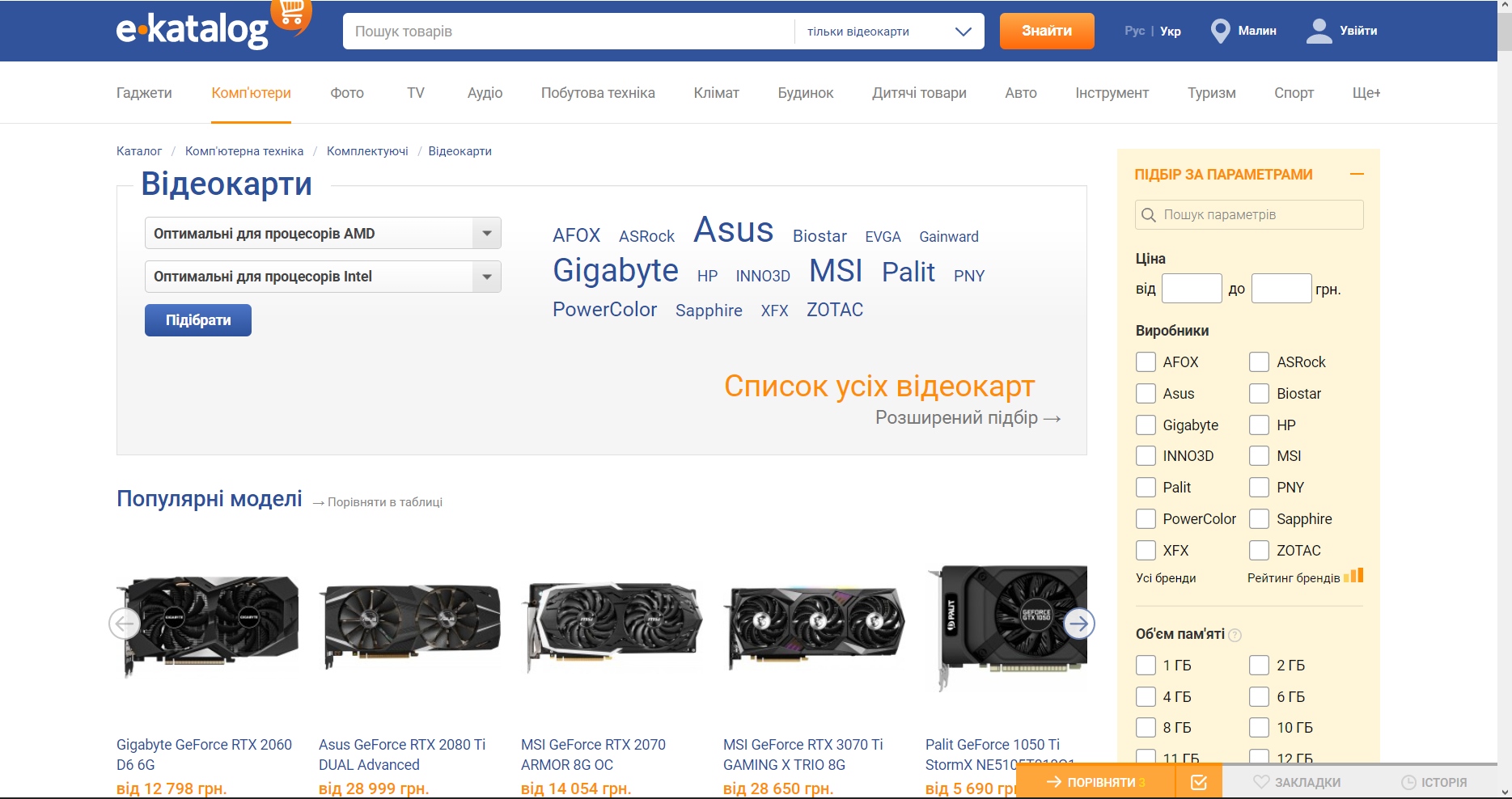


Рис. 1.2. Інтерфейс інформаційно-пошукової системи – “e.katalog”

Ця система пошуку містить фільтрацію, сортування, порівняння, пошук, реєстрацію та авторизацію, порівняння цін в інтернет магазинах, перегляд детальної інформації про конкретний товар.

Основною перевагою цієї програми є зв`язок з іншими інформаційно-пошуковими системами, що дозволяє в одному середовищі переглядати дані відразу з деяких. Недоліком цієї системи пошуку є робота в браузері та залежність від інтернету.

3) Третім аналогом візьмемо закордону систему пошуку GPUTracker, автором якої є Чарльз Кьюффер [(https://www.gputracker.eu/en/terms).](https://www.gputracker.eu/en/terms)

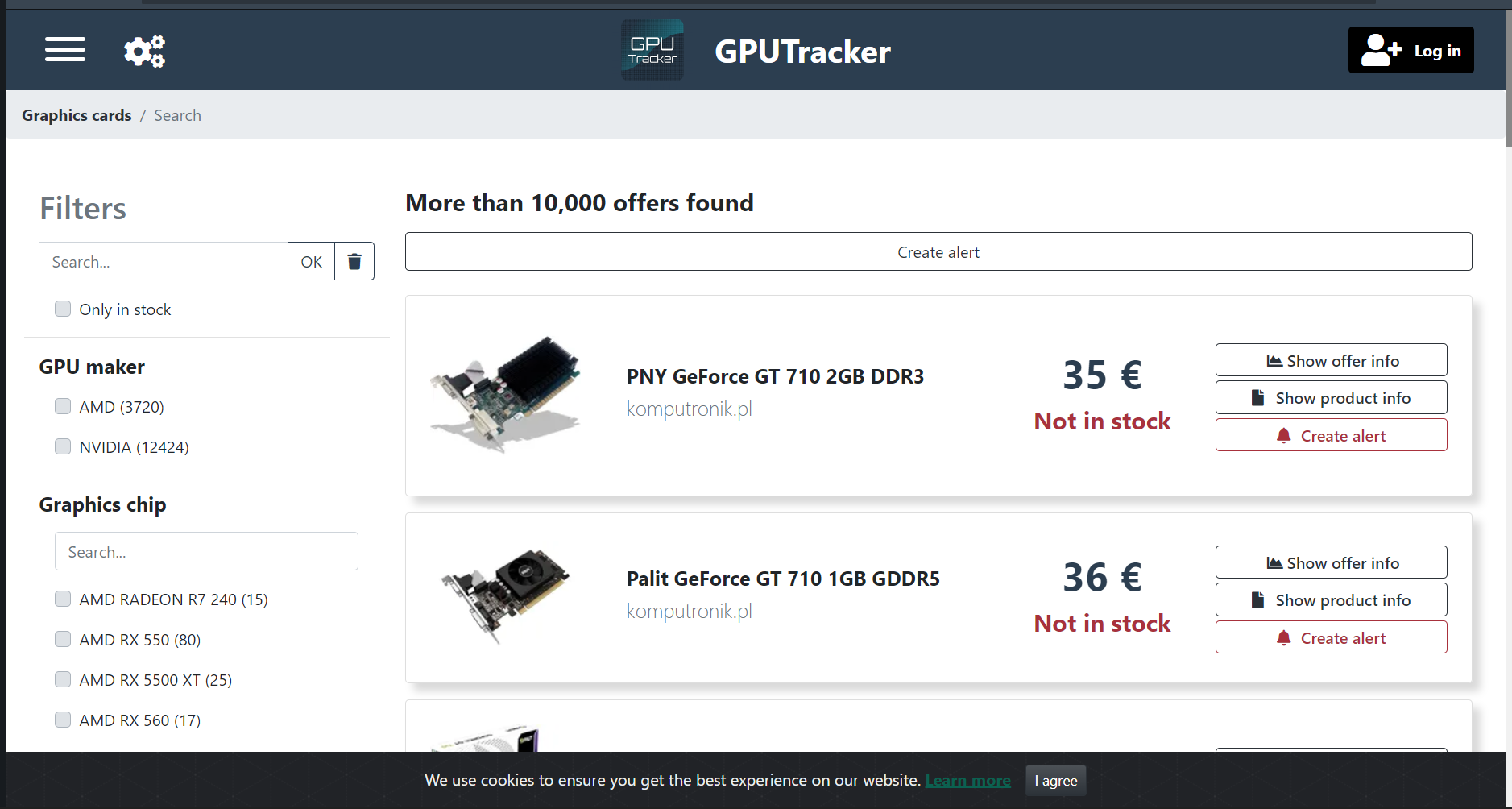


Рис. 1.3. Інтерфейс системи пошуку – “GPUTracker”

Система створена для знаходження найменшої ціни на відеокарту, тому містить відповідні функції й алгоритми: пошук, фільтрація, створення сповіщення про зниження ціни за відеокарту, реєстрація й авторизація та перегляд ціни в інтернет магазинах.

Перевагою є зв`язок з інтернет магазинами, а недоліками є відсутність української локалізації, відстуність сортування, мало інформації про відеокарту, присутня тільки про ціну та назва, і робота в браузері.

Отже, основним напрямом розробки програмного продукту є створення інформаційно-пошукової системи для роботи з базою даних відеокарт.

## Висновки до першого розділу

Таким чином, було визначено завдання і проведено його аналіз. Знайдено методи вирішення задачі та можливість їх реалізації. Проведено аналіз аналогів, знайдено їх недоліки та переваги, завдяки цьому визначено напрям розробки програмного продукту.

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Проектування загального алгоритму роботи програми

Відкривавши програму, користувача зустріне панель гостя з якої він зможе виконати потрібні йому дії (рис. 2.1).



Рис. 2.1. Загальна схема варіантів використання програми

На схемі зображено дії, які користувач може виконувати. Зустрічає його панель гостя, з якої він може перейти до реєстрації або авторизації. Якщо користувача буде авторизовано він перейде до панелі користувача або панелі адміна. У цих формах йому відкривається подальший та розширений функціонал. Адміністратор може редагувати, переглядати, шукати та створювати записи. А звичайний користувач порівнювати та фільтрувати записи, а також переглядати детальну інформацію щодо кожної відеокарти.

Для кращого розуміння переходу між функціоналом складемо детальну схему варіантів використання програми (рис. 2.2). На цій схемі детальніше розписано шляхи та умови переходу між формами.

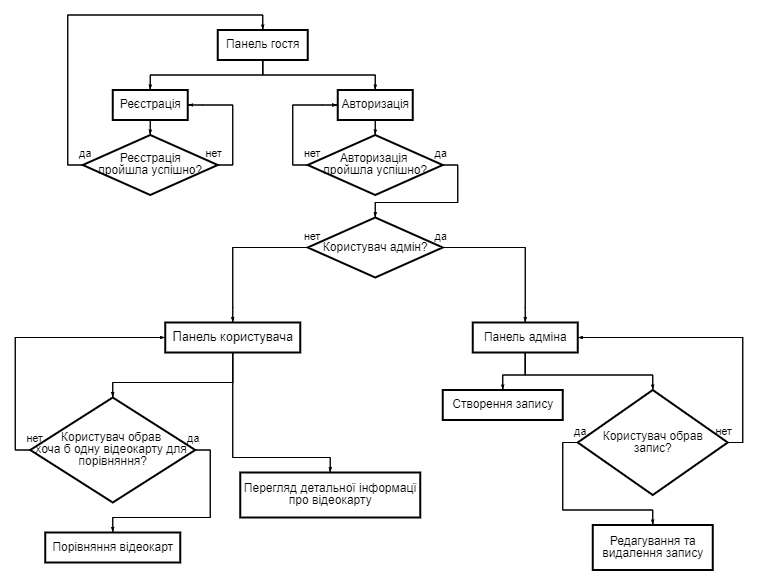


Рис. 2.2. Детальна схема варіантів використання програми

## 2.2 Розробка функціональних алгоритмів роботи програми

В першу чергу створимо діаграму класів для візуалізації взаємозв`язків класів у нашій програмі (рис. 2.3).

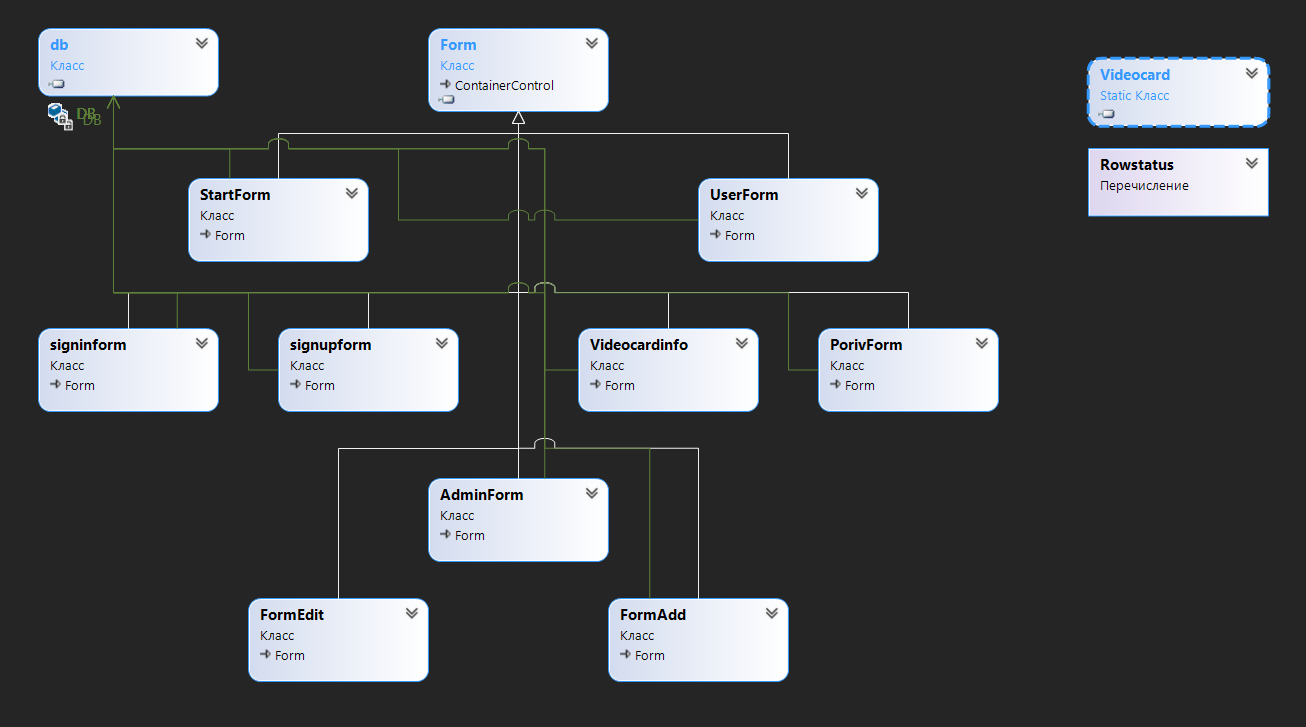


Рис. 2.3. Діаграма класів

На діаграмі класів ми бачимо, що всі форми успадковані від основої форми, ще бачимо що форми містять у собі поле класу db.cs. А також перехування Rowstatus та статичний клас Videocard.

Далі складемо блок-схеми методів програми:

1. Метод зчитування з бази даних (рис. 2.4). В ньому буде реалізовано створення команди Sql, для передачі інструкції базі даних. А також реалізовано команду зчитування з бази даних усіх записів, після чого буде викликаний метод для заповнення стовпців DataGridView.

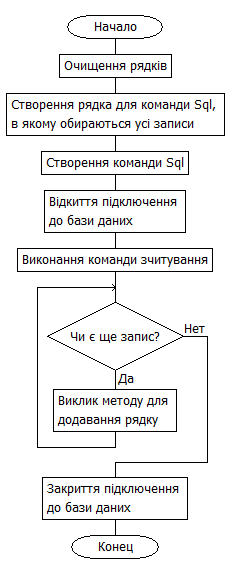


Рис. 2.4. Блок-схема для зчитування з бази даних

1. Метод пошуку (рис. 2.5). В ньому буде відбуватися вибірка записів з бази даних та подальший їх вивід.

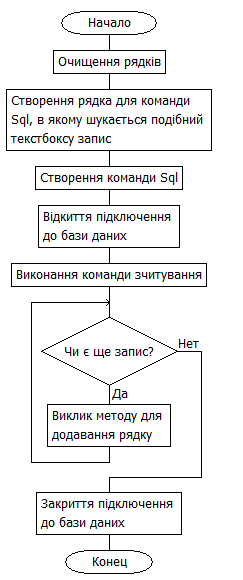


Рис. 2.5. Блок-схема методу пошуку

1. Метод оновлення DataGridView (рис. 2.6). В ньому буде відбуватися наново вивід записів з бази даних.

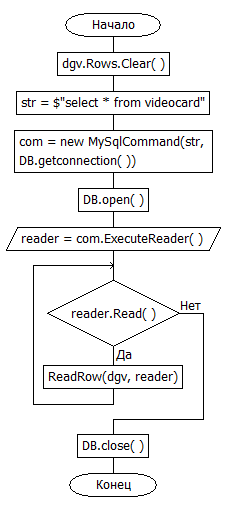


Рис. 2.6. Блок-схема методу оновлення DataGridView

1. Метод порівняння (рис. 2.7). В ньому буде відібрано записи, які обрав користувач та передано їх індекс у іншу форму для подальшого виведення.

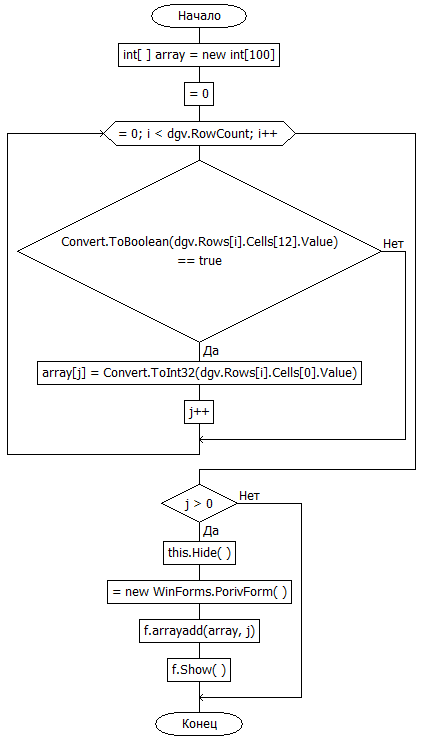


Рис. 2.7. Блок-схема методу порівняння

1. Метод видалення (рис. 2.8). В ньому буде знайдено індекс елемента, який потрібно видалити.

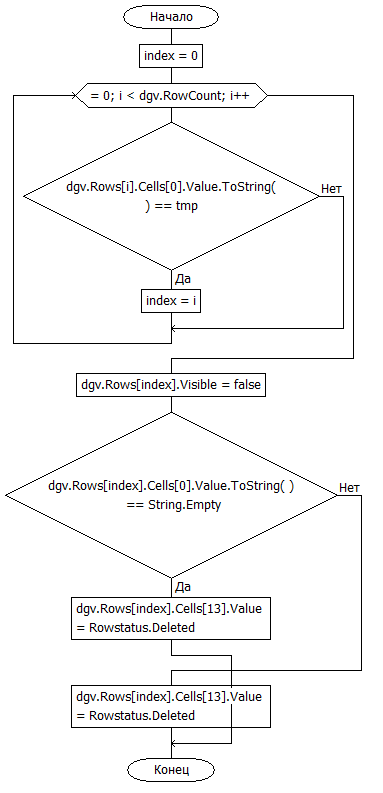


Рис. 2.8. Блок-схема методу видалення

1. Метод редагування (рис. 2.9). В ньому буде перезаписано усі поля запису.

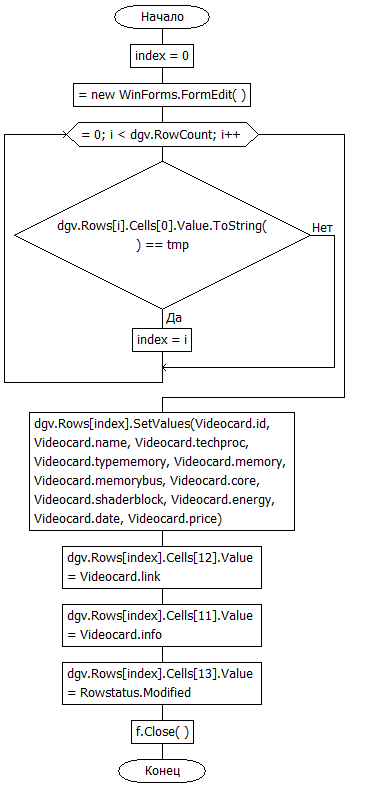


Рис. 2.9. Блок-схема методу редагування

1. Метод оновлення бази даних (рис. 2.10). В ньому буде передано базі даних зміни в записах.

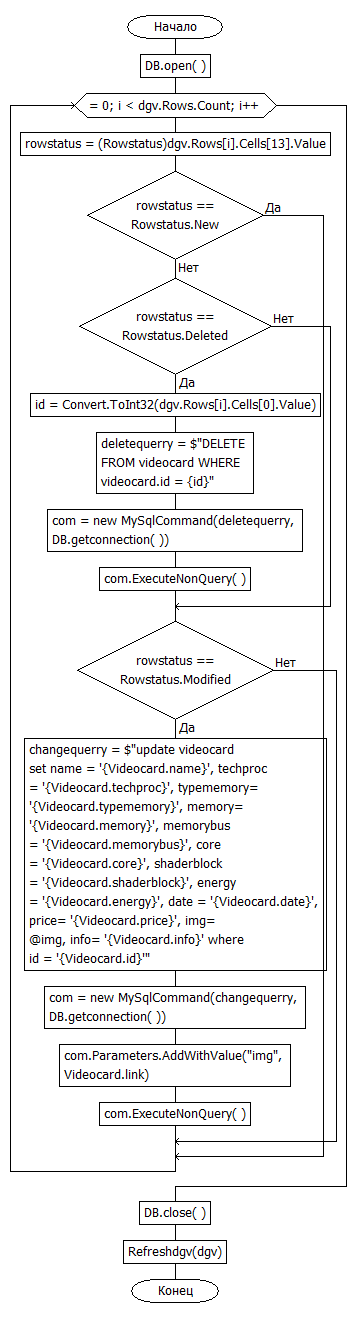


Рис. 2.10. Блок-схема оновлення бази даних

## 2.3 Розробка програмного забезпечення

Спочатку створимо клас, в якому будуть знаходитися методи для підключення до бази даних.

Лістинг:

public class db

{

MySqlConnection connection = new MySqlConnection("server = localhost; port=3306; username=root; password = root;database =database");

public void open()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public void close()

{

if (connection.State == System.Data.ConnectionState.Open)

connection.Close();

}

public MySqlConnection getconnection()

{

return connection;

}

}

Далі реалізуємо вивід даних у DataGridView.

1. Створимо стовпці:

Лістинг:

private void CreateColumns()

{

dgv.Columns.Add("id", "№");

dgv.Columns.Add("name", "Назва");

dgv.Columns.Add("techproc", "Техпроцес, нм");

dgv.Columns.Add("typememory", "Тип пам'яті");

dgv.Columns.Add("memory", "Об`єм пам`яті");

dgv.Columns.Add("memorybus", "Шина пам'яті");

dgv.Columns.Add("core", "Частота роботи ядра, МГц");

dgv.Columns.Add("shaderblock", "Шейдерних блоків");

dgv.Columns.Add("energy", "Споживання енергії, Вт");

dgv.Columns.Add("date", "Дата випуску");

dgv.Columns.Add("price", "Вартість, $");

dgv.Columns.Add("info", "info");

dgv.Columns.Add("img", "img");

dgv.Columns.Add("status", String.Empty);

}

1. Заповнимо їх:

Лістинг:

private void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), record.GetString(11), record.GetString(12), Rowstatus.New); ;

}

private void Refreshdgv(DataGridView dgv)

{

dgv.Rows.Clear();

string str = $"select \* from videocard";

MySqlCommand com = new MySqlCommand(str, DB.getconnection());

DB.open();

MySqlDataReader reader = com.ExecuteReader();

while (reader.Read())

ReadRow(dgv, reader);

DB.close();

}

Пошук в програмі буде відбуватися за допомогою оператора like в мові Sql.

Лістинг:

private void search(DataGridView dgv)

{

dgv.Rows.Clear();

string searchquerry = $"select \* from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%"+ textBoxsearch.Text +"%'";

MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

Дані для редагування та видалення будуть передавитися в форму FormAdd.cs за допомогою статичного класа Videocard.cs. Детальніше методи для реалізації цього будуть розписані в додатку А.

При виведені детальної інформації про відеокарту, користувач буде бачити фотографію, яка зберігається локально на жорстокому диску, а для редагування та додавання шляху написаний метод:

Лістинг:

private void buttonimg\_Click(object sender, EventArgs e)

{

ofd.Filter = "Image Files(\*.JPG;\*.PNG)|\*.JPG;\*.PNG|All files(\*.\*)|\*.\*";

if(ofd.ShowDialog() == DialogResult.OK)

{

try

{

Videocard.link = ofd.FileName;

}

catch

{

MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

Для обирання користувачем відеокарти, інформацію про яку потрібно показати детальніше, створенно подію CellContentDoubleClick.

Лістинг:

private void dgv\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

selectedrow = e.RowIndex;

int selectedcolumn = e.ColumnIndex;

if (selectedcolumn == 1)

{

if (e.RowIndex >= 0)

{

if (dgv.Rows[selectedrow].Cells[1].Value.ToString() != String.Empty)

{

Videocard.name = dgv.Rows[selectedrow].Cells[1].Value.ToString();

string querryimg = "select img from videocard where name like '%" + Videocard.name + "%'";

string querryinfo = "select info from videocard where name like '%" + Videocard.name + "%'";

DB.open();

MySqlCommand com = new MySqlCommand(querryimg, DB.getconnection());

Videocard.link = (string)com.ExecuteScalar();

com = new MySqlCommand(querryinfo, DB.getconnection());

Videocard.info = (string)com.ExecuteScalar();

DB.close();

this.Hide();

WinForms.Videocardinfo f = new WinForms.Videocardinfo();

f.Show();

}

}

}

}

## Висновки до другого розділу

Отже, створено загальний алгоритм роботи програми. Описано алгоритм роботи основних методів. Розроблено та наведено діаграму класів програми. Представлено та описано основні реалізовані методи.

# РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

## 3.1 Опис роботи з програмним додатком

Після запуску користувача зустрічатиме вікно “Панель гостя” на якій він бачить список записів, може знайти якийсь з них, та може зареєструватися або авторизуватися (рис. 3.1).

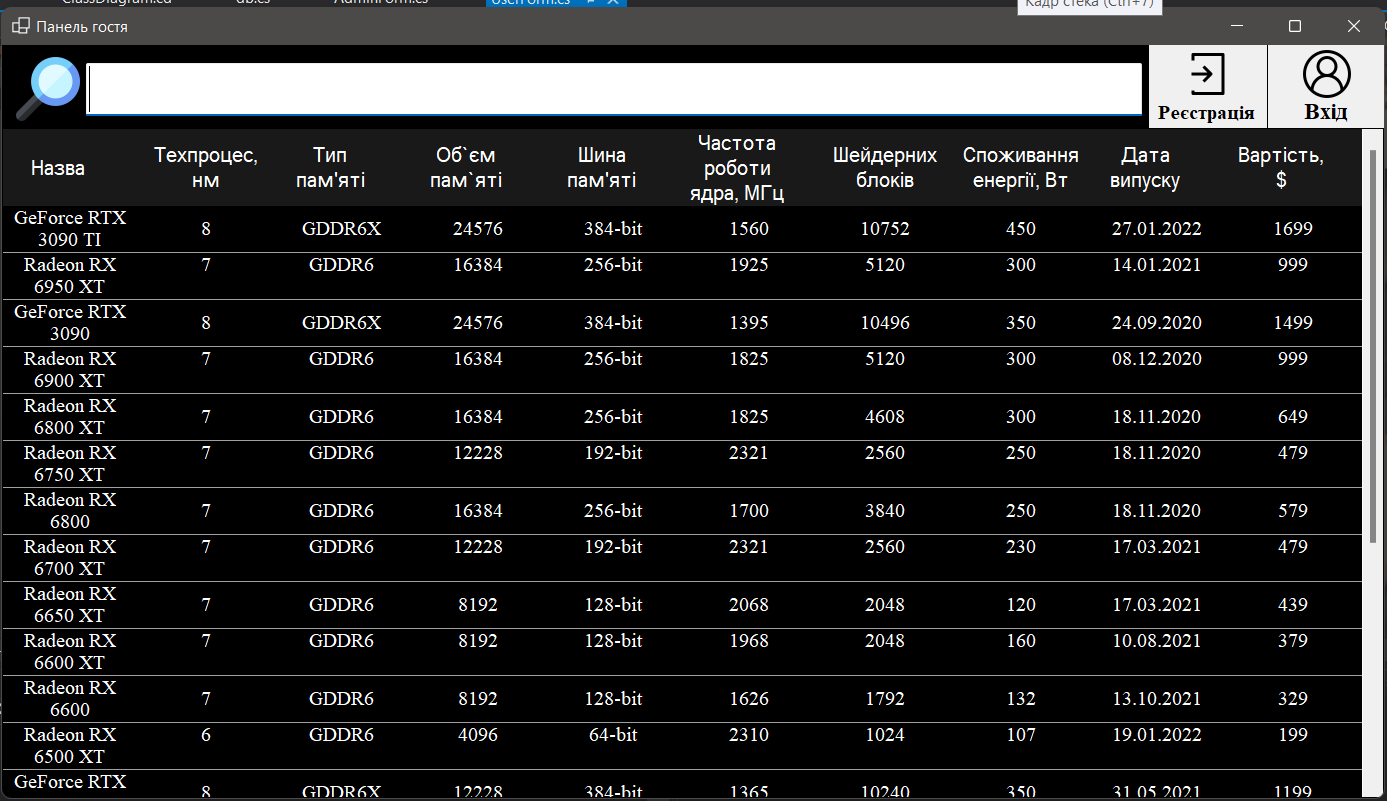


Рис. 3.1. Стартове меню програми

Якщо користувач вирішить знайти запис, то він може просто вписати потрібну йому інформацію у текстбокс, пошук відбувається по всім полям таблиці (рис. 3.2).

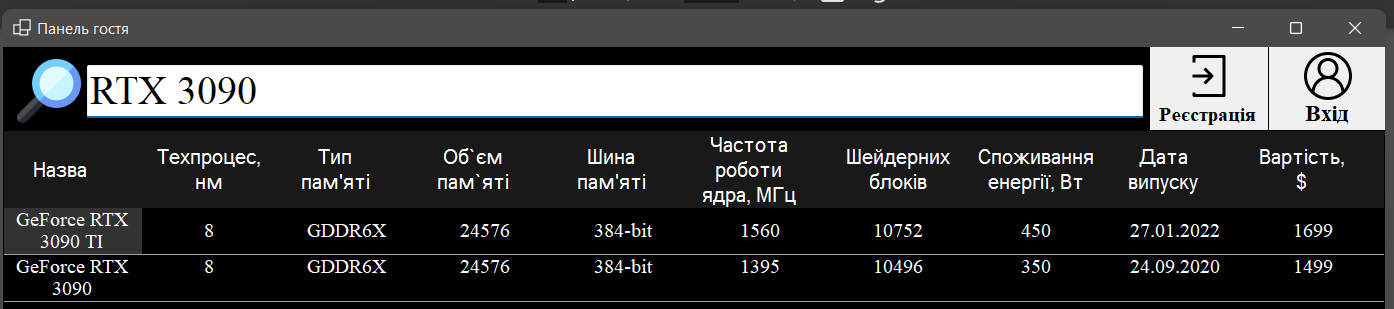


Рис. 3.2. Пошук запису в програмі

Для відкриття більшого функціоналу користувачу потрібно зареєструватися рисунок 3.3, або авторизуватися рисунок 3.4.

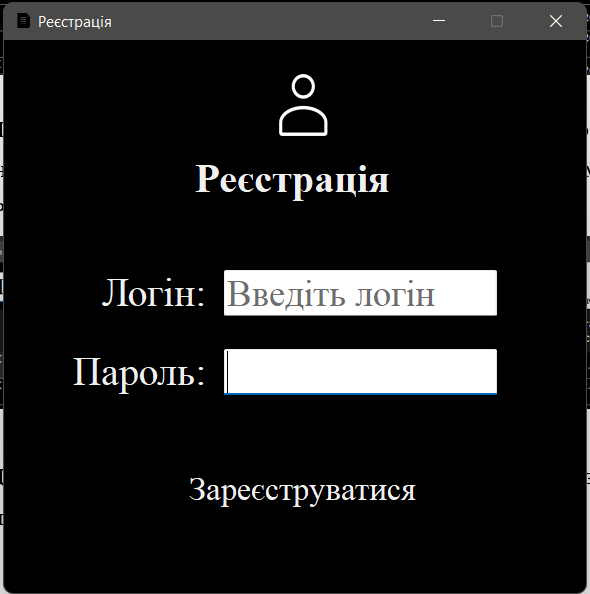


Рис. 3.3. Вікно реєстрації програми

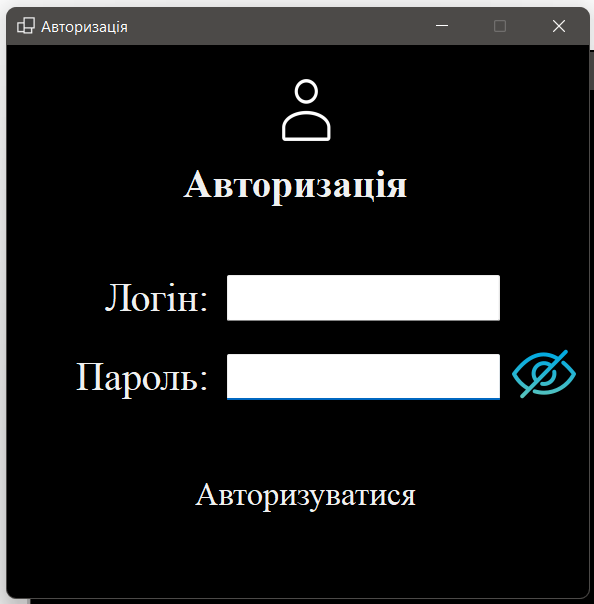


Рис. 3.4. Вікно авторизації програми

Після успішної авторизації звичайний користувач побачить “Панель користувача”, яку ми можемо побачити на рисунку 3.5.

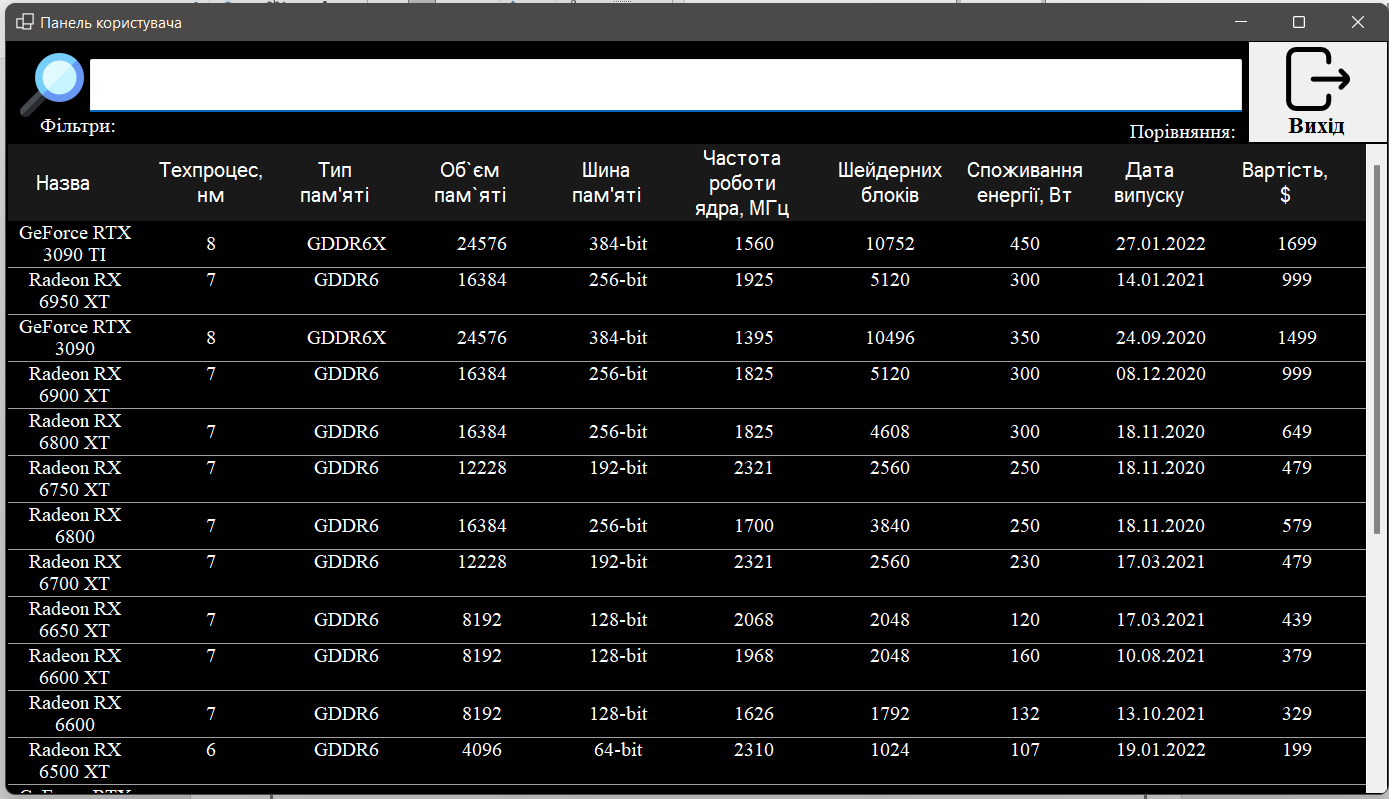


Рис. 3.5. Панель користувача програми

У авторизованого користувача відкривається додатковий функціонал: фільрування, порівняння та виведення детальної інформації.

Почнемо з фільтрування, для відкриття меню користувач має натиснути лівою клавішою миші на слово “Фільтри:”. Після цього відкриється додаткове меню, з яким уже можна фільтрувати записи (рис. 3.6).

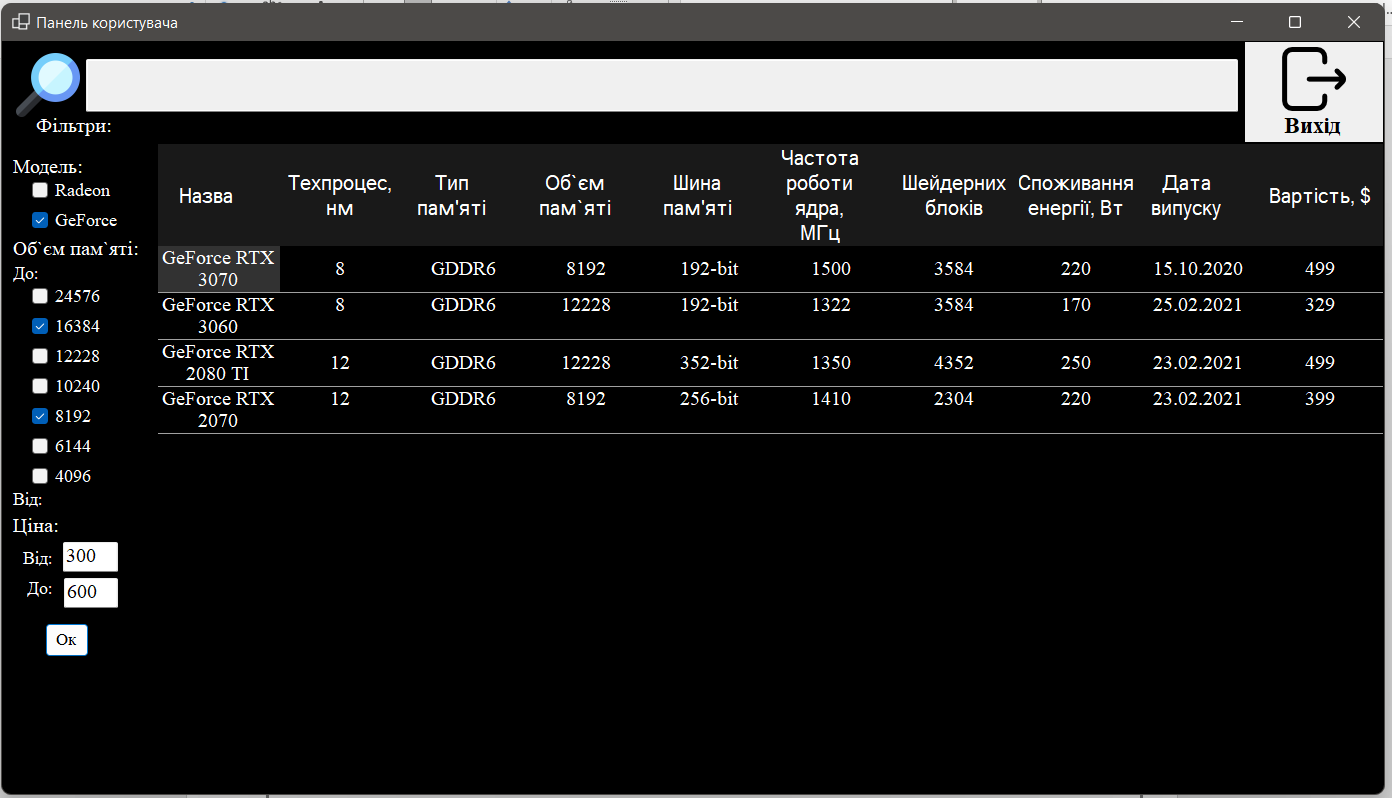


Рис. 3.7. Приклад фільтрації записів у програмі

Далі перейдемо до порівняння, для відкриття меню користувач має натиснути лівою клавішою миші на слово “Порівняння:”. Після цього можна буде обрати конкретні записи поставивши в колонці “Порівняння” галочку та вивести їх у окремому вікні, натиснувши кнопку “Порівняти”.

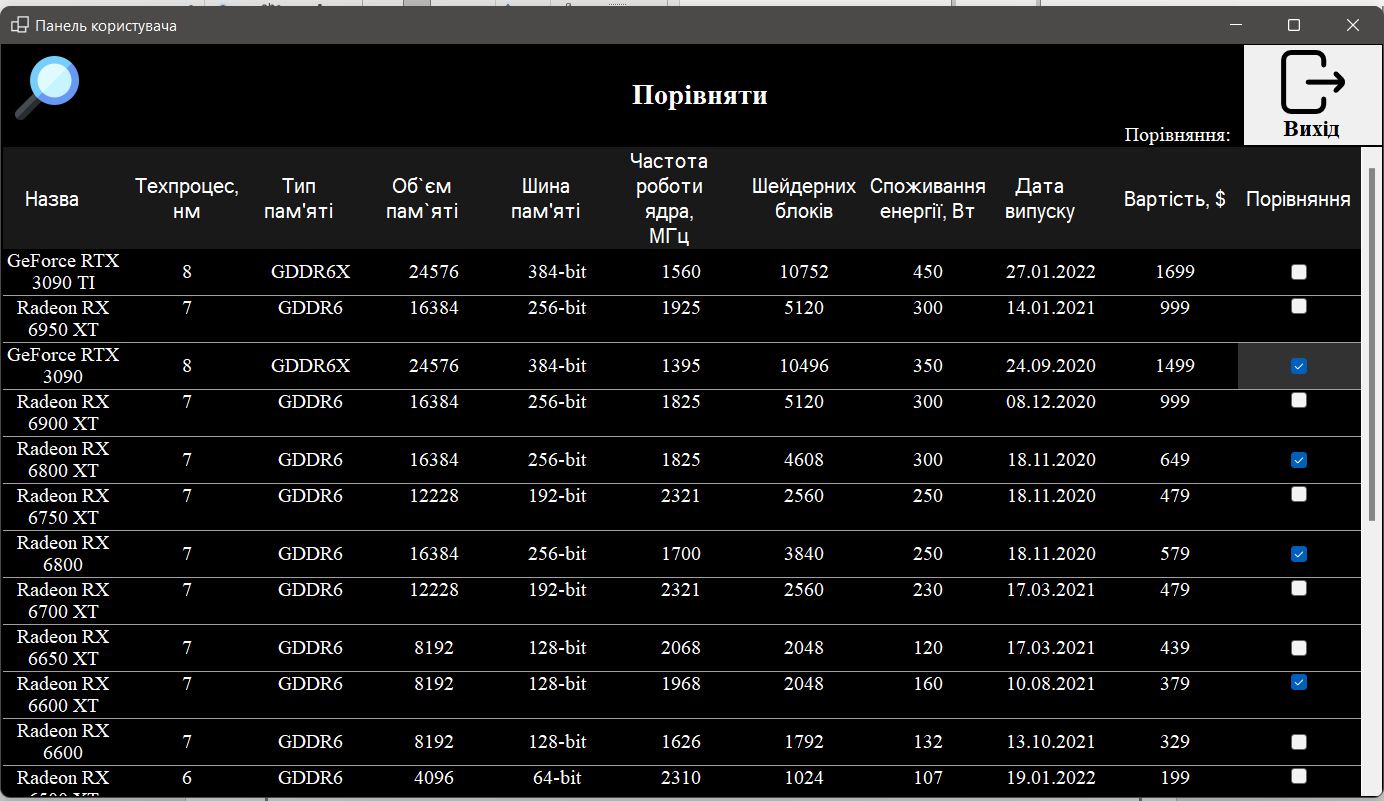


Рис. 3.8. Меню порівняння в програмі



Рис. 3.9. Приклад вікна порівняння

Щоб побачити більш детальну інформацію про відеокарту, користувачу потрібно двічі натиснути на назву відеокарти в таблиці та йому відкриється вікно (рис. 3.10).



Рис. 3.10. Вікно з детальною інформацією про відекарту

Якщо користувач є адміном, то він побачить вікно “Панель адміна” з іншим функціоналом (рис. 3.11).

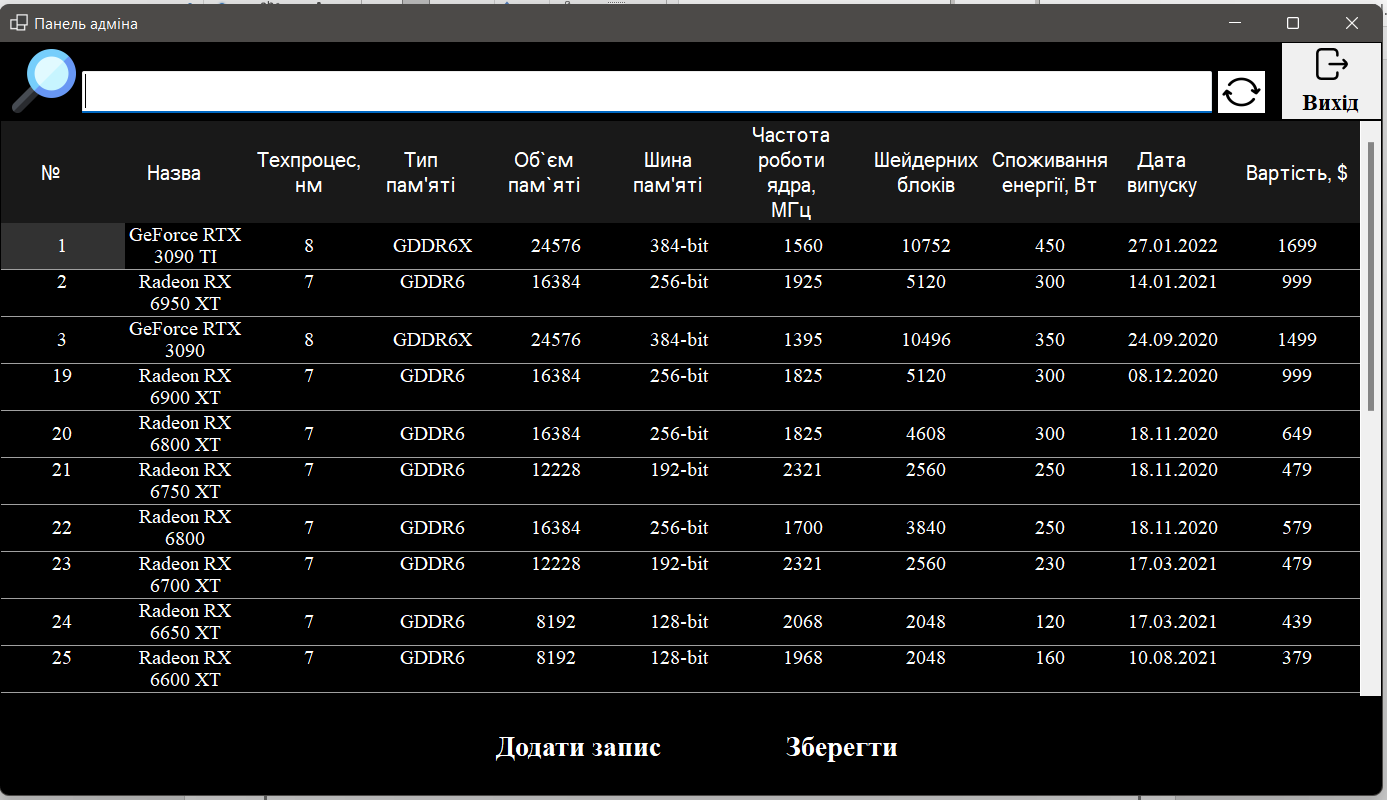


Рис. 3.11. Панель адміна в програмі

Ми бачимо додаткові кнопки “Додати запис” та “Зберегти”. Натиснувши кнопку “Додати запис”, користувач побачить вікно “Додавання запису” (рис. 3.12).

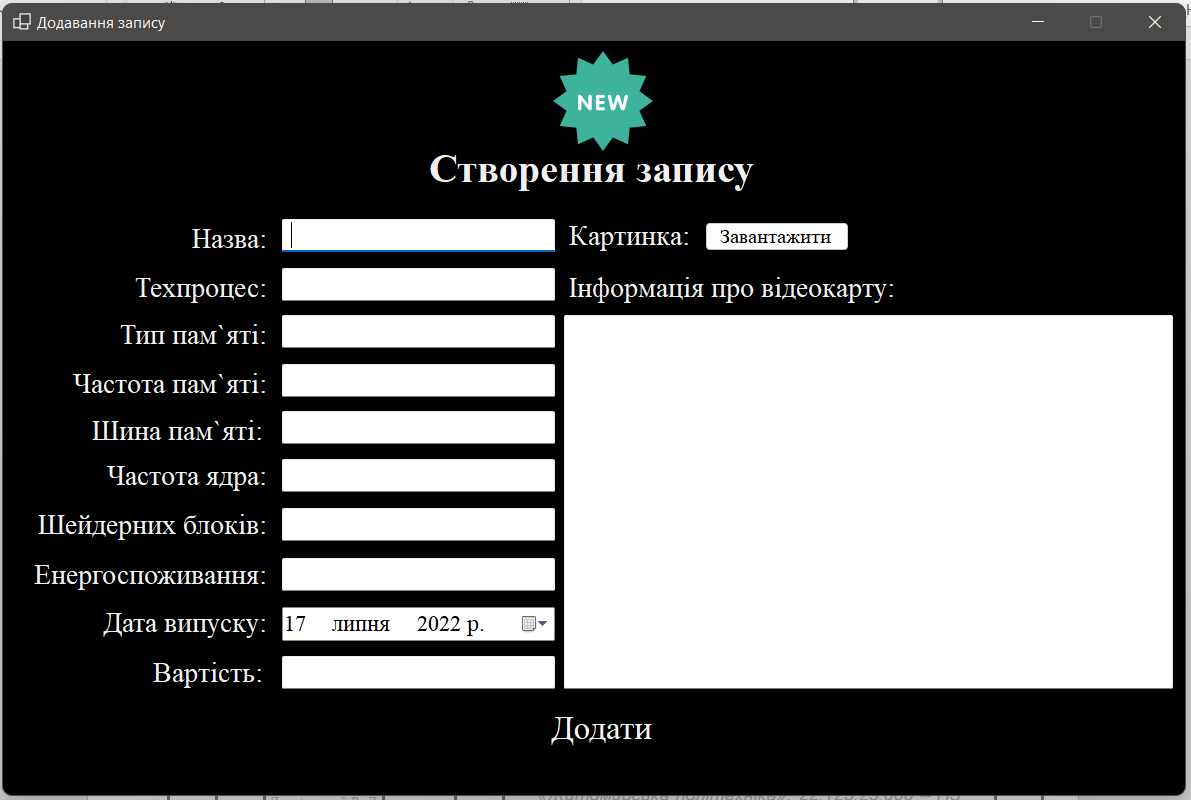


Рис. 3.12. Вікно додавання запису

Для редагування або видалення користувачу потрібно двічі клікнути на комірку таблиці з потрібним записом. Він побачить вікно, де зможе відредагувати або видалити запис (рис. 3.13).

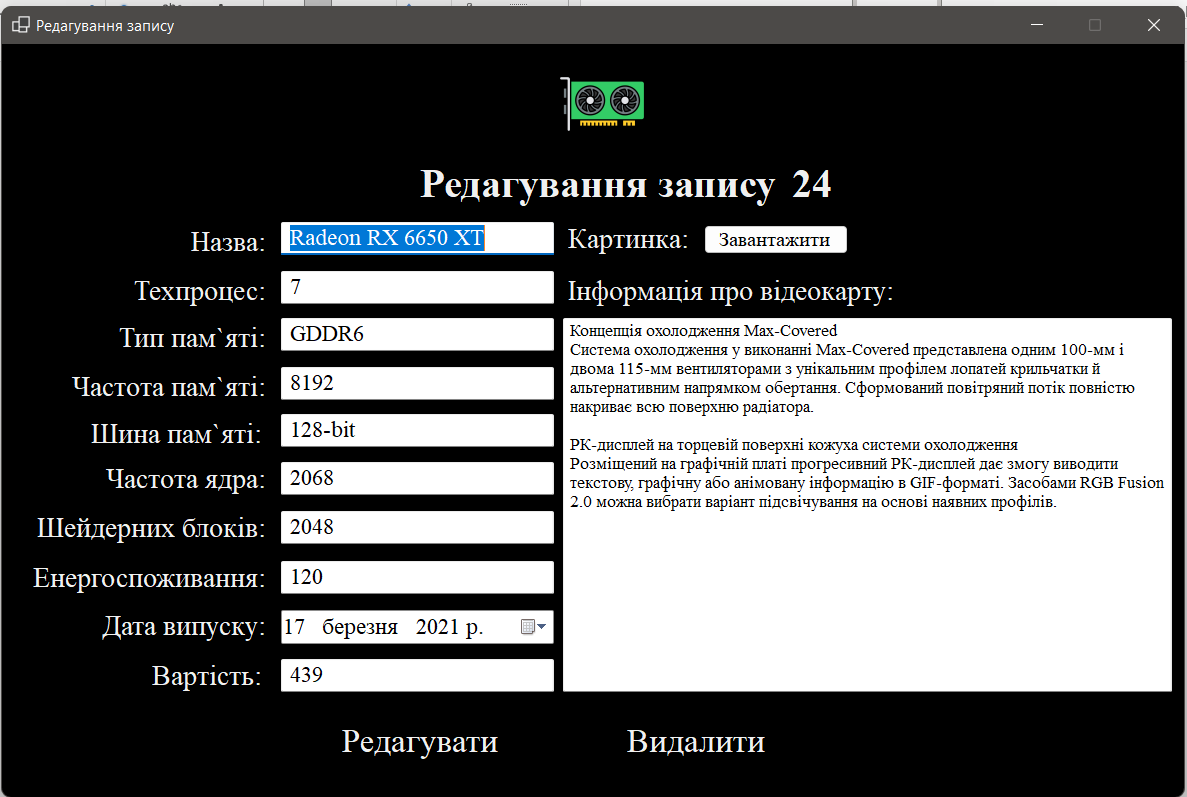


Рис. 3.13. Вікно редагування запису

## 3.2 Тестування роботи програмного забезпечення

Програма передбачає помилки при введені користувачем даних для запобіження помилок. Наприклад не можна при реєстрації вказати логін, який уже є в базі даних (рис. 3.14).

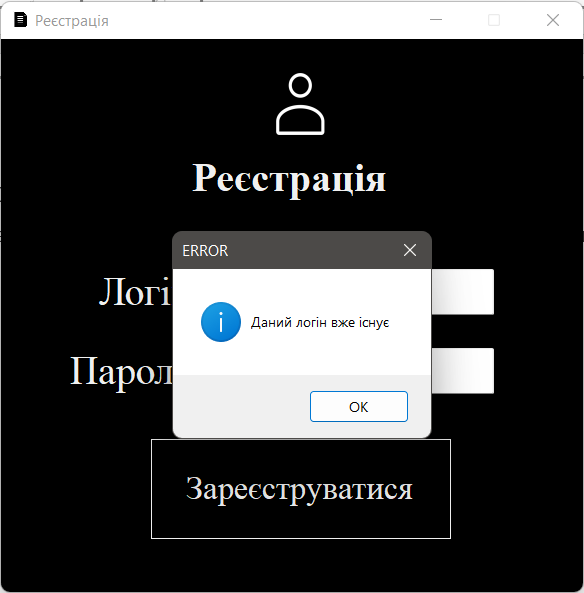


Рис. 3.14 Помилка “Даний логін вже існує”

Якщо користувач вирішить при редагуванні або додавані запису вписати текст у поле для числа, він також зустріне помилку рисунок 3.15.

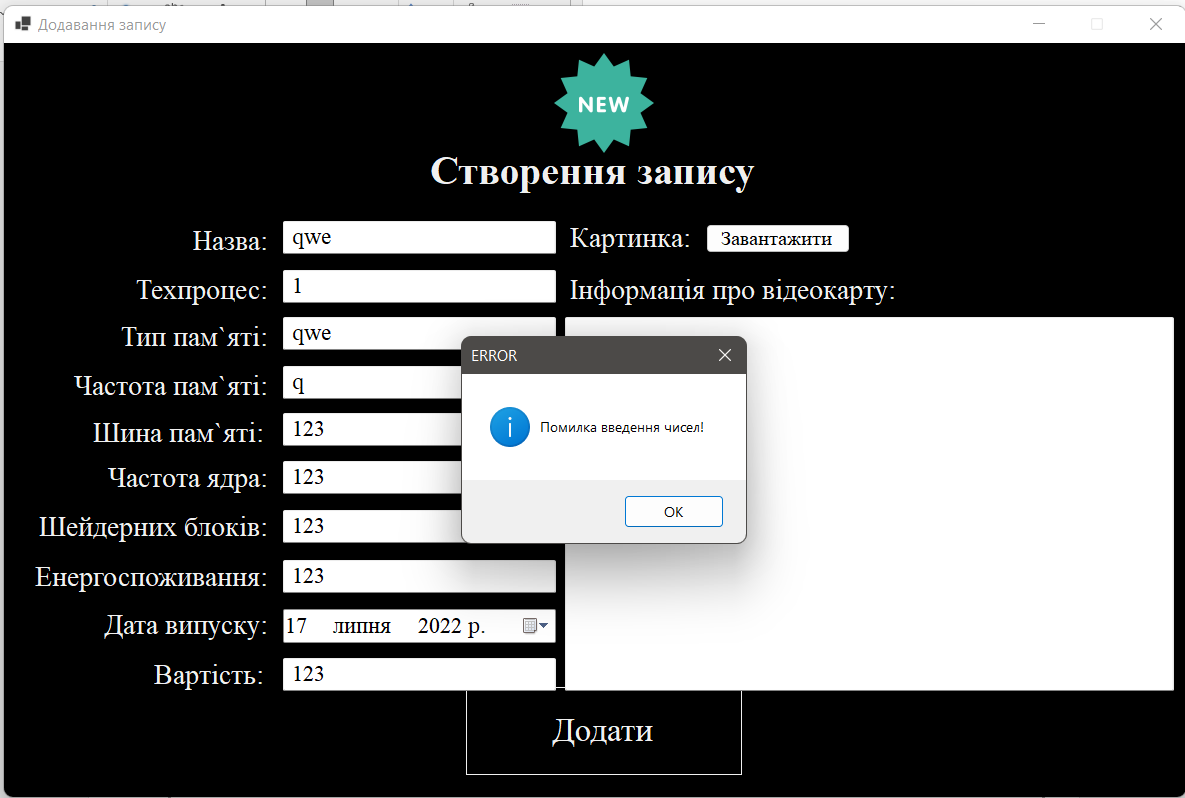


Рис. 3.15. Помилка “Помилка введеня чисел!”

В програмі передбачено неможливість під час фільтрації вписати вартість “від” більше “до” (рис. 3.16)

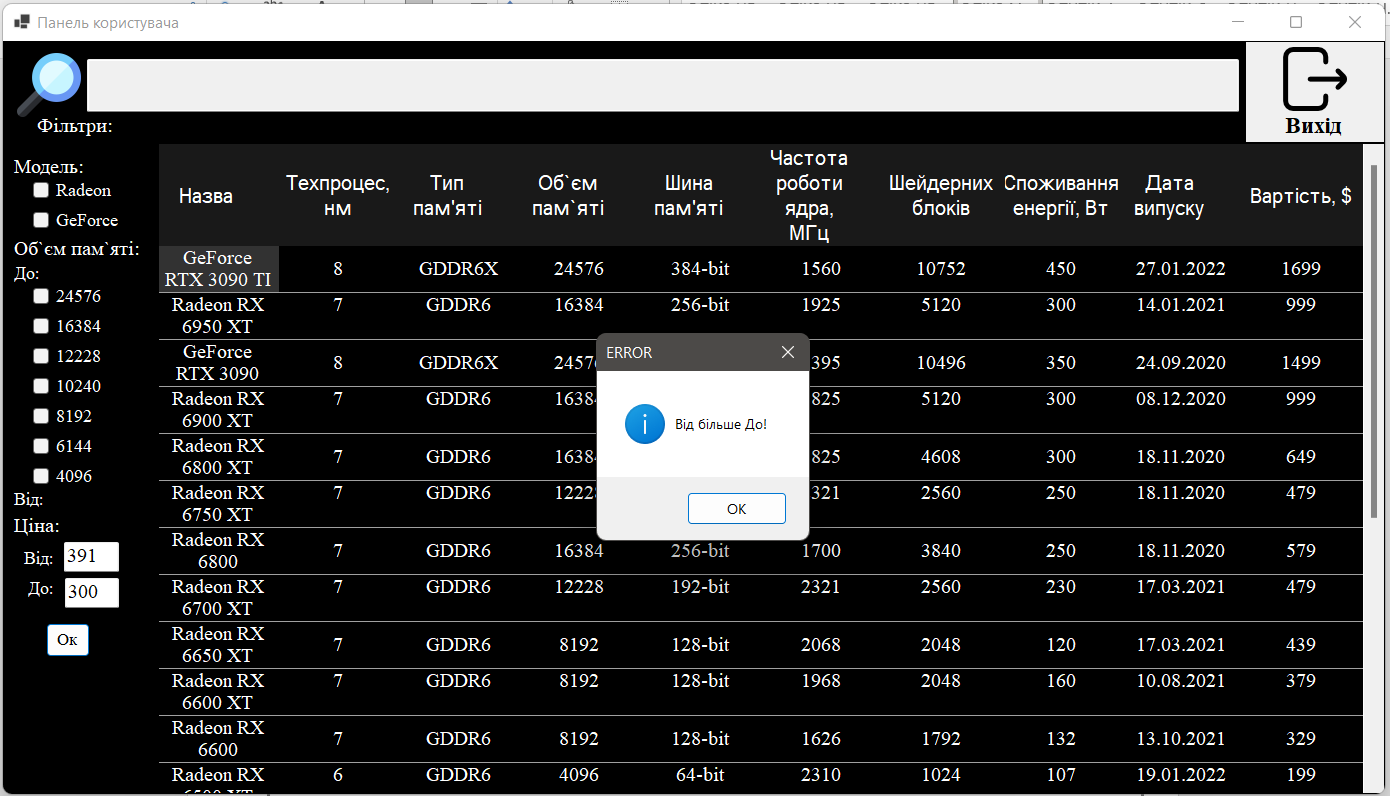


Рис. 3.16. Помилка “Від більше До!”

При введені неіснуючого логіну або неправильного паролю під час авторизації, користувача про це сповістить програма (рис. 3.17).

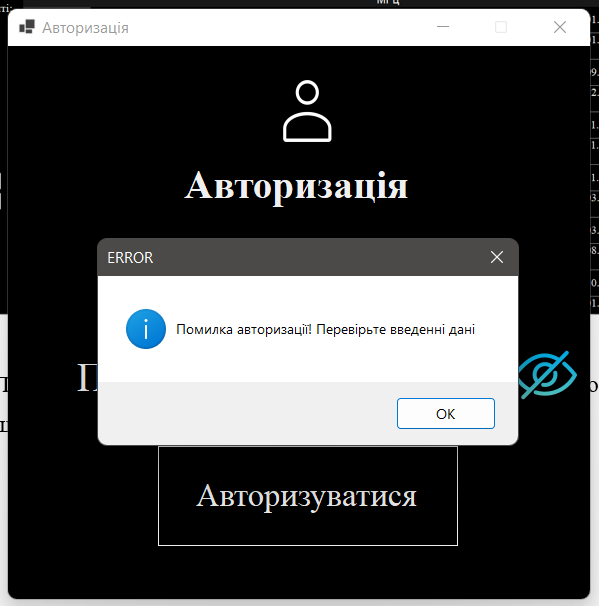


Рис. 3.17. Помилка “Помилка авторизації! Перевірьте введені дані”

## Висновки до третього розділу

Отже, в межах даного розділу було описано повністю роботу з інтерфейсом та програмою від моменту запуску до завершення роботи. Також було проаналізовано та протестовано помилки, які можуть виникнути під час роботи програми.

# ВИСНОВКИ

Отже, під час виконання курсової роботи було вдосконалено навички програмування мовою С#. Проведено роботу з формами, класами та базою даних та поглиблено знання з ООП. Було створено програмне забезпечення для програми на тему “Інформаційно-пошукова система: сучасні моделі відеокарт”.

У програмі було реалізовано увесь потрібний та запланований функціонал:

− Виведення даних на екран в зручному вигляді;

− Дані зберігаються у базі даних.

− Пошук даних (за всіма параметрами).

− Редагування даних. Перед редагуванням можна знайти потрібний запис.

− Видалення даних. Перед видаленням можна знайти потрібний запис.

− Фільтрація даних (за 3-а параметрами);

− Додавання нових записів.

− Порівняння даних. Перед порівнянням відмічаються потрібні записи

− Програма містить реєстрацію та авторизацію та здійснює перехід між різними формами.

Таким чином, програма відповідає усім поставленим вимогам та задачам. В подальшому може бути покращенна шляхом збільшення та доповнення функціоналу.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Эндрю Т. Язык Программирования C# 5.0 и платформа .NET 4.5 / Троелсен Эндрю., 2013. – 1168 с. – (6-е издание).

2. Аdmin. MS Access и C# [Електронний ресурс] / admin. – 2018. – Режим доступу до ресурсу: <https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-sbd.html>.

3. Admin. Добавление форм. Взаимодействие между формами [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/2.3.php>.

4. Admin. Создание меню MenuStrip [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/5.2.php>.

5. Admin. Контекстное меню ContextMenuStrip [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/windowsforms/5.4.php>.

6. Admin. Все операции с БД в графическом приложении [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/adonet/3.5.php>.

7. Admin. Операции со строками [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/7.2.php>.

8. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Джеффри Рихтер. – Питер, 2013. – 896 с. – (4-е изд).

9. Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.

10. Фленов М. Библия C# / Михаил Фленов. – Питер: БХВ-Петербург, 2011. – 541 с. – (2-е изд).

11. Warren G. Подключение к данным в базе данных Access (Windows Forms) [Електронний ресурс] / Genevieve Warren. – 2019. – Режим доступу до ре сурсу: <https://docs.microsoft.com/ru-ru/visualstudio/data-tools/connect-to-data-in-anaccess-database-windows-forms?view=vs-2019>.

12. Admin. Отношения между классами и объектами [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/patterns/1.2.php.

# ДОДАТКИ

Додаток А

Бібліотека класів lib

Лістинг db.cs:

using MySql.Data.MySqlClient;

using System;

namespace lib

{

public class db

{

MySqlConnection connection = new MySqlConnection("server = localhost; port=3306; username=root; password = root;database =database");

public void open()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public void close()

{

if (connection.State == System.Data.ConnectionState.Open)

connection.Close();

}

public MySqlConnection getconnection()

{

return connection;

}

}

}

Лістинг Videocard.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lib

{

public static class Videocard

{

public static int id;

public static string name;

public static int techproc;

public static string typememory;

public static int memory;

public static string memorybus;

public static int core;

public static int shaderblock;

public static int energy;

public static string date;

public static int price;

public static string info;

public static string link;

}

}

Додаток Б

Проект WinForms

Лістинг AdminForm.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace MainForm

{

enum Rowstatus{

Modified,

New,

Deleted

}

public partial class AdminForm : Form

{

db DB = new db();

int selectedRow;

public string tmp;

public AdminForm()

{

InitializeComponent();

}

private void CreateColumns()

{

dgv.Columns.Add("id", "№");

dgv.Columns.Add("name", "Назва");

dgv.Columns.Add("techproc", "Техпроцес, нм");

dgv.Columns.Add("typememory", "Тип пам'яті");

dgv.Columns.Add("memory", "Об`єм пам`яті");

dgv.Columns.Add("memorybus", "Шина пам'яті");

dgv.Columns.Add("core", "Частота роботи ядра, МГц");

dgv.Columns.Add("shaderblock", "Шейдерних блоків");

dgv.Columns.Add("energy", "Споживання енергії, Вт");

dgv.Columns.Add("date", "Дата випуску");

dgv.Columns.Add("price", "Вартість, $");

dgv.Columns.Add("info", "info");

dgv.Columns.Add("img", "img");

dgv.Columns.Add("status", String.Empty);

}

private void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), record.GetString(11), record.GetString(12), Rowstatus.New); ;

}

private void Refreshdgv(DataGridView dgv)

{

dgv.Rows.Clear();

string str = $"select \* from videocard";

MySqlCommand com = new MySqlCommand(str, DB.getconnection());

DB.open();

MySqlDataReader reader = com.ExecuteReader();

while (reader.Read())

ReadRow(dgv, reader);

DB.close();

}

private void Form1\_Load(object sender, EventArgs e)

{

CreateColumns();

Refreshdgv(dgv);

styledatagridview();

dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);

dgv.Columns["status"].Visible = false;

dgv.Columns["img"].Visible = false;

dgv.Columns["info"].Visible = false;

}

private void styledatagridview()

{

dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;

dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;

dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);

dgv.DefaultCellStyle.SelectionForeColor = Color.White;

dgv.BackgroundColor = Color.Black;

dgv.EnableHeadersVisualStyles = false;

dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;

dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);

dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25,25,25);

dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;

}

public void change()

{

int index = 0;

WinForms.FormEdit f = new WinForms.FormEdit();

for (int i = 0; i < dgv.RowCount; i++)

{

if (dgv.Rows[i].Cells[0].Value.ToString() == tmp)

{

index = i;

}

}

dgv.Rows[index].SetValues(Videocard.id, Videocard.name, Videocard.techproc, Videocard.typememory, Videocard.memory, Videocard.memorybus, Videocard.core, Videocard.shaderblock, Videocard.energy, Videocard.date, Videocard.price);

dgv.Rows[index].Cells[12].Value = Videocard.link;

dgv.Rows[index].Cells[11].Value = Videocard.info;

dgv.Rows[index].Cells[13].Value = Rowstatus.Modified;

f.Close();

}

private void button\_add\_Click(object sender, EventArgs e)

{

this.Hide();

WinForms.FormAdd f = new WinForms.FormAdd();

f.StartPosition = FormStartPosition.CenterScreen;

f.Show();

}

private void textBoxsearch\_TextChanged(object sender, EventArgs e)

{

search(dgv);

}

private void search(DataGridView dgv)

{

dgv.Rows.Clear();

string searchquerry = $"select \* from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%"+ textBoxsearch.Text +"%'";

MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

public void delete()

{

int index = 0;

for (int i = 0; i < dgv.RowCount; i++)

{

if (dgv.Rows[i].Cells[0].Value.ToString() == tmp)

{

index = i;

}

}

dgv.Rows[index].Visible = false;

if (dgv.Rows[index].Cells[0].Value.ToString() == String.Empty)

{

dgv.Rows[index].Cells[13].Value = Rowstatus.Deleted;

return;

}

dgv.Rows[index].Cells[13].Value = Rowstatus.Deleted;

}

private void update()

{

DB.open();

for (int i = 0; i < dgv.Rows.Count; i++)

{

var rowstatus = (Rowstatus)dgv.Rows[i].Cells[13].Value;

if (rowstatus == Rowstatus.New)

continue;

if (rowstatus == Rowstatus.Deleted)

{

var id = Convert.ToInt32(dgv.Rows[i].Cells[0].Value);

var deletequerry = $"DELETE FROM videocard WHERE videocard.id = {id}";

MySqlCommand com = new MySqlCommand(deletequerry, DB.getconnection());

com.ExecuteNonQuery();

}

if (rowstatus == Rowstatus.Modified)

{

string changequerry = $"update videocard set name = '{Videocard.name}', techproc = '{Videocard.techproc}', typememory= '{Videocard.typememory}', memory= '{Videocard.memory}', memorybus = '{Videocard.memorybus}', core = '{Videocard.core}', shaderblock = '{Videocard.shaderblock}', energy = '{Videocard.energy}', date = '{Videocard.date}', price= '{Videocard.price}', img= @img, info= '{Videocard.info}' where id = '{Videocard.id}'";

MySqlCommand com = new MySqlCommand(changequerry, DB.getconnection());

com.Parameters.AddWithValue("img", Videocard.link);

com.ExecuteNonQuery();

}

}

DB.close();

Refreshdgv(dgv);

}

private void button\_save\_Click(object sender, EventArgs e)

{

update();

}

private void dgv\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

selectedRow = e.RowIndex;

if (e.RowIndex >= 0)

{

DataGridViewRow row = dgv.Rows[selectedRow];

Videocard.name = row.Cells[1].Value.ToString();

Videocard.techproc = Convert.ToInt32(row.Cells[2].Value);

Videocard.typememory = row.Cells[3].Value.ToString();

Videocard.memory = Convert.ToInt32(row.Cells[4].Value);

Videocard.memorybus = row.Cells[5].Value.ToString();

Videocard.core = Convert.ToInt32(row.Cells[6].Value);

Videocard.shaderblock = Convert.ToInt32(row.Cells[7].Value);

Videocard.energy = Convert.ToInt32(row.Cells[8].Value);

Videocard.date = row.Cells[9].Value.ToString();

Videocard.price = Convert.ToInt32(row.Cells[10].Value);

Videocard.id = Convert.ToInt32(row.Cells[0].Value);

Videocard.link = row.Cells[12].Value.ToString();

Videocard.info = row.Cells[11].Value.ToString();

this.Hide();

WinForms.FormEdit f = new WinForms.FormEdit();

f.StartPosition = FormStartPosition.CenterScreen;

f.Show();

}

}

private void pictureBoxrefresh\_Click(object sender, EventArgs e)

{

Refreshdgv(dgv);

}

private void buttonexit\_Click(object sender, EventArgs e)

{

DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult1 == DialogResult.Yes)

{

DialogResult dialogResult2 = MessageBox.Show("Ви точно бажаєте покинути ваш чудовий акаунт?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult2 == DialogResult.Yes)

{

MainForm.StartForm f = new MainForm.StartForm();

this.Hide();

f.Show();

}

}

}

private void AdminForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult1 == DialogResult.Yes)

{

Application.Exit();

}

else

{

e.Cancel = true;

}

}

}

}

Лістинг FormAdd.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace WinForms

{

public partial class FormAdd : Form

{

db DB = new db();

public FormAdd()

{

InitializeComponent();

}

OpenFileDialog ofd = new OpenFileDialog();

private void buttonedit\_Click(object sender, EventArgs e)

{

DB.open();

Videocard.name = textBoxname.Text;

int techproc = Videocard.techproc;

Videocard.typememory = textBoxmemorytype.Text;

int memory = Videocard.memory;

Videocard.memorybus = textBoxmemorybus.Text;

int core = Videocard.core;

int shaderblock = Videocard.shaderblock;

int energy = Videocard.energy;

Videocard.date = dateTimePicker1.Value.ToString("s");

int price = Videocard.price;

Videocard.info = textBoxinfo.Text;

Videocard.link = ofd.FileName;

if (int.TryParse(textBoxtechproc.Text, out techproc) && int.TryParse(textBoxmemory.Text, out memory) && int.TryParse(textBoxcore.Text, out core) && int.TryParse(textBoxshaderblock.Text, out shaderblock) && int.TryParse(textBoxenergy.Text, out energy) && int.TryParse(textBoxprice.Text, out price))

{

Videocard.techproc = techproc;

Videocard.memory = memory;

Videocard.core = core;

Videocard.shaderblock = shaderblock;

Videocard.energy = energy;

Videocard.price = price;

var addQuerry = $"insert into videocard ( , name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price, info, img) values (NULL,'{Videocard.name}','{Videocard.techproc}','{Videocard.typememory}','{Videocard.memory}','{Videocard.memorybus}','{Videocard.core}','{Videocard.shaderblock}','{Videocard.energy}','{Videocard.date}','{Videocard.price}', '{Videocard.info}', @img)";

MySqlCommand com = new MySqlCommand(addQuerry, DB.getconnection());

com.Parameters.AddWithValue("img", Videocard.link);

com.ExecuteNonQuery();

MessageBox.Show("Запис успішно додано!", "Юху!", MessageBoxButtons.OK, MessageBoxIcon.Information);

MainForm.AdminForm form = new MainForm.AdminForm();

this.Hide();

form.Show();

}

else

{

MessageBox.Show("Помилка введення чисел!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

DB.close();

}

private void FormAdd\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.AdminForm form = new MainForm.AdminForm();

form.Show();

}

private void buttonimg\_Click(object sender, EventArgs e)

{

ofd.Filter = "Image Files(\*.JPG;\*.PNG)|\*.JPG;\*.PNG|All files(\*.\*)|\*.\*";

if (ofd.ShowDialog() == DialogResult.OK)

{

try

{

Videocard.link = ofd.FileName;

}

catch

{

MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

Лістинг FormEdit.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

namespace WinForms

{

public partial class FormEdit : Form

{

public int tmp;

public FormEdit()

{

InitializeComponent();

}

OpenFileDialog ofd = new OpenFileDialog();

private void FormAdd\_Load(object sender, EventArgs e)

{

textBoxname.Text = Videocard.name;

textBoxtechproc.Text = Videocard.techproc.ToString();

textBoxmemorytype.Text = Videocard.typememory;

textBoxmemory.Text = Videocard.memory.ToString();

textBoxmemorybus.Text = Videocard.memorybus;

textBoxcore.Text = Videocard.core.ToString();

textBoxshaderblock.Text = Videocard.shaderblock.ToString();

textBoxenergy.Text = Videocard.energy.ToString();

dateTimePicker1.Text = Videocard.date;

textBoxprice.Text = Videocard.price.ToString();

labelid.Text = Videocard.id.ToString();

textBoxinfo.Text = Videocard.info;

ofd.FileName = Videocard.link;

}

private void buttonedit\_Click(object sender, EventArgs e)

{

MainForm.AdminForm f = new MainForm.AdminForm();

f.tmp = labelid.Text;

Videocard.id = Convert.ToInt32(labelid.Text);

Videocard.name = textBoxname.Text;

int techproc = Videocard.techproc;

Videocard.typememory = textBoxmemorytype.Text;

int memory = Videocard.memory;

Videocard.memorybus = textBoxmemorybus.Text;

int core = Videocard.core;

int shaderblock = Videocard.shaderblock;

int energy = Videocard.energy;

Videocard.date = dateTimePicker1.Value.ToString("s");

int price = Videocard.price;

Videocard.info = textBoxinfo.Text;

Videocard.link = ofd.FileName;

if (int.TryParse(textBoxtechproc.Text, out techproc) && int.TryParse(textBoxmemory.Text, out memory) && int.TryParse(textBoxcore.Text, out core) && int.TryParse(textBoxshaderblock.Text, out shaderblock) && int.TryParse(textBoxenergy.Text, out energy) && int.TryParse(textBoxprice.Text, out price))

{

Videocard.techproc = techproc;

Videocard.memory = memory;

Videocard.core = core;

Videocard.shaderblock = shaderblock;

Videocard.energy = energy;

Videocard.price = price;

this.Hide();

f.Show();

f.change();

}

else

{

MessageBox.Show("Помилка введення чисел!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void buttondel\_Click(object sender, EventArgs e)

{

MainForm.AdminForm f = new MainForm.AdminForm();

f.tmp = labelid.Text;

this.Hide();

f.Show();

f.delete();

}

private void FormEdit\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.AdminForm f = new MainForm.AdminForm();

f.Show();

}

private void buttonimg\_Click(object sender, EventArgs e)

{

ofd.Filter = "Image Files(\*.JPG;\*.PNG)|\*.JPG;\*.PNG|All files(\*.\*)|\*.\*";

if(ofd.ShowDialog() == DialogResult.OK)

{

try

{

Videocard.link = ofd.FileName;

}

catch

{

MessageBox.Show("Неправильний формат картинки!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

Лістинг PorivForm.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace WinForms

{

public partial class PorivForm : Form

{

db DB = new db();

int[] array = new int[100];

int j;

int i = 0;

public PorivForm()

{

InitializeComponent();

}

public void arrayadd(int[] a, int i)

{

array = a;

j = i;

}

public void CreateColumns()

{

dgv.Columns.Add("id", "№");

dgv.Columns.Add("name", "Назва");

dgv.Columns.Add("techproc", "Техпроцес, нм");

dgv.Columns.Add("typememory", "Тип пам'яті");

dgv.Columns.Add("memory", "Об`єм пам`яті");

dgv.Columns.Add("memorybus", "Шина пам'яті");

dgv.Columns.Add("core", "Частота роботи ядра, МГц");

dgv.Columns.Add("shaderblock", "Шейдерних блоків");

dgv.Columns.Add("energy", "Споживання енергії, Вт");

dgv.Columns.Add("date", "Дата випуску");

dgv.Columns.Add("price", "Вартість, $");

}

public void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10)); ;

}

private void Refreshdgv(DataGridView dgv)

{

string str = $"select \* from videocard where id = '{array[i]}'";

MySqlCommand com = new MySqlCommand(str, DB.getconnection());

DB.open();

MySqlDataReader reader = com.ExecuteReader();

while (reader.Read())

ReadRow(dgv, reader);

DB.close();

}

private void PorivForm\_Load(object sender, EventArgs e)

{

CreateColumns();

while (i <= j)

{

Refreshdgv(dgv);

i++;

}

styledatagridview();

dgv.Columns["id"].Visible = false;

dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);

}

private void styledatagridview()

{

dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;

dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;

dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);

dgv.DefaultCellStyle.SelectionForeColor = Color.White;

dgv.BackgroundColor = Color.Black;

dgv.EnableHeadersVisualStyles = false;

dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;

dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);

dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);

dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;

}

private void PorivForm\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.UserForm form = new MainForm.UserForm();

form.Show();

}

}

}

Лістинг signinform.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace signinform

{

public partial class signinform : Form

{

db DB = new db();

public signinform()

{

InitializeComponent();

this.textBoxpass.AutoSize = false;

this.textBoxpass.Size = new Size(this.textBoxpass.Size.Width, 46);

}

public static Regex regex;

private void button1\_Click(object sender, EventArgs e)

{

string login = textBoxlogin.Text;

string pass = textBoxpass.Text;

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand com = new MySqlCommand("SELECT \* FROM user WHERE login=@uL AND password=@uP", DB.getconnection());

com.Parameters.Add("@uL", MySqlDbType.VarChar).Value = login;

com.Parameters.Add("@uP", MySqlDbType.VarChar).Value = pass;

adapter.SelectCommand = com;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Авторизовано", "Успіх!", MessageBoxButtons.OK, MessageBoxIcon.Information);

if (login == "admin" && pass == "admin")

{

this.Hide();

MainForm.AdminForm f = new MainForm.AdminForm();

f.Show();

}

else

{

this.Hide();

MainForm.UserForm f = new MainForm.UserForm();

f.Show();

}

}

else

MessageBox.Show("Помилка авторизації! Перевірьте введенні дані", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void signinform\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.StartForm form = new MainForm.StartForm();

form.Show();

}

private void pictureBoxvisible\_Click(object sender, EventArgs e)

{

if (textBoxpass.UseSystemPasswordChar == true)

{

pictureBoxvisible.Image = Image.FromFile("C:\\Users\\Victor\\source\\repos\\kursova\\MainForm\\images\\view.png");

textBoxpass.UseSystemPasswordChar = false;

}

else

{

pictureBoxvisible.Image = Image.FromFile("C:\\Users\\Victor\\source\\repos\\kursova\\MainForm\\images\\hide.png");

textBoxpass.UseSystemPasswordChar = true;

}

}

}

}

Лістинг signupform.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace signupform

{

public partial class signupform : Form

{

public signupform()

{

InitializeComponent();

this.textBoxpass.AutoSize = false;

this.textBoxpass.Size = new Size(this.textBoxpass.Size.Width, 46);

}

public static Regex regex;

db DB = new db();

private void button1\_Click(object sender, EventArgs e)

{

int tmp = 0;

regex = new Regex(@"^[a-zA-Z][a-zA-Z0-9-\_\.]{1,15}$");

if (!(regex.IsMatch(textBoxlogin.Text)))

{

tmp++;

MessageBox.Show("Помилка введення логіну", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

regex = new Regex(@"([a-z]+[A-Z]+[0-9]+|[a-z]+[0-9]+[A-Z]+|[A-Z]+[a-z]+[0-9]+|[A-Z]+[0-9]+[a-z]+|[0-9]+[a-z]+[A-Z]+|[0-9]+[A-Z]+[a-z]+)");

if (!(regex.IsMatch(textBoxpass.Text)))

{

tmp++;

MessageBox.Show("Помилка введення паролю\nМає бути щонайменше один символ верхнього регістру, один нижнього і цифра", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

DB.open();

MySqlCommand command = new MySqlCommand("select id from user where login= @uL", DB.getconnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = textBoxlogin.Text;

int id = Convert.ToInt32(command.ExecuteScalar());

if (id == 0)

{

MySqlCommand com = new MySqlCommand("INSERT INTO user (id, login, password) VALUES (NULL, @login, @password);", DB.getconnection());

com.Parameters.AddWithValue("login", textBoxlogin.Text);

com.Parameters.AddWithValue("password", textBoxpass.Text);

if (tmp == 0)

if (Convert.ToInt32(com.ExecuteNonQuery()) != 0)

{

MessageBox.Show("Реєстрація пройшла успішно!", "Успіх!", MessageBoxButtons.OK, MessageBoxIcon.Information);

DB.close();

this.Hide();

MainForm.StartForm f = new MainForm.StartForm();

f.Show();

}

}

else

{

MessageBox.Show("Даний логін вже існує", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void signupform\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.StartForm form = new MainForm.StartForm();

form.Show();

}

}

}

Лістинг StartForm.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace MainForm

{

public partial class StartForm : Form

{

db DB = new db();

public StartForm()

{

InitializeComponent();

}

public void CreateColumns()

{

dgv.Columns.Add("id", "№");

dgv.Columns.Add("name", "Назва");

dgv.Columns.Add("techproc", "Техпроцес, нм");

dgv.Columns.Add("typememory", "Тип пам'яті");

dgv.Columns.Add("memory", "Об`єм пам`яті");

dgv.Columns.Add("memorybus", "Шина пам'яті");

dgv.Columns.Add("core", "Частота роботи ядра, МГц");

dgv.Columns.Add("shaderblock", "Шейдерних блоків");

dgv.Columns.Add("energy", "Споживання енергії, Вт");

dgv.Columns.Add("date", "Дата випуску");

dgv.Columns.Add("price", "Вартість, $");

dgv.Columns.Add("status", String.Empty);

}

public void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), Rowstatus.New); ;

}

private void Refreshdgv(DataGridView dgv)

{

dgv.Rows.Clear();

string str = $"select \* from videocard";

MySqlCommand com = new MySqlCommand(str, DB.getconnection());

DB.open();

MySqlDataReader reader = com.ExecuteReader();

while (reader.Read())

ReadRow(dgv, reader);

DB.close();

}

private void Form1\_Load(object sender, EventArgs e)

{

CreateColumns();

Refreshdgv(dgv);

styledatagridview();

dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);

dgv.Columns["status"].Visible = false;

dgv.Columns["id"].Visible = false;

}

private void styledatagridview()

{

dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;

dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;

dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);

dgv.DefaultCellStyle.SelectionForeColor = Color.White;

dgv.BackgroundColor = Color.Black;

dgv.EnableHeadersVisualStyles = false;

dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;

dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);

dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);

dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;

}

private void buttonsignin\_Click(object sender, EventArgs e)

{

signinform.signinform signin = new signinform.signinform();

this.Hide();

signin.StartPosition = FormStartPosition.CenterScreen;

signin.Show();

}

private void buttonsignup\_Click(object sender, EventArgs e)

{

signupform.signupform signup = new signupform.signupform();

this.Hide();

signup.StartPosition = FormStartPosition.CenterScreen;

signup.Show();

}

private void textBoxsearch\_TextChanged(object sender, EventArgs e)

{

search(dgv);

}

private void search(DataGridView dgv)

{

dgv.Rows.Clear();

string searchquerry = $"select \* from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%" + textBoxsearch.Text + "%'";

MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

private void StartForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Ви впевнені що хочете вийти?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else if (dialogResult == DialogResult.No)

{

e.Cancel = true;

}

}

}

}

Лістинг UserForm.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace MainForm

{

public partial class UserForm : Form

{

public string tmp;

int selectedrow;

db DB = new db();

public UserForm()

{

InitializeComponent();

}

public void CreateColumns()

{

dgv.Columns.Add("id", "№");

dgv.Columns.Add("name", "Назва");

dgv.Columns.Add("techproc", "Техпроцес, нм");

dgv.Columns.Add("typememory", "Тип пам'яті");

dgv.Columns.Add("memory", "Об`єм пам`яті");

dgv.Columns.Add("memorybus", "Шина пам'яті");

dgv.Columns.Add("core", "Частота роботи ядра, МГц");

dgv.Columns.Add("shaderblock", "Шейдерних блоків");

dgv.Columns.Add("energy", "Споживання енергії, Вт");

dgv.Columns.Add("date", "Дата випуску");

dgv.Columns.Add("price", "Вартість, $");

dgv.Columns.Add("status", String.Empty);

DataGridViewCheckBoxColumn dgvcheckbox = new DataGridViewCheckBoxColumn();

dgvcheckbox.ValueType = typeof(bool);

dgvcheckbox.Name = "check";

dgvcheckbox.HeaderText = "Порівняння";

dgvcheckbox.ReadOnly = false;

dgv.Columns.Add(dgvcheckbox);

}

public void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetByte(0), record.GetString(1), record.GetByte(2), record.GetString(3), record.GetInt32(4), record.GetString(5), record.GetInt32(6), record.GetInt32(7), record.GetInt32(8), record.GetDateTime(9).Date.ToString("d"), record.GetInt32(10), Rowstatus.New); ;

}

private void Refreshdgv(DataGridView dgv)

{

dgv.Rows.Clear();

string str = $"select \* from videocard";

MySqlCommand com = new MySqlCommand(str, DB.getconnection());

DB.open();

MySqlDataReader reader = com.ExecuteReader();

while (reader.Read())

ReadRow(dgv, reader);

DB.close();

}

private void Form1\_Load(object sender, EventArgs e)

{

panelform.Width = 1;

button\_poriv.Visible = false;

buttonok.Visible = false;

label1.Visible = false;

label7.Visible = false;

textBoxot.Visible = false;

textBoxdo.Visible = false;

checkBox\_4096.Visible = false;

checkBox\_6144.Visible = false;

checkBox\_8192.Visible = false;

checkBox\_10240.Visible = false;

checkBox\_12228.Visible = false;

checkBox\_16384.Visible = false;

checkBox\_24576.Visible = false;

checkBox\_amd.Visible = false;

checkBox\_nvidia.Visible = false;

label2.Visible = false;

label3.Visible = false;

label4.Visible = false;

label5.Visible = false;

label6.Visible = false;

CreateColumns();

Refreshdgv(dgv);

styledatagridview();

dgv.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.RowHeadersDefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dgv.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);

dgv.Columns["id"].ReadOnly = true;

dgv.Columns["name"].ReadOnly = true;

dgv.Columns["techproc"].ReadOnly = true;

dgv.Columns["typememory"].ReadOnly = true;

dgv.Columns["memory"].ReadOnly = true;

dgv.Columns["memorybus"].ReadOnly = true;

dgv.Columns["core"].ReadOnly = true;

dgv.Columns["shaderblock"].ReadOnly = true;

dgv.Columns["energy"].ReadOnly = true;

dgv.Columns["date"].ReadOnly = true;

dgv.Columns["price"].ReadOnly = true;

dgv.Columns["status"].Visible = false;

dgv.Columns["id"].Visible = false;

dgv.Columns["check"].Visible = false;

}

private void styledatagridview()

{

dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Black;

dgv.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;

dgv.DefaultCellStyle.SelectionBackColor = Color.FromArgb(50, 50, 50);

dgv.DefaultCellStyle.SelectionForeColor = Color.White;

dgv.BackgroundColor = Color.Black;

dgv.EnableHeadersVisualStyles = false;

dgv.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;

dgv.ColumnHeadersDefaultCellStyle.Font = new Font("MS Times New Roman", 12);

dgv.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(25, 25, 25);

dgv.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;

}

private void buttonexit\_Click(object sender, EventArgs e)

{

DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult1 == DialogResult.Yes)

{

DialogResult dialogResult2 = MessageBox.Show("Ви точно бажаєте покинути ваш чудовий акаунт?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult2 == DialogResult.Yes)

{

MainForm.StartForm f = new MainForm.StartForm();

this.Hide();

f.Show();

}

}

}

private void textBoxsearch\_TextChanged(object sender, EventArgs e)

{

search(dgv);

}

private void search(DataGridView dgv)

{

dgv.Rows.Clear();

string searchquerry = $"select \* from videocard where concat (id, name, techproc, typememory, memory, memorybus, core, shaderblock, energy, date, price) like '%" + textBoxsearch.Text + "%'";

MySqlCommand com = new MySqlCommand(searchquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

private void labelfilt\_Click(object sender, EventArgs e)

{

if (label2.Visible == false)

{

labelporiv.Visible = false;

textBoxsearch.ReadOnly = true;

panelform.Width = 155;

buttonok.Visible = true;

checkBox\_4096.Visible = true;

checkBox\_6144.Visible = true;

checkBox\_8192.Visible = true;

checkBox\_10240.Visible = true;

checkBox\_12228.Visible = true;

checkBox\_16384.Visible = true;

checkBox\_24576.Visible = true;

checkBox\_amd.Visible = true;

checkBox\_nvidia.Visible = true;

label2.Visible = true;

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

label6.Visible = true;

label1.Visible = true;

label7.Visible = true;

textBoxot.Visible = true;

textBoxdo.Visible = true;

Refreshdgv(dgv);

}

else

{

labelporiv.Visible = true;

textBoxsearch.ReadOnly = false;

panelform.Width = 1;

buttonok.Visible = false;

label1.Visible = false;

label7.Visible = false;

textBoxot.Visible = false;

textBoxdo.Visible = false;

checkBox\_4096.Visible = false;

checkBox\_6144.Visible = false;

checkBox\_8192.Visible = false;

checkBox\_10240.Visible = false;

checkBox\_12228.Visible = false;

checkBox\_16384.Visible = false;

checkBox\_24576.Visible = false;

checkBox\_amd.Visible = false;

checkBox\_nvidia.Visible = false;

label2.Visible = false;

label3.Visible = false;

label4.Visible = false;

label5.Visible = false;

label6.Visible = false;

Refreshdgv(dgv);

}

}

private void checkBox\_amd\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void checkBox\_nvidia\_CheckedChanged(object sender, EventArgs e)

{

check();

}

int maxmem = 0, minmem = 25000;

int minprice = 0, maxprice = 99999;

private int findmax()

{

if (checkBox\_24576.Checked == true && Convert.ToInt32(checkBox\_24576.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_24576.Text);

return maxmem;

}

else if (checkBox\_16384.Checked == true && Convert.ToInt32(checkBox\_16384.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_16384.Text);

return maxmem;

}

if (checkBox\_12228.Checked == true && Convert.ToInt32(checkBox\_12228.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_12228.Text);

return maxmem;

}

else if (checkBox\_10240.Checked == true && Convert.ToInt32(checkBox\_10240.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_10240.Text);

return maxmem;

}

else if (checkBox\_8192.Checked == true && Convert.ToInt32(checkBox\_8192.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_8192.Text);

return maxmem;

}

else if (checkBox\_6144.Checked == true && Convert.ToInt32(checkBox\_6144.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_6144.Text);

return maxmem;

}

else if (checkBox\_4096.Checked == true && Convert.ToInt32(checkBox\_4096.Text) > minmem)

{

maxmem = Convert.ToInt32(checkBox\_4096.Text);

return maxmem;

}

else return maxmem = 25000;

}

private int findmin()

{

if (checkBox\_4096.Checked == true && Convert.ToInt32(checkBox\_4096.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_4096.Text);

return minmem;

}

else if (checkBox\_6144.Checked == true && Convert.ToInt32(checkBox\_6144.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_6144.Text);

return minmem;

}

else if (checkBox\_8192.Checked == true && Convert.ToInt32(checkBox\_8192.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_8192.Text);

return minmem;

}

else if (checkBox\_10240.Checked == true && Convert.ToInt32(checkBox\_10240.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_10240.Text);

return minmem;

}

else if (checkBox\_12228.Checked == true && Convert.ToInt32(checkBox\_12228.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_12228.Text);

return minmem;

}

else if (checkBox\_16384.Checked == true && Convert.ToInt32(checkBox\_16384.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_16384.Text);

return minmem;

}

else if (checkBox\_24576.Checked == true && Convert.ToInt32(checkBox\_24576.Text) < minmem)

{

minmem = Convert.ToInt32(checkBox\_24576.Text);

return minmem;

}

else return minmem = 0;

}

private void check()

{

maxmem = 0;

int max = findmax();

minmem = 25000;

int min = findmin();

if (checkBox\_nvidia.Checked == false && checkBox\_amd.Checked == true)

{

dgv.Rows.Clear();

string filterquerry = $"select \* from videocard where memory between {minmem} and {maxmem} and price between {minprice} and {maxprice} and name like '%" + checkBox\_amd.Text + "%'";

MySqlCommand com = new MySqlCommand(filterquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

else if (checkBox\_nvidia.Checked == true && checkBox\_amd.Checked == false)

{

dgv.Rows.Clear();

string filterquerry = $"select \* from videocard where memory between {minmem} and {maxmem} and price between {minprice} and {maxprice} and name like '%" + checkBox\_nvidia.Text + "%'";

MySqlCommand com = new MySqlCommand(filterquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

else

{

dgv.Rows.Clear();

string filterquerry = $"select \* from videocard where memory between {minmem} and {maxmem} and price between {minprice} and {maxprice}";

MySqlCommand com = new MySqlCommand(filterquerry, DB.getconnection());

DB.open();

MySqlDataReader read = com.ExecuteReader();

while (read.Read())

{

ReadRow(dgv, read);

}

read.Close();

}

}

private void checkBox\_24576\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void checkBox\_16384\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void checkBox\_12228\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void checkBox\_10240\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void checkBox\_8192\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBoxdo.Text != String.Empty && textBoxot.Text != String.Empty)

{

if (Convert.ToInt32(textBoxot.Text) >= Convert.ToInt32(textBoxdo.Text))

{

MessageBox.Show("Від більше До!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

minprice = int.Parse(textBoxot.Text);

maxprice = int.Parse(textBoxdo.Text);

check();

}

}

}

private void UserForm\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialogResult1 = MessageBox.Show("Ви впевнені що хочете вийти?", "Вихід", MessageBoxButtons.YesNo);

if (dialogResult1 == DialogResult.Yes)

{

Application.Exit();

}

else

{

e.Cancel = true;

}

}

private void checkBox\_6144\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void labelporiv\_Click(object sender, EventArgs e)

{

if (dgv.Columns["check"].Visible == false)

{

button\_poriv.Visible = true;

textBoxsearch.Visible = false;

labelfilt.Visible = false;

dgv.Columns["check"].Visible = true;

}

else

{

button\_poriv.Visible = false;

textBoxsearch.Visible = true;

labelfilt.Visible = true;

dgv.Columns["check"].Visible = false;

}

}

private void button\_poriv\_Click(object sender, EventArgs e)

{

int[] array = new int[100];

int j = 0;

for (int i = 0; i < dgv.RowCount; i++)

{

if (Convert.ToBoolean(dgv.Rows[i].Cells[12].Value) == true)

{

array[j] = Convert.ToInt32(dgv.Rows[i].Cells[0].Value);

j++;

}

}

if (j > 0)

{

this.Hide();

WinForms.PorivForm f = new WinForms.PorivForm();

f.arrayadd(array, j);

f.Show();

}

}

private void checkBox\_4096\_CheckedChanged(object sender, EventArgs e)

{

check();

}

private void dgv\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

selectedrow = e.RowIndex;

int selectedcolumn = e.ColumnIndex;

if (selectedcolumn == 1)

{

if (e.RowIndex >= 0)

{

if (dgv.Rows[selectedrow].Cells[1].Value.ToString() != String.Empty)

{

Videocard.name = dgv.Rows[selectedrow].Cells[1].Value.ToString();

string querryimg = "select img from videocard where name like '%" + Videocard.name + "%'";

string querryinfo = "select info from videocard where name like '%" + Videocard.name + "%'";

DB.open();

MySqlCommand com = new MySqlCommand(querryimg, DB.getconnection());

Videocard.link = (string)com.ExecuteScalar();

com = new MySqlCommand(querryinfo, DB.getconnection());

Videocard.info = (string)com.ExecuteScalar();

DB.close();

this.Hide();

WinForms.Videocardinfo f = new WinForms.Videocardinfo();

f.Show();

}

}

}

}

}

}

Лістинг Videocardinfo.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using lib;

using MySql.Data.MySqlClient;

namespace WinForms

{

public partial class Videocardinfo : Form

{

db DB = new db();

public Videocardinfo()

{

InitializeComponent();

}

private void Videocard\_Load(object sender, EventArgs e)

{

pictureBox1.Image = Image.FromFile(Videocard.link);

labelname.Text = Videocard.name;

labelinfo.Text = Videocard.info;

pictureBox1.Focus();

}

private void Videocardinfo\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm.UserForm form = new MainForm.UserForm();

form.Show();

}

}

}