

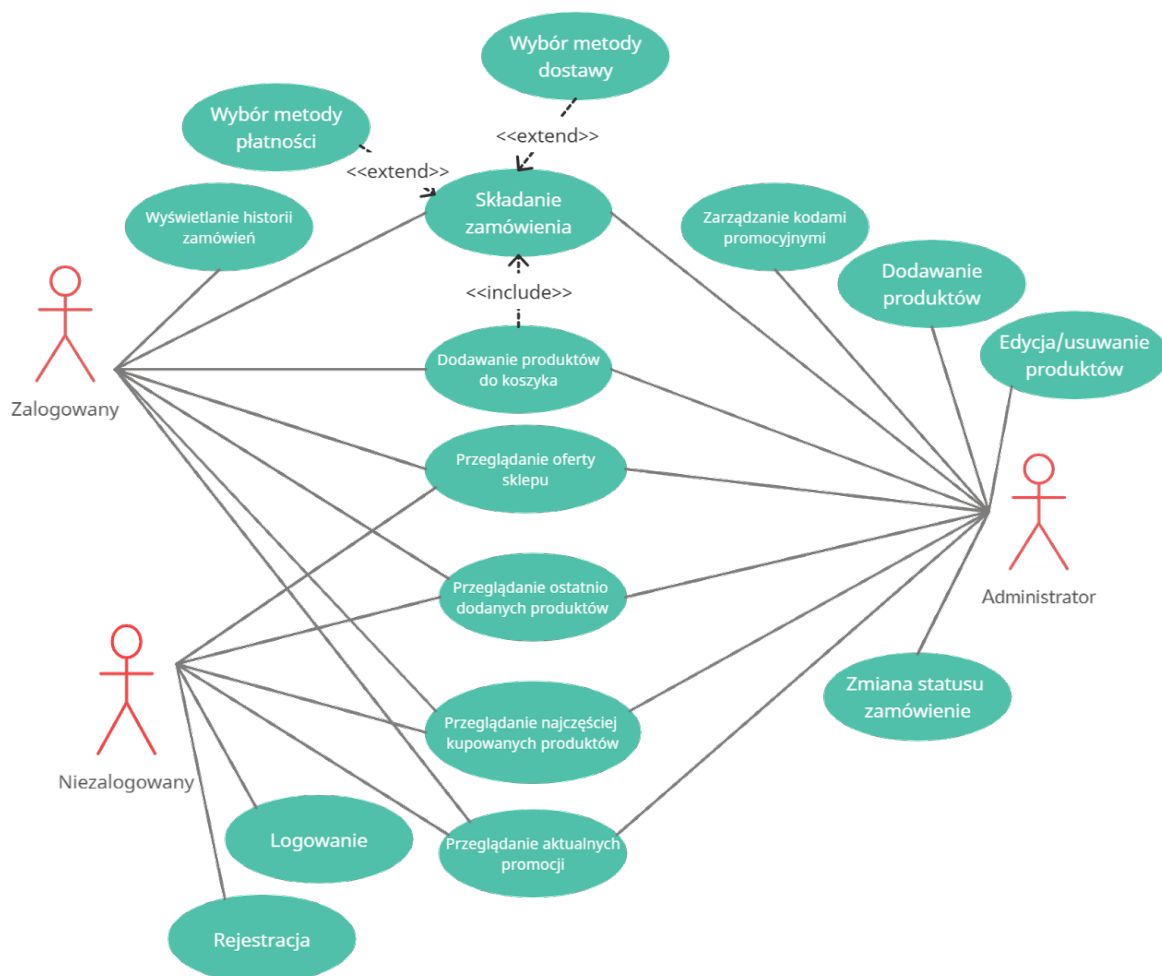
Zespół:

- Paweł Awramiuk
- Gabriel Chomiczewski
- Łukasz Czeremcha
- Mateusz Jeżewski

1. Opis aplikacji bazodanowej. Diagram przypadków użycia

Tematem projektu jest sklep internetowy z artykułami komputerowymi, umożliwiający klientom składanie zamówień. Administratorowi udostępniony jest panel administratora, z poziomu którego może zarządzać produktami w sklepie, poszczególnymi cechami produktów np. kategorią, oraz tworzyć i edytować kody promocyjne dla określonych produktów.

Diagram przypadków użycia:



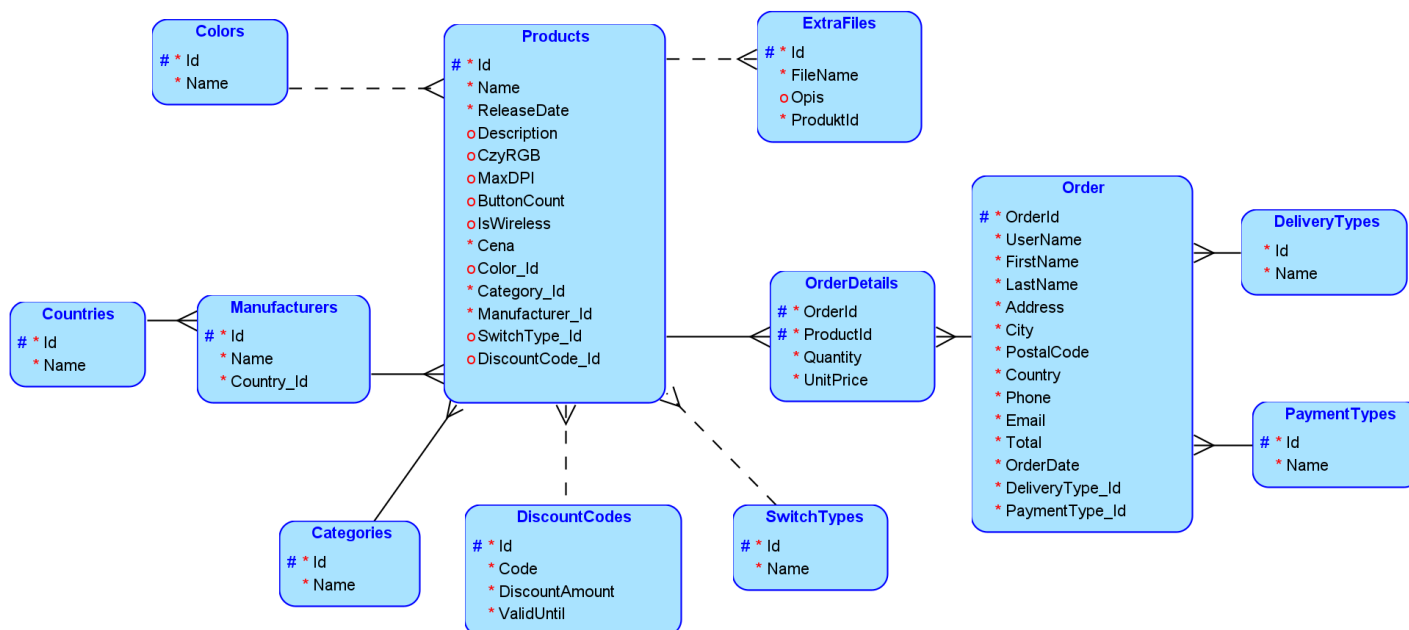
Opisy użytkowników (aktorów):

- **Niezalogowani użytkownicy** mogą przeglądać ofertę sklepu z podziałem na kategorie produktów, ostatnio dodane produkty, najczęściej kupowane produkty, wyświetlać listę obecnie występujących promocji oraz logować lub rejestrować się na stronie.
- **Zalogowani klienci** mogą dodatkowo składać zamówienia poprzez dodawanie wybranych produktów do koszyka, na końcu podając dane do zamówienia, które przesyłane są do bazy danych. Na stronie swojego profilu mogą wyświetlić historię składanych przez siebie zamówień.

- **Administrator** może robić wszystko to, co klienci sklepu, a dodatkowo ma dostęp do panelu administratora, w którym może dodawać nowe produkty, kategorie, określone parametry produktów (np. kolor, producentów), a także zarządzać obecnie aktywnymi kodami rabatowymi i dodawać nowe. Administrator może edytować właściwości wszystkich produktów, usuwać produkty, a także zmieniać status realizacji zamówień.

2. Model koncepcyjny bazy danych.

2.1 Diagram ER



Klucze główne i obce w tabelach:

- **Countries** - Id (główny)
- **Colors** - Id (główny)
- **Manufacturers** - Id (główny), Country_Id (obcy)
- **Products** - Id (główny); Color_Id, Category_Id, Manufacturer_Id, SwitchType_Id, DiscountCode_Id (obce)
- **Categories** - Id (główny)
- **ExtraFiles** - Id (główny), ProduktId (obcy)
- **DiscountCodes** - Id (główny)
- **SwitchTypes** - Id (główny)
- **Order** - OrderId (główny), DeliveryType_Id, PaymentType_Id (obce)
- **OrderDetails** – (OrderId, ProductId) (główny)
- **DeliveryTypes** – Id (główny)
- **PaymentTypes** – Id (główny)

2.2 Diagram relacyjny

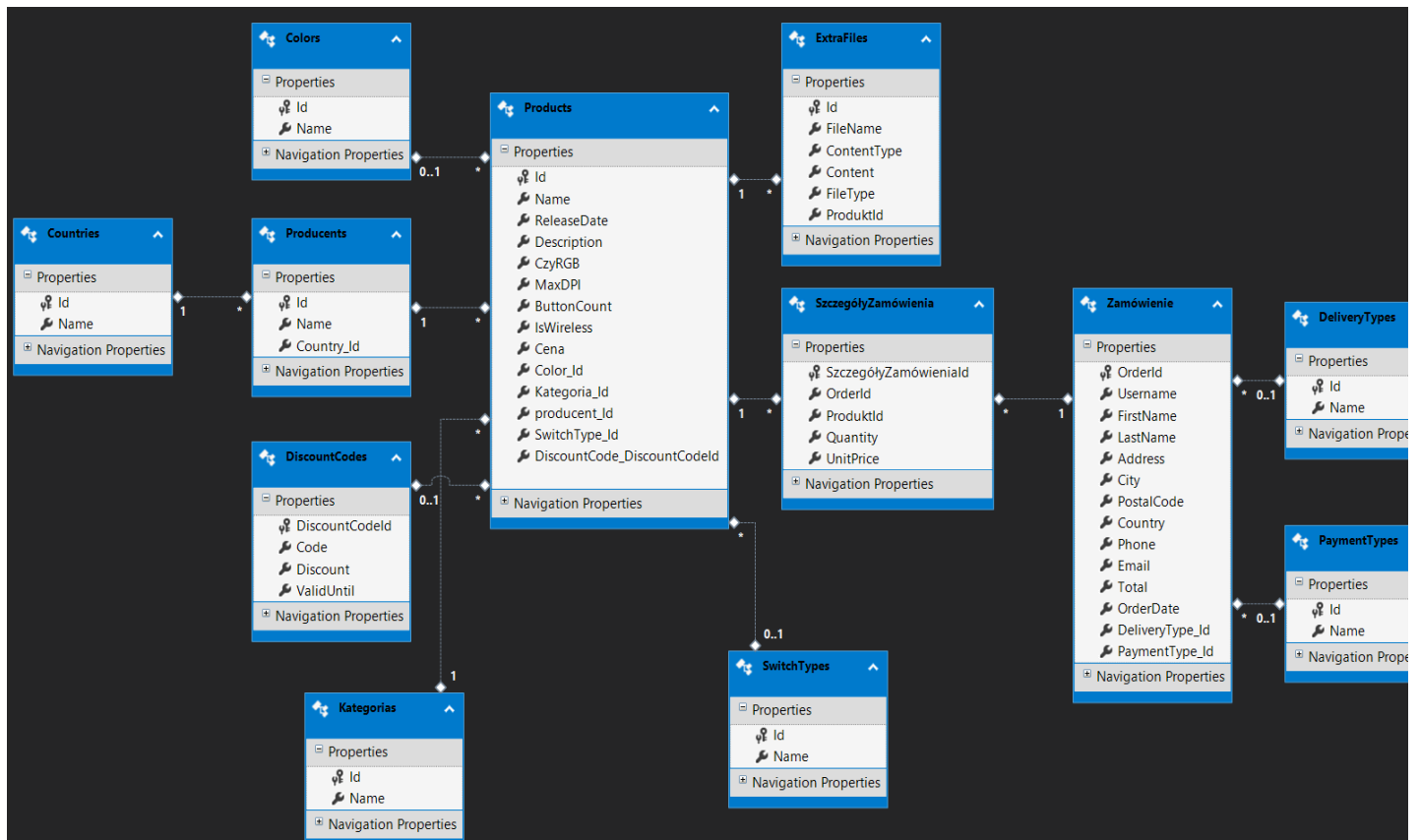


Diagram wygenerowany w MS Visual Studio 2019 na podstawie klas stworzonych we wstępnej wersji aplikacji. Ilość klas w końcowej wersji programu może być większa. Baza danych zmapowana została przez Entity Framework.

Co przechowuje rekord tabeli:

- **Countries:**
 - Id – identyfikator kraju
 - Name - nazwa kraju pochodzenia producenta produktu
- **Colors:**
 - Id - identyfikator koloru
 - Name - nazwa koloru produktu
- **Producents:**
 - Id – identyfikator producenta
 - Country_Id - id kraju pochodzenia
 - Name - nazwa producenta
- **Products**
 - Id – identyfikator produktu
 - Name – nazwa produktu
 - ReleaseDate – data dodania do sklepu
 - Description – opis produktu
 - CzyRGB – czy produkt posiada podświetlanie diodami RGB
 - ButtonCount – ilość przycisków danego urządzenia
 - IsWireless – czy urządzenie jest bezprzewodowe
 - Cena – cena produktu
 - Color_Id – identyfikator koloru produktu (jeśli posiada)
 - Kategoria_Id – identyfikator kategorii produktu
 - producent_Id – identyfikator producenta produktu

- SwitchType_Id – identyfikator typu przełączników (w przypadku klawiatur)
- DiscountCode_DiscountCodeId – identyfikator przypisanego produktowi kodu rabatowego.
- **Kategorias:**
 - Id – identyfikator kategorii
 - Name - nazwa kategorii produktu
- **ExtraFiles (to może ulec zmianie):**
 - id – identyfikator pliku
 - FileName - nazwa pliku dodatkowego do produktu
 - ContentType – typ zawartości pliku (zdjęcie, instrukcja itp.)
 - FileType – rozszerzenie pliku
 - ProduktId – identyfikator produktu, do którego plik jest przypisany (założono, że jeden plik może być przypisany tylko do jednego produktu)
- **DiscountCodes:**
 - id – identyfikator kodu rabatowego
 - Code – treść kodu
 - Discount - wartość zniżki
 - ValidUntil - data ważności kodu promocyjnego
- **SwitchTypes:**
 - id – identyfikator typu przełączników
 - Name - nazwa przełączników w klawiaturze
- **Zamówienie:**
 - id – identyfikator zamówienia
 - Username, FirstName, LastName, Address, City, PostalCode, Country, Phone, Email - dane adresowe zamawiającego
 - Total – całkowita wartość zamówienia
 - OrderDate – data złożenia zamówienia
 - DeliveryType_Id – identyfikator wybranego rodzaju dostawy zamówienia
 - PaymentType_Id - identyfikator wybranego rodzaju płatności
- **SzczegółyZamówienia**
 - SzczegółyZamówieniaId – identyfikator pozycji w zamówieniu
 - OrderId – identyfikator całego zamówienia
 - ProduktId – identyfikator produktu
 - Quantity – ilość produktu w zamówieniu
 - UnitPrice - cena
- **DeliveryType**
 - Id – identyfikator dostępnego rodzaju dostawy zamówienia
 - Name - nazwa dostępnego rodzaju dostawy zamówienia
- **PaymentType:**
 - Id – identyfikator dostępnego rodzaju płatności przy realizacji zamówienia
 - Name - nazwa dostępnego rodzaju płatności przy realizacji zamówienia

3. Wybór serwera bazodanowego i implementacja bazy danych.

Aplikacja tworzona jest w oparciu o platformę ASP.NET MVC z wykorzystaniem Entity Framework. Serwerem aplikacji jest IIS Express, zaś serwerem bazodanowym – Microsoft SQL Server. Wykorzystano podejście Code-First.

4. Mapowanie klas na tabele bazodanowe

W aplikacji wykorzystane jest podejście Code-First, gdzie najpierw tworzone są klasy obiektów w języku C#.

```
public class Product
{
    public int Id { get; set; }

    [Display(Name = "Producent: ")]
    public Producent producent { get; set; }

    [Display(Name = "Model: ")]
    public string Name { get; set; }

    [Display(Name = "Data wydania: ")]
    public DateTime ReleaseDate { get; set; }

    [Display(Name = "Kategoria: ")]
    public Kategoria Kategoria { get; set; }

    [Display(Name = "Opis: ")]
    public string Description { get; set; }

    [Display(Name = "RGB: ")]
    public bool CzyRGB { get; set; }

    [Display(Name = "Max DPI: ")]
    public int MaxDPI { get; set; }

    [Display(Name = "Ilość przycisków: ")]
    public int ButtonCount { get; set; }

    [Display(Name = "Rodzaj przełączników: ")]
    public SwitchType SwitchType { get; set; }

    [Display(Name = "Kolor obudowy: ")]
    public Color Color { get; set; }

    [Display(Name = "Bezprzewodowa: ")]
    public bool IsWireless { get; set; }

    [Display(Name = "Cena (PLN): ")]
    public decimal Cena { get; set; }

    [Display(Name = "Pliki dodatkowe: ")]
    public virtual ICollection<ExtraFile> PlikiDodatkowe { get; set; }
}
```

Klasa obiektu „Produkt” w C#.

Następnie tworzona jest klasa dziedzicząca z klasy DbContext, zawierająca obiekty klasy DbSet dla określonych klas obiektów. Na ich podstawie tworzone będą tabele w bazie danych.

```
public class XmoreltronikEntities : DbContext
{
    public DbSet<Product> Products { get; set; }
    public DbSet<Kategoria> Categories { get; set; }
    public DbSet<Color> Colors { get; set; }
    public DbSet<Country> Countries { get; set; }
    public DbSet<ExtraFile> ExtraFiles { get; set; }
    public DbSet<Producent> Producers { get; set; }
    public DbSet<SwitchType> SwitchTypes { get; set; }

    public DbSet<Koszyk> Koszyki { get; set; }
    public DbSet<Zamówienie> Zamówienia { get; set; }
    public DbSet<SzczegółyZamówienia> OrderDetails { get; set; }

    public DbSet<DiscountCode> DiscountCodes { get; set; }
    public DbSet<PaymentType> PaymentTypes { get; set; }
    public DbSet<DeliveryType> DeliveryTypes { get; set; }
```

Na podstawie klasy XmoreltronikEntities oraz klasy XmoreltronikConfiguration można utworzyć migrację bazy danych, która zmapuje klasy obiektów na tabele w bazie danych.

```
internal sealed class XmoreltronikConfiguration : DbMigrationsConfiguration<Models.XmoreltronikEntities>
{
    Odwołania: 0
    public XmoreltronikConfiguration()
    {
        AutomaticMigrationsEnabled = false;
        MigrationsDirectory = @"Migrations\XmoreltronikEntities";
        ContextKey = "ProjektMVCPodejscie4.Models.XmoreltronikEntities";
    }

    1 odwołanie
    protected override void Seed(Models.XmoreltronikEntities context)
```

Klasa XmoreltronikConfiguration jest konieczna do określenia przy wydaniu poleceń, ponieważ aplikacja korzysta z dwóch baz danych: jednej do przechowywania wszystkich przedstawionych powyżej elementów sklepu, a druga, tworzona automatycznie przy tworzeniu projektu, do przechowywania informacji o użytkownikach. Migracje tworzono poprzez wydanie w „Konsoli menedżera pakietów” programu MS Visual Studio 2019 polecenia:

```
Add-Migration 'nazwa_migracji' -ConfigurationTypeName XmoreltronikConfiguration
Update-Database -ConfigurationTypeName XmoreltronikConfiguration
```

Dzięki temu uzyskano migrację, w której znajdują się polecenia służące do utworzenia nowych tabel w bazie, określenia ich pól, typów danych w nich przechowywanych, a także nakładania na nie ograniczeń, w tym określenia kluczy głównych i obcych.

```
CreateTable(
    "dbo.Products",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        prod = c.String(),
        Name = c.String(),
        ReleaseDate = c.DateTime(nullable: false),
        Description = c.String(),
        CzyRGB = c.Boolean(nullable: false),
        MaxDPI = c.Int(nullable: false),
        ButtonCount = c.Int(nullable: false),
        IsWireless = c.Boolean(nullable: false),
        Cena = c.Decimal(nullable: false, precision: 18, scale: 2),
        Color_Id = c.Int(),
        Kategoria_Id = c.Int(),
        producent_Id = c.Int(),
        SwitchType_Id = c.Int(),
    })
.PrimaryKey(t => t.Id)
.ForeignKey("dbo.Colors", t => t.Color_Id)
.ForeignKey("dbo.Kategorias", t => t.Kategoria_Id)
.ForeignKey("dbo.Producents", t => t.producent_Id)
.ForeignKey("dbo.SwitchTypes", t => t.SwitchType_Id)
.Index(t => t.Color_Id)
.Index(t => t.Kategoria_Id)
.Index(t => t.producent_Id)
.Index(t => t.SwitchType_Id);
```

Przykładowe polecenie tworzenia tabeli „Products” w bazie, znajdujące się w utworzonej migracji.

Powyższe polecenie pochodzi z wstępnej wersji projektu i nie zawiera wszystkich wartości przechowywanych w końcowej wersji tabeli, ale istnieje możliwość wykonania kolejnych migracji dodających nowe kolumny w tabeli po zmianie klas obiektów. Wtedy w nowej migracji znajdzie się tylko polecenie modyfikujące istniejącą tabelę poprzez dodanie nowej kolumny. Przykładowo: w kolejnej migracji dodano tabelę z kodami rabatowymi oraz zmodyfikowano tabelę dbo.Products, aby do produktu można było przypisać kod rabatowy.

```
public override void Up()
{
    CreateTable(
        "dbo.DiscountCodes",
        c => new
        {
            DiscountCodeId = c.Int(nullable: false, identity: true),
            Code = c.String(),
            Discount = c.Double(nullable: false),
            DaysValid = c.Int(nullable: false),
            ValidUntil = c.DateTime(nullable: false),
        })
        .PrimaryKey(t => t.DiscountCodeId);

    AddColumn("dbo.Products", "DiscountCode_DiscountCodeId", c => c.Int());
    CreateIndex("dbo.Products", "DiscountCode_DiscountCodeId");
    AddForeignKey("dbo.Products", "DiscountCode_DiscountCodeId", "dbo.DiscountCodes", "DiscountCodeId");
}
```

Po utworzeniu tabeli dzięki migracjom można dodawać dane do nowej / zmodyfikowanej tabeli.

| | Id | Name | ReleaseDate | Description | CzyRGB | MaxDPI | ButtonCount | IsWireless | Cena | Color_Id | Kategoria_Id | produkt |
|--|----|--------------------|------------------|--------------------|--------|--------|-------------|------------|--------|----------|--------------|---------|
| | 3 | G502 Hero | 01.10.2018 00:00 | Mysz została <... | True | 25600 | 11 | False | 249,99 | 2 | 1 | 2 |
| | 5 | DeathAdder V2 | 22.09.2020 00:00 | Myszka bezprze... | True | 20000 | 8 | False | 663,65 | 2 | 1 | 6 |
| | 6 | G102 Lightsync | 01.05.2020 00:00 | Mysz wyposażo... | True | 8000 | 6 | False | 107,00 | 2 | 1 | 2 |
| | 7 | Mamba Elite | 01.08.2018 00:00 | Mamba Elite po... | True | 16000 | 9 | False | 251,00 | 2 | 1 | 6 |
| | 8 | X-748k | 01.08.2014 00:00 | Mysz A4Tech z s... | False | 2000 | 7 | False | 82,78 | 2 | 1 | 4 |
| | 9 | G Pro Wireless | 03.12.2020 00:00 | PRO Wireless to... | True | 25600 | 8 | True | 479,00 | 2 | 1 | 2 |
| | 10 | Playstation 5 D... | 18.06.2021 00:00 | Kultowy gamep... | False | 0 | 14 | True | 299,00 | 1 | 5 | 7 |
| | 11 | Xbox One Gam... | 22.11.2013 00:00 | Z przewodowy... | False | 0 | 13 | True | 388,68 | 2 | 5 | 5 |
| | 12 | Hard Gaming G... | 01.09.2014 00:00 | Szybkie i nagle... | False | 0 | 0 | True | 117,00 | 2 | 6 | 2 |
| | 13 | XGame X7-500... | 01.08.2014 00:00 | Duże rozmiary... | False | 0 | 0 | True | 27,90 | 2 | 6 | 4 |
| | 14 | Gigantus | 26.05.2020 00:00 | Podkładka stwo... | False | 0 | 0 | True | 111,09 | 2 | 6 | 6 |
| | 15 | G733 Lightspee... | 01.09.2020 00:00 | G733 jest bezpr... | True | 0 | 0 | True | 549,00 | 2 | 3 | 2 |
| | 16 | H560 Pro Surro... | 18.03.2021 00:00 | Wysokiej jakoś... | False | 0 | 0 | False | 265,99 | 2 | 3 | 1 |
| | 17 | G PRO X Gamin... | 01.08.2020 00:00 | Wbudowany mi... | False | 0 | 0 | False | 419,00 | 2 | 3 | 2 |
| | 18 | Cloud Alpha S | 01.03.2020 00:00 | Kultowy model... | False | 0 | 0 | False | 489,00 | 2 | 3 | 3 |
| | 19 | K70 RGB MK2 L... | 25.10.2018 00:00 | Nowa, smuka, l... | True | 0 | 113 | False | 649,00 | 2 | 2 | 1 |
| | 20 | Huntsman Mini | 14.07.2020 00:00 | Klawiatura idea... | True | 0 | 61 | False | 499,00 | 1 | 2 | 6 |
| | 21 | G413 Carbon | 01.04.2017 00:00 | Klawiatura jest... | False | 0 | 104 | False | 289,00 | 2 | 2 | 2 |
| | 22 | G213 Prodigy | 01.09.2016 00:00 | Logitech G213... | True | 0 | 113 | False | 239,00 | 2 | 2 | 2 |
| | 23 | Alloy Core RGB | 01.11.2018 00:00 | Klawiaturę Hyp... | True | 0 | 114 | False | 219,00 | 2 | 2 | 3 |
| | 24 | K95 RGB Platin... | 01.01.2017 00:00 | Interfejs USB ł... | True | 0 | 119 | False | 719,00 | 2 | 2 | 1 |
| | 25 | Alloy Origins C... | 22.02.2021 00:00 | HyperX Alloy O... | True | 0 | 87 | False | 389,00 | 2 | 2 | 3 |
| | 0 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |