



POLITECHNIKA ŚLĄSKA
Wydział Inżynierii Materiałowej

Podstawy baz danych

TEMAT:
AutoKomis – system zarządzający

Uwagi prowadzącego:

Data przyjęcia:

Imię i nazwisko:

1. Kostyra Mateusz
2. Bładziak Konrad
3. Czajkowski Michał
4. Kluczewski Stanisław

Podpis prowadzącego:

Imię i nazwisko opiekuna:
dr inż. Adam Duszeńko

Grupa dziekańska: IP20pp

Sekcja laboratoryjna: 3

1. Opis projektu

Projekt pozwala na zarządzaniem komis. Wyświetla aktualnie zatrudnionych pracowników, pozwala utworzyć nowych, edytować albo ich usuwać. Dzięki tej bazie pracownicy mogą wyszukiwać samochodów dla swoich klientów o odpowiednich parametrach, tworzyć umowy czy aktualizować pojazdy.

2. Technologia:

- SQLite, MSSQL
- C#
- MVC ASP .NET
- HTML 5
- CSS
- jQuery
- Bootstrap

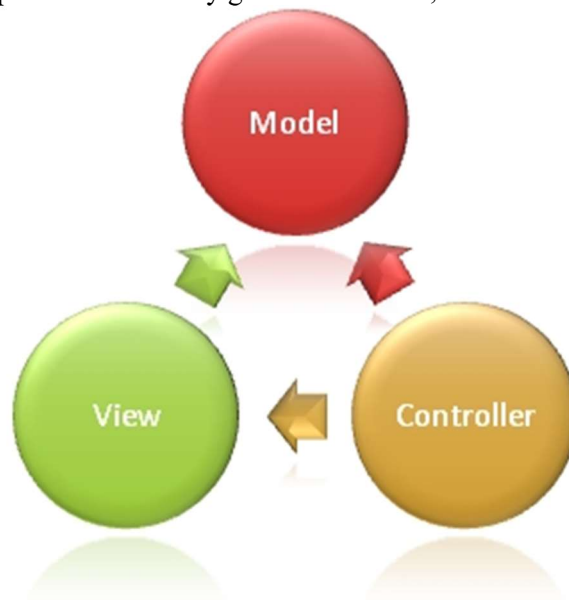
W projekcie wykorzystywane są następujące zagadnienia:

MVC - Wzorzec MVC

Wzorzec architektury Model-View-Controller (MVC) dzieli aplikację na trzy główne grupy składników: modele, widoki i kontrolery.

Ten wzorzec pomaga w rozdzieleniu problemów. Korzystając z tego wzorca, żądania użytkowników są kierowane do kontrolera, który jest odpowiedzialny za pracę z modelem w celu wykonywania akcji użytkownika i/lub pobierania wyników zapytań. Kontroler wybiera widok do wyświetlenia użytkownikowi i dostarcza do niego wszystkie dane modelu, których potrzebuje.

Na poniższym diagramie przedstawiono trzy główne składniki, które odwołują się do innych:



- **Modele.** Obiekty modelu to części aplikacji, które implementują logikę dla domeny danych aplikacji. Często obiekty modelu pobierają i przechowują stan modelu w bazie danych. Na przykład obiekt Product może pobierać informacje z bazy danych, działać na nim, a następnie zapisywać zaktualizowane informacje z powrotem do tabeli Products w SQL Server. W małych aplikacjach model jest często separacją koncepcyjną, a nie fizyczną. Jeśli na przykład aplikacja odczytuje tylko zestaw danych i wysyła go do widoku, aplikacja nie ma warstwy modelu fizycznego ani skojarzonych klas. W takim przypadku zestaw danych przejmuje rolę obiektu modelu.

- Widoki. Widoki to składniki, które wyświetlają interfejs użytkownika aplikacji. Zazwyczaj ten interfejs użytkownika jest tworzony na podstawie danych modelu. Przykładem może być widok edycji tabeli Products, który wyświetla pola tekstowe, listy rozwijane i pola wyboru na podstawie bieżącego stanu obiektu Products.
- Kontrolery. Kontrolery to składniki, które obsługują interakcję z użytkownikiem, pracują z modelem i ostatecznie wybierają widok do renderowania, który wyświetla interfejs użytkownika. W aplikacji MVC w widoku są wyświetlane tylko informacje; Kontroler obsługuje dane wejściowe i interakcje użytkownika oraz reaguje na nie. Na przykład kontroler obsługuje wartości ciągu zapytania i przekazuje te wartości do modelu, który z kolei wysyła zapytanie do bazy danych przy użyciu wartości .

CRUD - skrótowiec CRUD może być stosowany w odniesieniu do interfejsu użytkownika większości aplikacji, które zazwyczaj pozwalają użytkownikowi na:

utworzenie lub dodanie nowych informacji (CREATE)
 odczytanie lub wyświetlenie istniejących informacji (READ)
 modyfikowanie lub edycję istniejących informacji (UPDATE)
 usuwanie istniejących informacji (DELETE)

DTO – Data Transfer Object jest to tak jak nazwa wskazuje obiekt tworzony do przesyłania danych pomiędzy aplikacjami. Głównym celem wzorca jest zmniejszenie liczby połączeń w obie strony do serwera poprzez grupowanie wielu parametrów w jednym wywołaniu. Znacząco zmniejsza to obciążenie sieci w zdalnych operacjach. DTO dzielimy na request oraz response czyli zapytanie i odpowiedź. Przykładowa implementacja DTO w naszej bazie danych:

Request:

```

1  namespace AutoKomis.DTO.Request
2  {
3      0 references
4      public class AutoRequest
5      {
6          0 references
7          public int IdSamochodu { get; set; }
8          0 references
9          public int IdKomisu { get; set; }
10         0 references
11         public int IdSprzedazy { get; set; }
12     }
13 }

```

Response:

```
1 using System.ComponentModel;
2
3 namespace AutoKomis.DTO.Response
4 {
5     4 references
6     public class AutoResponse
7     {
8         [DisplayName("Marka")]
9         2 references
10        public string Marka { get; set; }
11
12        [DisplayName("Model")]
13        2 references
14        public string Model { get; set; }
15
16        [DisplayName("Rok produkcji")]
17        2 references
18        public string RokProdukcji { get; set; }
19
20        public string Vin { get; set; }
21        2 references
22        public string Przebieg { get; set; }
23        2 references
24        public string Cena { get; set; }
25        2 references
26        public string Kolor { get; set; }
27
28        [DisplayName("Pojemność Silnika")]
29        2 references
30        public double PojemnoscSilnika { get; set; }
31
32        [DisplayName("Typ Nadwozia")]
33        2 references
34        public string TypNadwozia { get; set; }
35
36        [DisplayName("Rodzaj Paliwa")]
37        2 references
38        public string RodzajPaliwa { get; set; }
39        2 references
40        public char Segment { get; set; }
41
42        [DisplayName("Ilość Drzwi")]
43        2 references
44        public int IloscDrzwi { get; set; }
45
46        [DisplayName("Czy Dostępny")]
47        2 references
48        public bool CzyDostepny { get; set; }
49    }
50 }
```

Migracja – jest to proces, który pozwala na bezkonfliktowe tworzenie i modyfikacje struktury bazy danych.

Encja – w bazach danych jest to reprezentacja obiektu (grupy obiektów). Podstawową cechą encji jest to, że jest rozróżnialna od innych encji, zgodnie z założeniem modelu relacyjnego baz danych.

LINQ – jest to rozszerzenie języka C# umożliwiające zadawanie pytań na obiektach. Za pomocą tych samych metod operujemy na różnych obiektach z wielu typów źródeł. Funkcje tego narzędzia wykorzystaliśmy głównie do pisania kwerend podczas pisania funkcjonalności.

Przykład wykorzystania LINQ:

```
[HttpGet("SzukajDostepnychSamochodow")]
0 references
public IActionResult SzukajDostepnychSamochodow()
{
    var lista = _db.Auta.FirstOrDefault(x => x.CzyDostepny == true);

    if (lista == null)
    {
        return NotFound();
    }

    var result = from auto in _db.Auta
                  where auto.CzyDostepny == lista.CzyDostepny
                  select auto;

    return View(result);
}
```

DB Context - Klasa DbContext jest integralną częścią Entity Framework. Instancja DbContext reprezentuje sesję z bazą danych, która może być użyta do zapytań oraz poleceń do bazy danych.

DbContext pozwala na wykonywanie następujących zadań:

- Zarządzanie połączeniem z bazą danych;
- Konfigurację modelu i relacji;
- Wykonywanie zapytań na bazie danych;
- Zapisywanie danych w bazie danych;
- Konfigurację ustawień związanych ze śledzeniem zmian;
- Buforowanie danych;
- Zarządzanie transakcjami.

Przykładowe wykorzystanie DbContext'u w kontrolerach w naszej bazie danych

```
using AutoKomis.Data;
using AutoKomis.DTO.Response;
using AutoKomis.Models;
using Microsoft.AspNetCore.Mvc;

namespace AutoKomis.Controllers
{
    1 reference
    public class AutoController : Controller
    {
        private readonly ApplicationDbContext _db;

        0 references
        public AutoController(ApplicationDbContext db)
        {
            _db = db;
        }

        0 references
        public IActionResult Index()
        {
            var lista = _db.Auta.ToList();
            return View(lista);
        }
    }
}
```

(Wstrzykiwanie zależności DbContextu):

```
12 references
public class ApplicationDbContext : DbContext
{
    0 references
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
    {
    }
    0 references
}
```

OnModelCreating(Konfiguracja Modelu) -. Jest to najbardziej zaawansowana metoda konfiguracji, która pozwala określić konfigurację bez konieczności modyfikowania klas jednostek

```
protected override void OnModelCreating(ModelBuilder builder)
{
    builder.Entity<Komis>()
        .HasKey(x => x.Id);

    builder.Entity<Pracownik>()
        .HasKey(x => x.IdPracownika);

    builder.Entity<Pracownik>()
        .HasOne(x => x.Komis)
        .WithMany(x => x.Pracownicy)
        .HasForeignKey(x => x.IdKomisu);

    builder.Entity<Auto>()
        .HasKey(x => x.IdSamochodu);

    builder.Entity<Auto>()
        .HasOne(x => x.Sprzedaz)
        .WithMany(x => x.Auta)
        .HasForeignKey(x => x.IdSprzedazy);
}
```

```
builder.Entity<Auto>()
    .HasOne(x => x.Komis)
    .WithMany(x => x.Auta)
    .HasForeignKey(x => x.IdKomisu);

builder.Entity<Sprzedaz>()
    .HasKey(x => x.IdSprzedazy);

builder.Entity<Sprzedaz>()
    .HasOne(x => x.Klient)
    .WithMany(x => x.Sprzedaze)
    .HasForeignKey(x => x.IdKlienta);

builder.Entity<Klient>()
    .HasKey(x => x.IdKlienta);

base.OnModelCreating(builder);
}
```


DbSet – Klasa, która reprezentuje zestaw jednostek, który może służyć do wykonywania operacji CRUD (CREATE, READ, UPDATE, DELETE). Klasa kontekstu (pochodząca z DbContext) musi zawierać właściwości typu DbSet dla jednostek, które mapują do tabel i widoków bazy danych.

Użycie DbSet'u w naszej bazie danych:

```
0 references
public DbSet<Komis> Komis { get; set; }
10 references
public DbSet<Pracownik> Pracownicy { get; set; }
8 references
public DbSet<Sprzedaz> Sprzedaze { get; set; }
0 references
public DbSet<Klient> Klienci { get; set; }
11 references
public DbSet<Auto>Auta { get; set; }
```

3. Wymagania:

- Komunikacja z bazą danych
- Wyświetlanie aktualnych pracowników i samochodów
- Dodawanie, modyfikacja, usuwanie pracowników oraz samochodów
- Wyszukiwanie samochodów podając odpowiednie parametry
- Wyszukiwanie pracowników po danej funkcji
- Przejrzysta oraz czytelne SPA(Single Page Application)

4. Baza danych i tabele

Komis	Pracownicy	Klienci	Auto	Sprzedaże
IdKomisu	IdPracownika	IdKlienta	IdSamochodu	IdSprzedaży
Nazwa			IdKomisu	
Miasto			IdSprzedaży	IdSamochodu
Ulica	Marka			
Kod pocztowy	Imię	Imię	Model	IdPracownika
KRS			Rok produkcji	
NIP	Nazwisko	Nazwisko	Vin	IdKlienta
Imię właściciela			Przebieg	
Nazwisko Właściciela	Stanowisko	Numer dowodu	Cena	Data zawarcia umowy
	Numer dowodu		Kolor	Kwoty
Pojemność silnika				
Typ nadwozia				
Rodzaj paliwa				
Segment				
Ilość drzwi				
Czy dostępny				

5. Schemat bazy danych

Odwzorowanie stworzenia poszczególnych tabel w MSSQL -> Dzięki Entity Framework schemat Tabel, zależności oraz Baza Danych jest wygenerowana po użyciu migracji automatycznie

```
CREATE TABLE Komis(  
    ID int NOT NULL,  
    Nazwa varchar(MAX) NOT NULL,  
    Miasto varchar(MAX),  
    Ulica varchar(MAX),  
    KodPocztowy varchar(6),  
    KRS varchar(10),  
    NIP varchar(10),  
    ImieWlasciciela varchar(MAX),  
    NazwiskoWlasciciela varchar(MAX),  
    KosztyUtrzymaniaMiesieczne decimal,  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE Pracownik(  
    IdPracownika int NOT NULL PRIMARY KEY,  
    IdKomsiu int NOT NULL FOREIGN KEY REFERENCES Komis(ID),  
    Imie varchar(MAX),  
    Nazwisko varchar(MAX) NOT NULL,  
    Ulica varchar(MAX),  
    Stanowisko varchar(50),  
    NumerDowodu varchar(9),  
);
```

```
use AutoKomis;
```

```
CREATE TABLE Auto (  
    IdSamochodu int NOT NULL PRIMARY KEY,  
    IdSprzedazy int FOREIGN KEY REFERENCES Sprzedaz(IdSprzedazy),  
    IdKomisu int FOREIGN KEY REFERENCES Komis(ID),  
    Marka varchar(50),  
    Model varchar(50),  
    RokProdukcji varchar(4),  
    Vin varchar(30),  
    Przebieg varchar(7),  
    Cena decimal,  
    Kolor varchar(50),  
    PojemnoscSilnika float,  
    TypNadwozia varchar(20),  
    RodzajPaliwa varchar(20),  
    Segment char(1),  
    IloscDrzwi char(2),  
    CzyDostepny bit,  
);
```

```

CREATE TABLE [dbo].[Sprzedaz](
    [IdSprzedazy] [int] NOT NULL,
    [DataZawarciaUmowy] [datetime] NOT NULL,
    [Kwota] [decimal](18, 0) NULL,
PRIMARY KEY CLUSTERED
(
    [IdSprzedazy] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Klient](
    [IdKlienta] [int] NOT NULL,
    [Imie] [varchar](max) NULL,
    [Nazwisko] [varchar](max) NOT NULL,
    [NumerDowodu] [varchar](9) NULL,
    [IdSprzedazy] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [IdKlienta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Klient] WITH CHECK ADD FOREIGN KEY([IdKlienta])
REFERENCES [dbo].[Sprzedaz] ([IdSprzedazy])
GO

ALTER TABLE [dbo].[Klient] WITH CHECK ADD FOREIGN KEY([IdSprzedazy])
REFERENCES [dbo].[Sprzedaz] ([IdSprzedazy])
GO

```

6. Schemat aplikacji

Strona startowa:

AutoKomis
Home
Pracownicy
Sprzedaż
Auta

Witamy w AutoKomis



Na samochodach znamy się naprawdę dobrze. Od lat śledzimy trendy i zmiany w motoryzacji. Wiemy, jak ważne są sprawne cztery koła podczas codziennej eksploatacji – zarówno na trasach długich, jak i krótkich. Działamy na rynku od 2015 roku. Za nami wiele udanych transakcji i zadowolonych klientów. W naszym serwisie można znaleźć ogłoszenia sprzedaży nowych i używanych pojazdów – różne marki, modele i roczniki. U nas możesz nie tylko kupić samochód ale również wystawić swój lub zostawić go w rozliczeniu i cieszyć się nowszym samochodem.

Ul. Krasińskiego 8
40-019 Katowice
KRS: 0003133154
NIP: 9623165183
















Widoki poszczególnych kontrolerów:

AutoKomis Home Pracownicy Sprzedaż Auta

Pracownicy

Szukaj stanowiska:
Sprzedawca Szukaj

Dodaj pracownika:


IdPracownika	Imie	Nazwisko	Stanowisko	Modyfikuj
1	Mateusz	Kostyra	Kierownik	 
2	Konrad	Bładziak	Kierownik	 
3	Czajkowski	Michał	Sprzedawca	 
4	Alicja	Kwaitek	Księgowa	 
5	Tomasz	Górski	Detaler	 
7	Adrian	Grzyb	Technik	 

AutoKomis Home Pracownicy Sprzedaż Auta

Pracownicy o danym stanowisku:

Powrót do Indexu

IdPracownika	Imie	Nazwisko	Stanowisko
4	Alicja	Kwaitek	Księgowa

AutoKomis Home Pracownicy Sprzedaż Auta

Dodaj Pracownika

Imie

Nazwisko

Stanowisko

IdKomisu

Numer Dowodu

Create Back

AutoKomisHomePracownicySprzedażeAuta

Sprzedaże

Szukaj sprzedaży:

dd.mm.rrrrdd.mm.rrrrSzukaj

Dodaj sprzedaż:

IdSprzedaży	IdSamochodu	IdPracownika	IdKlienta	Data zawarcia umowy	Kwota	Modify
6	0	0	0	16.04.2022 16:33:00	0,0	<div><div></div><div></div></div>
7	0	0	0	01.01.0001 00:00:00	0,0	<div><div></div><div></div></div>
8	6	1	6142	06.04.2022 16:39:00	1141,0	<div><div></div><div></div></div>
9	4	1	13	03.05.2021 16:39:00	121,0	<div><div></div><div></div></div>
12	9	1	1	23.04.2022 11:11:00	1234,0	<div><div></div><div></div></div>

AutoKomisHomePracownicySprzedażeAuta

Sprzedaże w danych latach:

Powrót do Indexu

IdSprzedaży	IdSamochodu	IdPracownika	IdKlienta	Data zawarcia umowy	Kwota
6	0	0	0	16.04.2022 16:33:00	0,0
8	6	1	6142	06.04.2022 16:39:00	1141,0
12	9	1	1	23.04.2022 11:11:00	1234,0

AutoKomisHomePracownicySprzedażeAuta

Dodaj Sprzedaż

IdSamochodu

IdPracownika

IdKlienta

DataZawarciaUmowy

dd.mm.rrrr--:--

Kwota

Create

Back

AutoKomis

HomePracownicySprzedajeAuta

Znajdź Auto:

Benzyna

Szukaj

Sprawdź dostępne auta

Sprawdź

Dodaj auto:

Marka	Model	RokProdukcji	Paliwo	Vin	Przebieg	Cena	Czy Dostępny	Modyfikuj
Honda	Acord	1999	LPG	SHHCG9690YU031900	342500	12500	True	
BMW	M2	2018	Benzyna	WBAAVG6904603A900	42000	248000	True	
Ford	Mondeo	2003	Diesel	AS124ASFA9875AS	199 000	10 000	True	
Kia	Soul	2011	Diesel	SA397ASF55845A8	199 000	20 000	True	
Toyota	Corolla	2006	Diesel	UT2N412KDSF3	199 000	15 000	True	
BMW	E39	1999	Benzyna	BA412FA125F22	250 000	10 000	False	
BMW	E36	2001	LPG	ZS91155A5FA33Z5	220 000	7000	False	
Swift	Suzuki	2009	Benzyna	25hhxbas99313afd	173950	12900	True	

AutoKomis

HomePracownicySprzedajeAuta

Auta o danych parametrach:

Powrót do Indexu

Marka	Model	RokProdukcji	Vin	Przebieg	Cena	Kolor	Pojemność Silnika	Typ Nadwozia	Rodzaj Paliwa	Segment	Ilość drzwi	Czy Dostępny
Honda	Acord	1999	SHHCG9690YU031900	342500	12500	Srebrny	2	coupe	LPG	C	2	True

AutoKomis

HomePracownicySprzedajeAuta

Dostępne samochody:

Powrót do Indexu

Marka	Model	RokProdukcji	Vin	Przebieg	Cena	Kolor	Pojemność Silnika	Typ Nadwozia	Rodzaj Paliwa	Segment	Ilość drzwi	Czy Dostępny
Honda	Acord	1999	SHHCG9690YU031900	342500	12500	Srebrny	2	coupe	LPG	C	2	True
BMW	M2	2018	WBAAVG6904603A900	42000	248000	Czarny	3	coupe	Benzyna	C	2	True
Ford	Mondeo	2003	AS124ASFA9875AS	199 000	10 000	czarny	0	sedan	Diesel	C	5	True
Kia	Soul	2011	SA397ASF55845A8	199 000	20 000	Srebrny	1,6	hatchback	Diesel	B	5	True
Toyota	Corolla	2006	UT2N412KDSF3	199 000	15 000	Srebrny	0	kombi	Diesel	C	5	True
Swift	Suzuki	2009	25hhxbas99313afd	173950	12900	Żółty	0	Kombi	Benzyna	C	5	True

AutoKomis

HomePracownicySprzedajeAuta

Dodaj Auto

IdKomisu

Rok produkcji

Cena

IdSprzedazy

Rodzaj Paliwa

Kolor

IdSamochodu

Typ Nadwozia

Pojemność Silnika

Model

Vin

Segment

Marka

Przebieg

Ilość Drzwi

Czy Dostępny

☒

Create

Back

7. Diagram ERD

