

**Технически Университет Варна – Катедра ФИТА Специалност СИТ:**

Курсов проект

На тема:

Продажба на автомобили

Дисциплина:

Системи с бази от данни

Изготвил: Виктор Красенов Драгоев

Факултетни №: 17621640

Специалност: СИТ

Курс: 3 Група: 2а

1. ЗАДАНИЕ

Да се разработи система за работа с продажба на автомобили.

Програмата съхранява и обработва данни за автомобили, клиенти, оператори и компании. Системата позволява множествен достъп.

Системата поддържа достъп от различни видове потребители, които имат различен достъп до функционалнистите и.

Администраторски акаунт - трябва да може да добавя нови акаунти за оператори (който продават коли) , да редактира съществуващите акаунти и да ги изтрива.

Администраторкия акаунт също трябва да може да добавя нови : компании , цветове на автомобили и клас на автомобили.

Потребителси акаунт - трябва да може да добавя нови автомобили по следните параметри: модел на автомобил ,състояние на автомобил,цена на автомобил, наличност , тип на двигателя , текущи километри , клас на автомобила и цвят на автомобила.

Потребителския акаунт също трябва да може да може да:

- добавя клиенти по следните параметри: име на клиент , адрес на клиент и номер на лична карта.

-да продава коли на съществуващ клиент в системата и възможност за филтриране на оператор , клиент и марка от наличните .Също така и избор на дата на продажбата.

-възможност за преглед на продадените коли.

Справки за :

-Справка за коли продадени от конкретен оператор.

-Справка за коли от конкретна фирма.

Операции за работа със системата:

Влизане в системата от оператори , който продават автомобили.

Влизане в състематра от администратори - имат вскички функционалности

Добавяне, променяне, търсене и изтриване на:

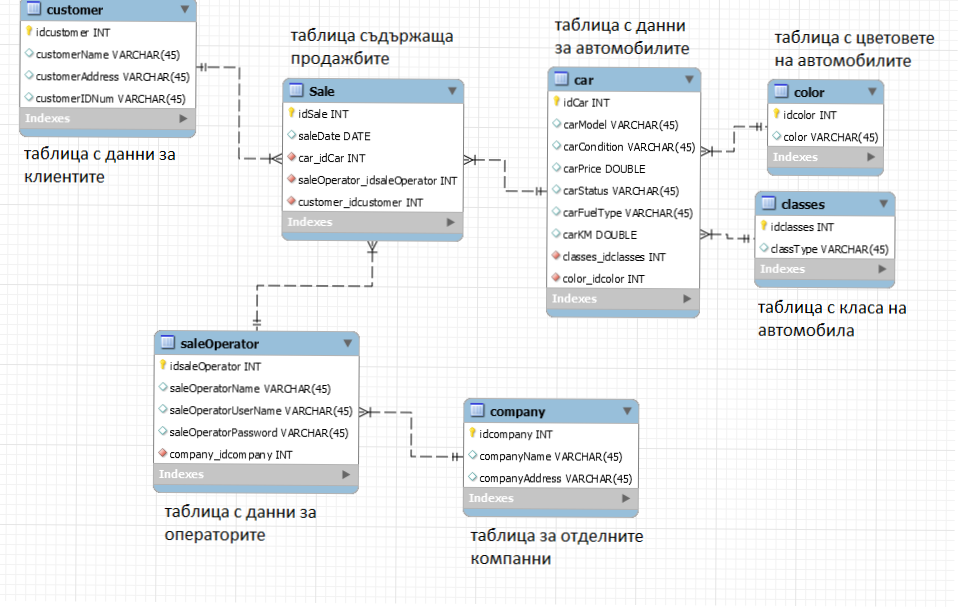
1. Работна позиция
2. Потребител
3. Клиент
4. Компания
5. Автомобил
6. Назначаване на потребител на длъжност

Справки за:

Автомобили отдадени от представителство

Автомобили отдадени от потребител

1. Модел на БД с пояснения

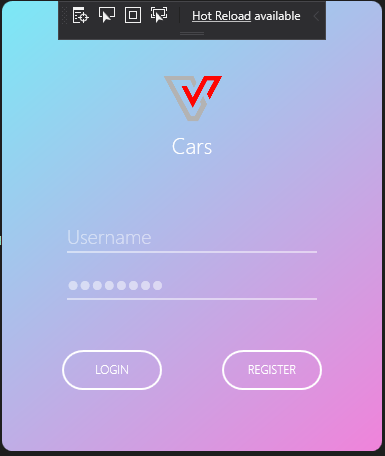


1. Описание на избраните средства за резализация на проекта

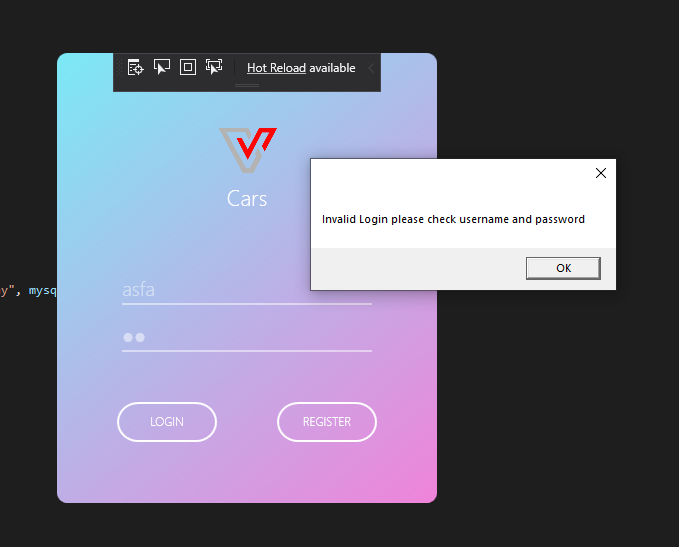
Условно можем да разделим заданието на две.

1. База данни : За реализацията на базата данни съм използвал MySQL workbench най-вече, защото е безплатно приложение и също така работи доста добре. Процедурите се пишат лесно и извикването им става също толкова лесно.
2. Графичен интерфейс и бизнес логика – за графичния интерфейс съм използвал Windows Presentation Foundation (WPF) UI framework понеже съвременния дизайн на интерфейса се прави с него и предлага най-много възможности.Също така исках да пробвам нещо различно от javaFX . Бизнес логиката е написана на C# като от него най-вече се извикват отделните процедури написани в MySQL
3. Ръководство на потребителя

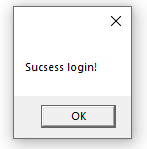
При първоначалното стартиране на програмата от потребителя се изисква да въведе потребителско име и парола за да влезе в системата.



Ако потребителя въведе грешно име или парола излиза съобщение за грешка което го подканва да опита отново.



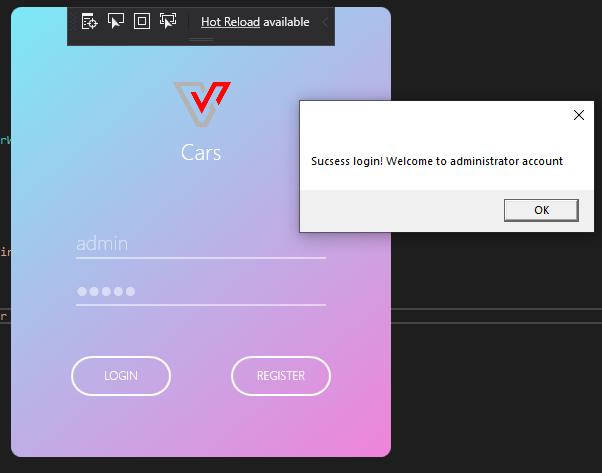
При правилно въведени потребителско име и парола потребителя се уведомява за това и му се дава достъп до функционалностите на програмата.



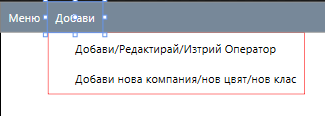
**Администраторски акаунт**

Опции от менюто на администраторски акаунт:

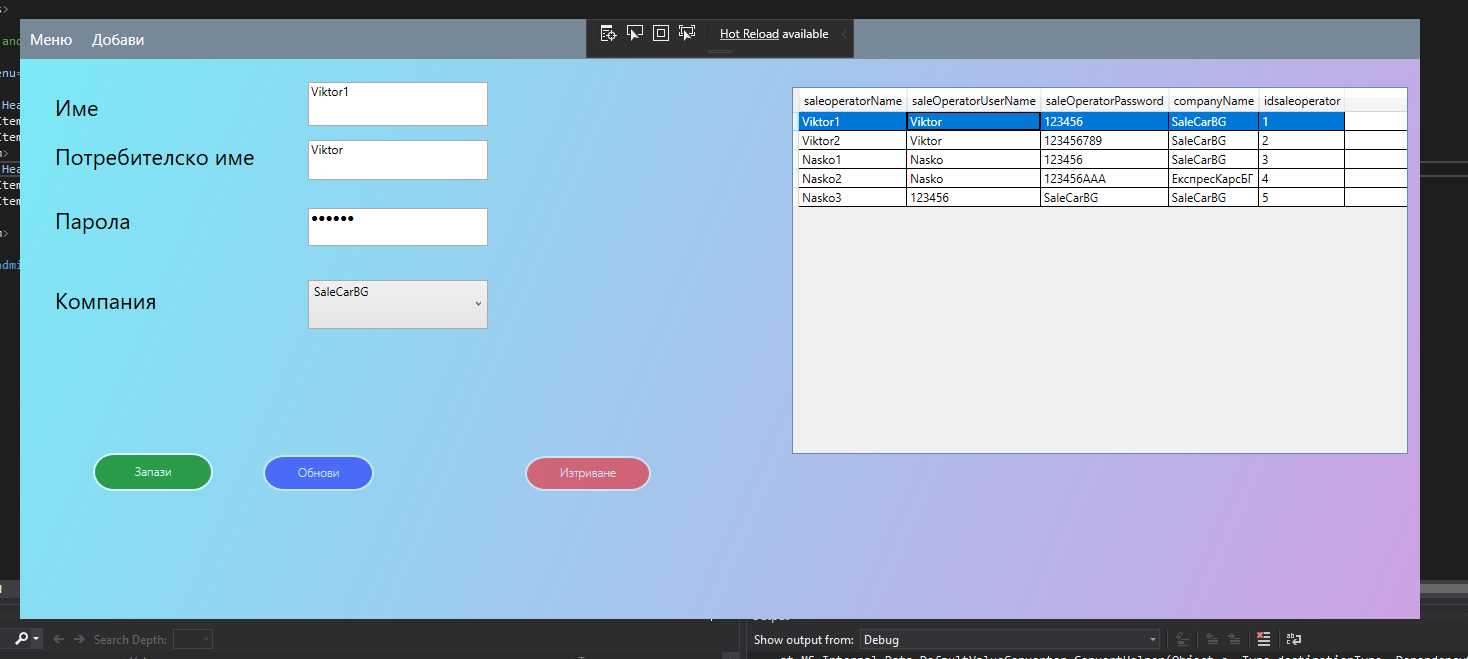
За да влезете в администраторкия акаунт трябва да напишете потребителско име-admin и парола -admin.По този начин системата знае, че влиза администратор и се отвааря специален панел само за администратори , който не може да бъде достъпен от операторките акаунти.



**Добавяне, редактиране,изтриване на операторски акаунт.**



От меню добави се избира Добави/Редактирай/Изтрий Оператор и се отваря следния фрагмент:

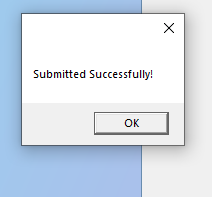


От където с двойно кликане върху оператора в таблицата данните му се зареждат в контролите и могат да бъдат редактирани след което се натиска бутон Обнови.

При назначаване на нов оператор контролите са празни и след попълването им се натиска бутон запис.

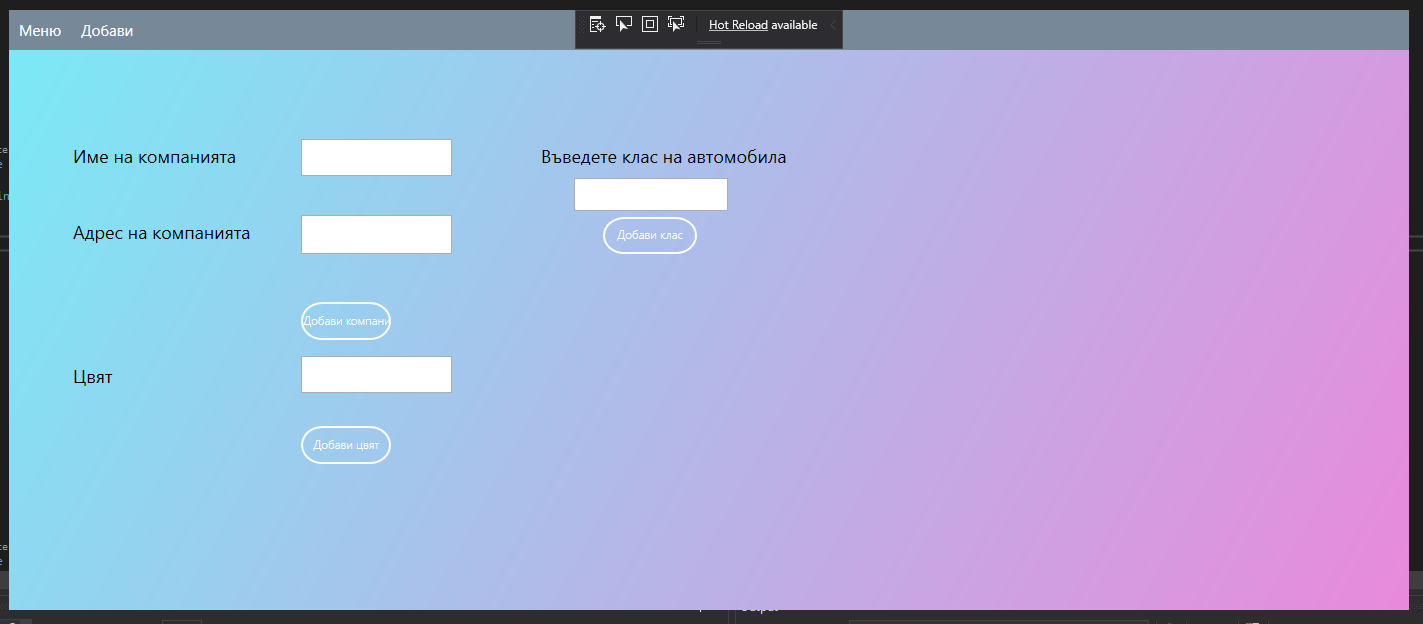
При изтриване на запис се избира от таблицата съответния оператор и се натиска бутон изтриване.

При всяка една операция излиза уведомение дали опрацията във базата данни е завършила успешо или не .



**Добавяне, редактиране,изтриване на операторски акаунт.**

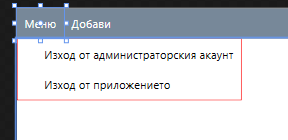
От меню добави се избира Добави нова компания / нов цвят / нов клас и се отваря следния фрагмент:



Където за добавяне се попълват контролите и се натиска съотвения бутон.

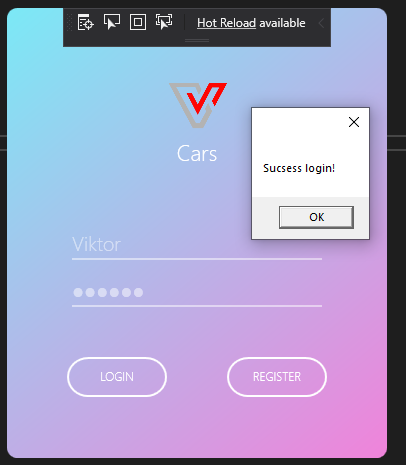
При всяка една операция излиза уведомение дали опрацията във базата данни е завършила успешо или не .

**За изход от администраторския акаунт** се избира Меню ->изход от администраторски акаунт или-> изход от приложението.

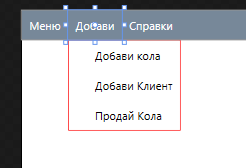


**Операторски акаунт**

За да влезете в акаунт на оператор първо трябва администратор да ви назначи и да ви направи акаунт . След се използват данните за влизане в операторския акаунт.

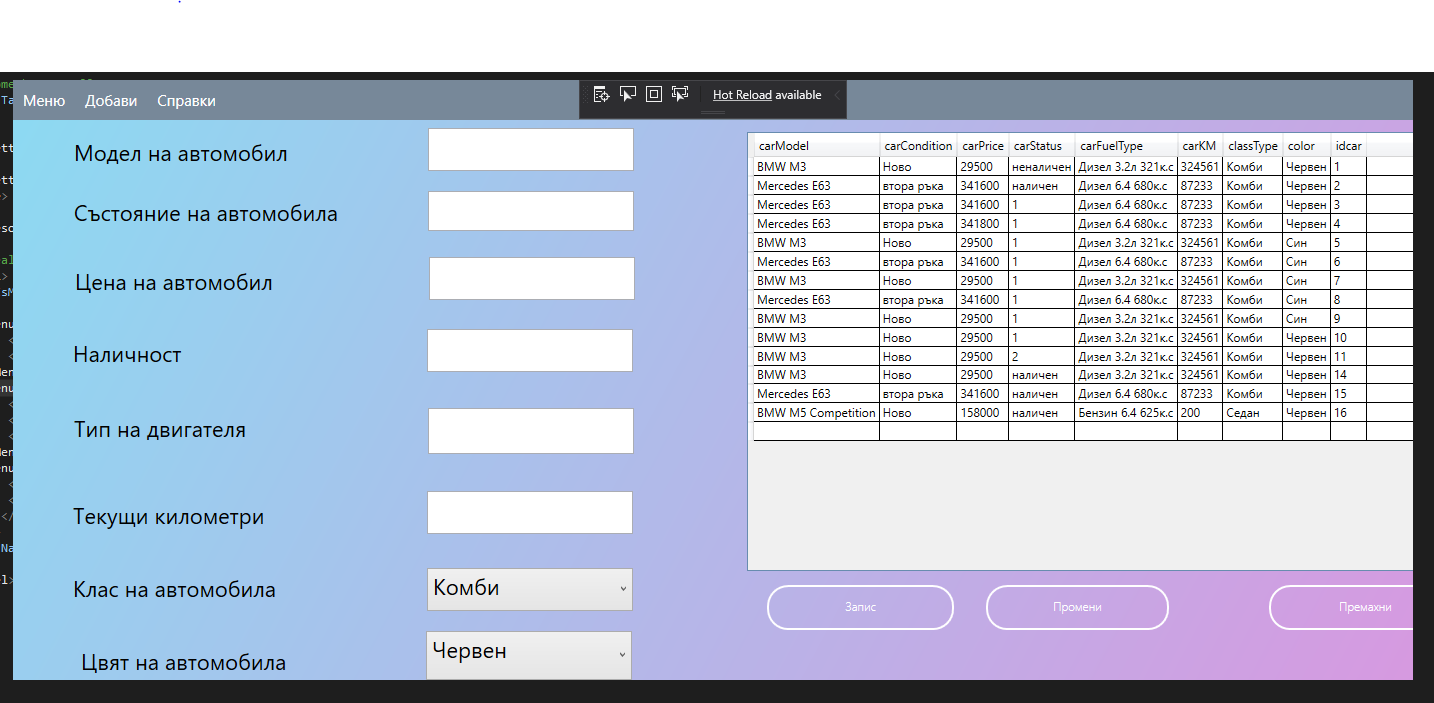


**меню Добави**



Тук може да изберете дали да добавяте кола , клиент , или да продавате кола.

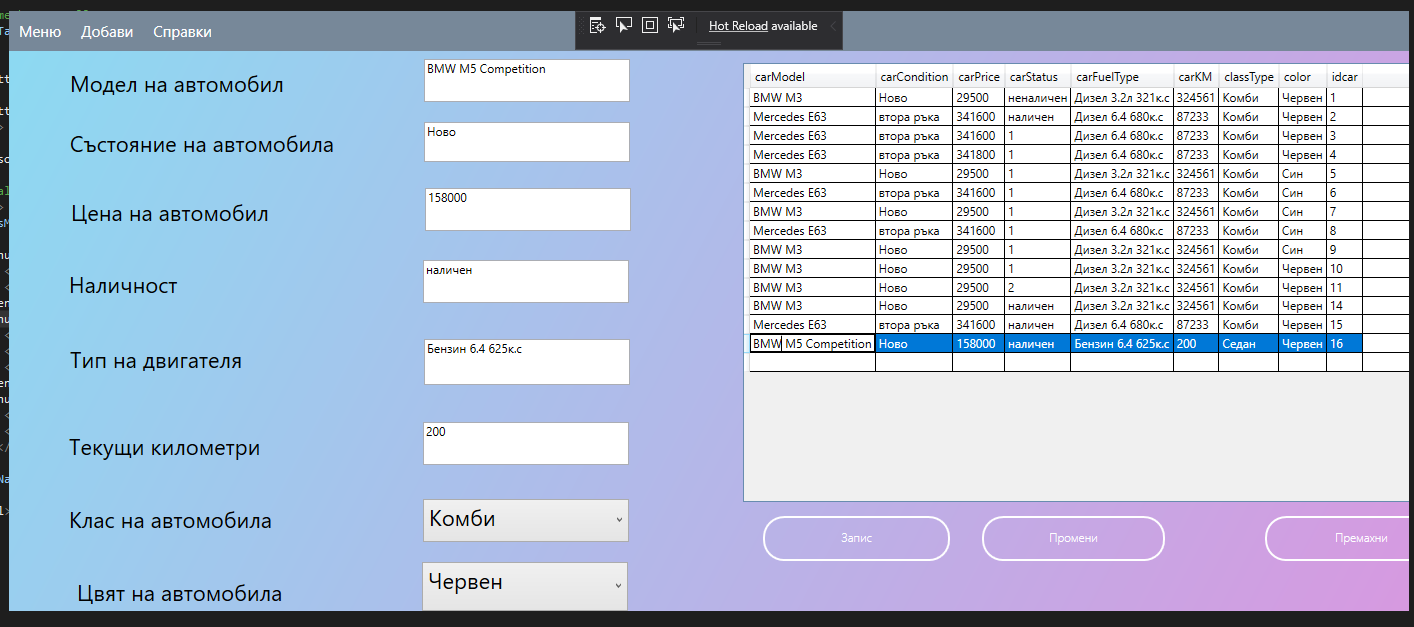
**Добави кола**



При попълване на контролите и натискане на бутон запис се добавя нова кола.

Като опците за клас и звят могат да бъдат добавяни само от администратор и избирани от комбобокс тук.

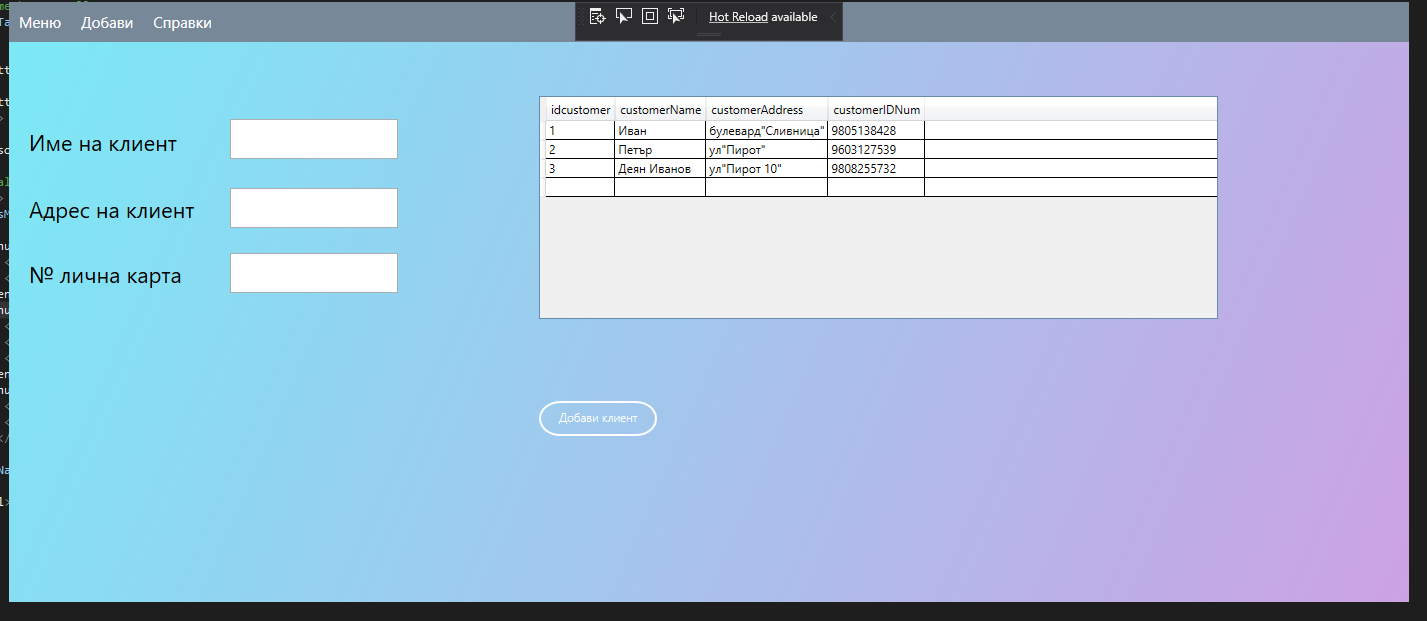
При двойно кликане върху съществуваща кола от таблицата в дясно данните и се попълват автоматично в контролите и по този начин могат да бъдат редактирани с бутон редактирай.



За изтриване на неналична кола се избира от таблицата в дясно и се натиска бутон Премахни.

При всяка една операция излиза уведомение дали опрацията във базата данни е завършила успешо или не .

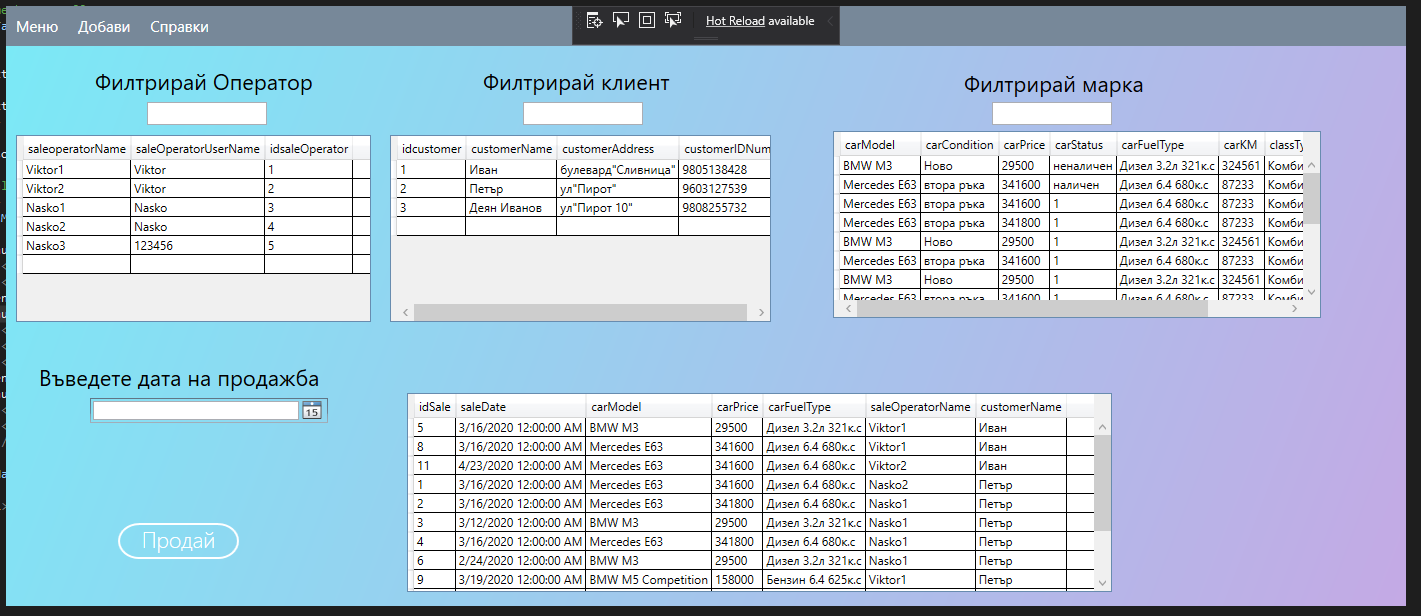
**Добави Киент**



При попълване на контролите и натискане на бутон добави клиент се добавя клиент в таблицата.

След операцията излиза уведомение дали опрацията във базата данни е завършила успешо или не .

**Продай кола**



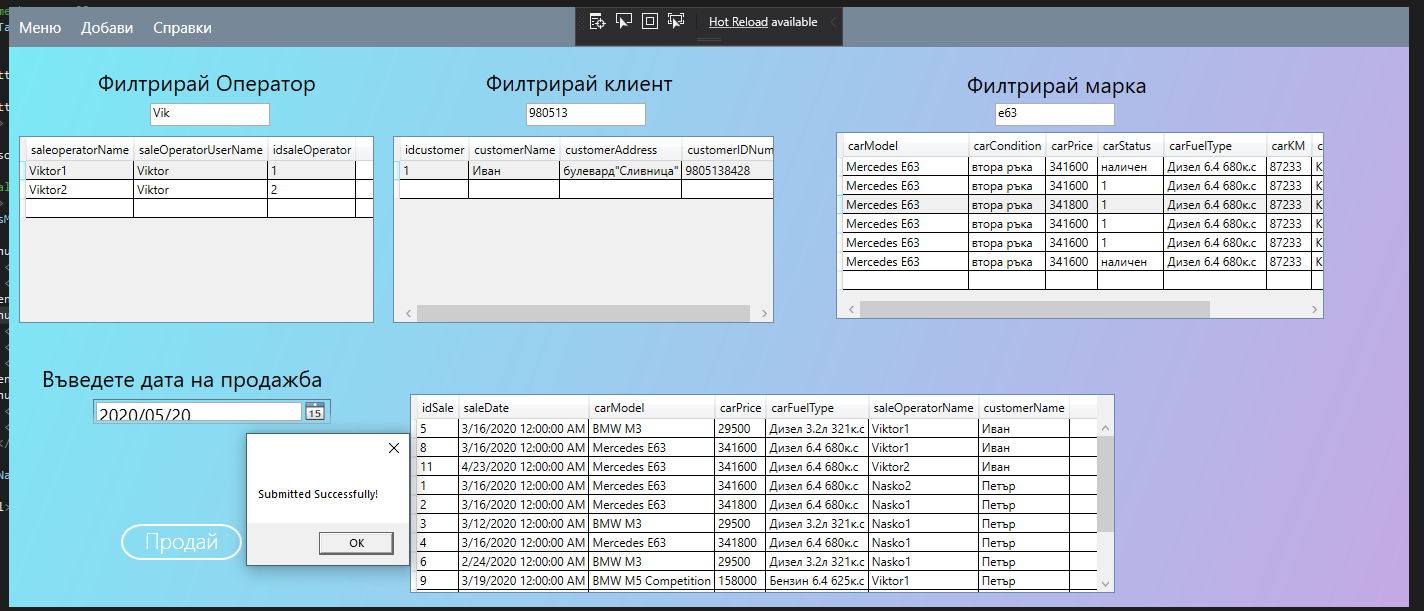
Тук се осъществява продаването на кола от оператор на клиент.

От първата таблица се избира оператора който продава колата след това от втората таблица се избира клиента (който трябва да е вече регистриран в предното меню), след това се избира кола от 3тата таблица и дата от полето за избиране на дата .

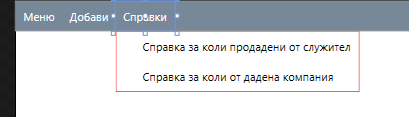
След натискане на бутон Проддай излиза уведомение дали опрацията във базата данни е завършила успешо или не .

Ако операцията е успешна продажбата излиза в таблицата в дясно .

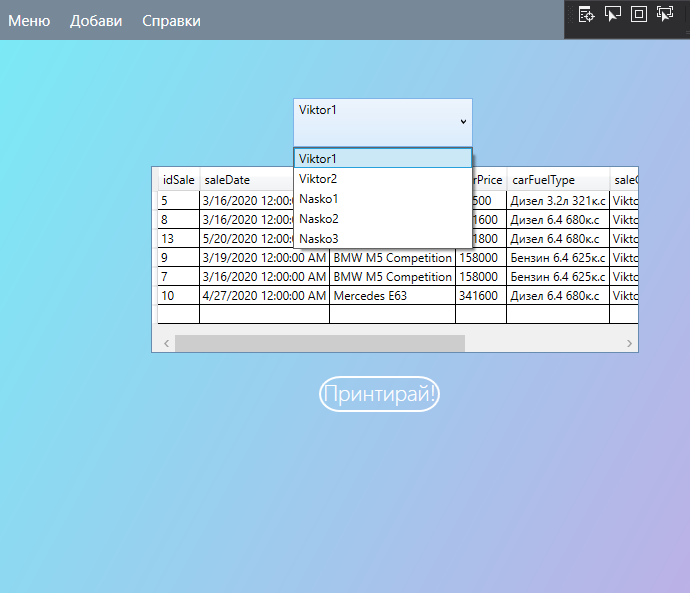
Също така може да се филтрират даните динамично.



**Справки**

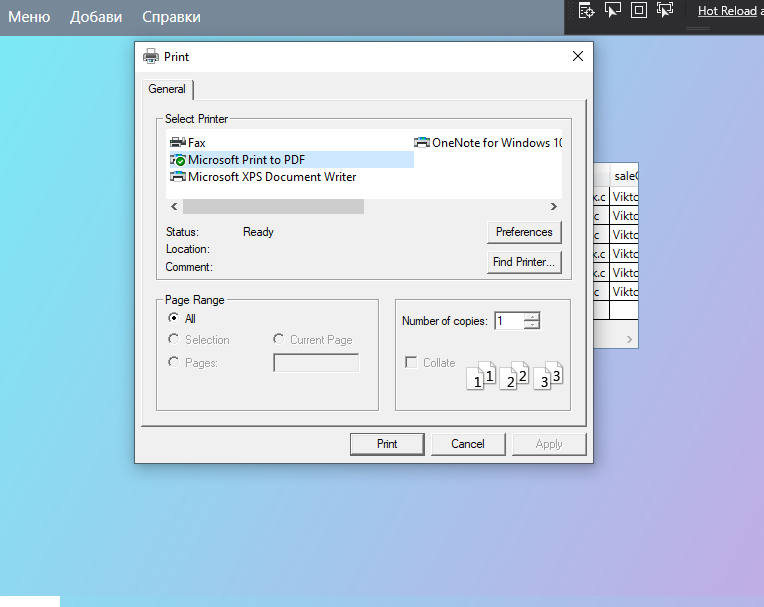


**Справка за коли продадени от оператор:**

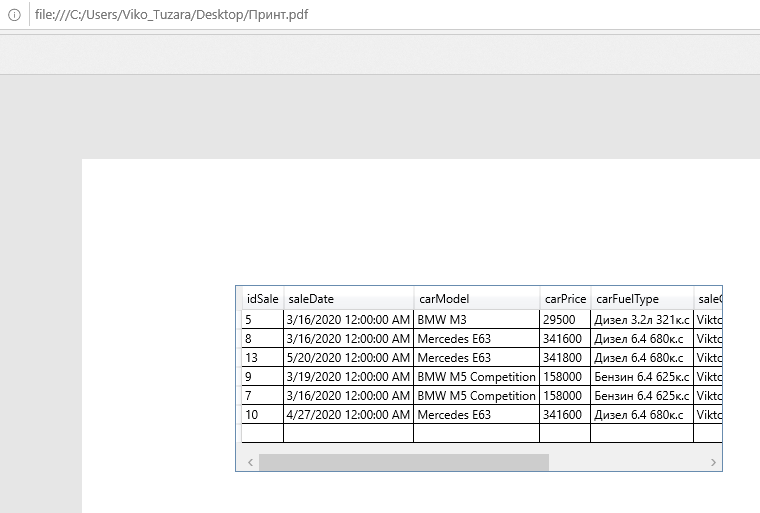


От комбобокса се избира оператор и се извеждат неговите продажби с всички детайли.

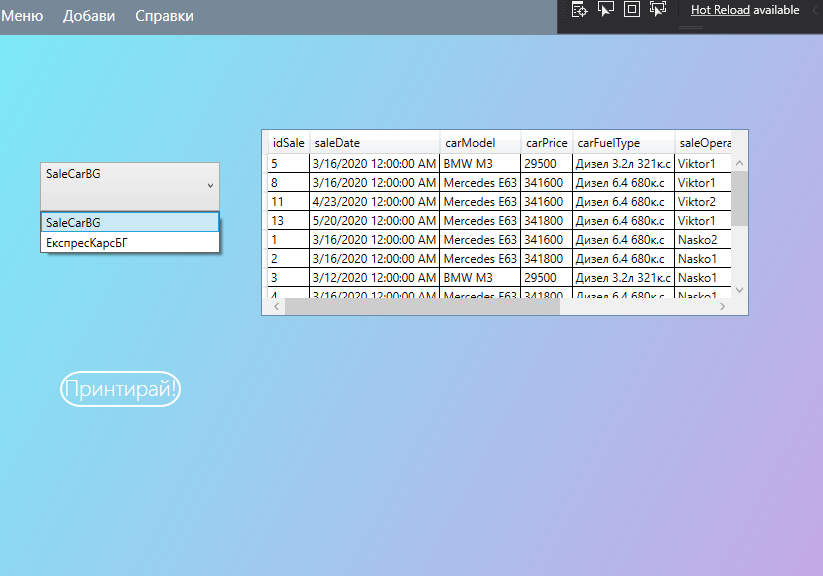
Натискането на бутон Принтирай извежда следното меню



От където се избира принтер и може да се принтира .



**Справка за коли от дадена компания:**



От комбобокса се избира компания и се извеждат продажби от дадената компания с всички детайли.

Натискането на бутон Принтирай дава възможност за принтиране , аналогично на справкката по-горе.

1. Код на БД и на интерфейса

*Скрипт за създаване на базата от данни:*

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema SaleDB

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema SaleDB

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `SaleDB` DEFAULT CHARACTER SET utf8 ;

USE `SaleDB` ;

-- -----------------------------------------------------

-- Table `SaleDB`.`classes`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`classes` (

`idclasses` INT NOT NULL AUTO\_INCREMENT,

`classType` VARCHAR(45) NULL,

PRIMARY KEY (`idclasses`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`color`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`color` (

`idcolor` INT NOT NULL AUTO\_INCREMENT,

`color` VARCHAR(45) NULL,

PRIMARY KEY (`idcolor`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`car`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`car` (

`idCar` INT NOT NULL AUTO\_INCREMENT,

`carModel` VARCHAR(45) NULL,

`carCondition` VARCHAR(45) NULL,

`carPrice` DOUBLE NULL,

`carStatus` VARCHAR(45) NULL,

`carFuelType` VARCHAR(45) NULL,

`carKM` DOUBLE NULL,

`classes\_idclasses` INT NOT NULL,

`color\_idcolor` INT NOT NULL,

PRIMARY KEY (`idCar`),

INDEX `fk\_car\_classes\_idx` (`classes\_idclasses` ASC) VISIBLE,

INDEX `fk\_car\_color1\_idx` (`color\_idcolor` ASC) VISIBLE,

CONSTRAINT `fk\_car\_classes`

FOREIGN KEY (`classes\_idclasses`)

REFERENCES `SaleDB`.`classes` (`idclasses`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_car\_color1`

FOREIGN KEY (`color\_idcolor`)

REFERENCES `SaleDB`.`color` (`idcolor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`company`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`company` (

`idcompany` INT NOT NULL AUTO\_INCREMENT,

`companyName` VARCHAR(45) NULL,

`companyAddress` VARCHAR(45) NULL,

PRIMARY KEY (`idcompany`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`saleOperator`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`saleOperator` (

`idsaleOperator` INT NOT NULL AUTO\_INCREMENT,

`saleOperatorName` VARCHAR(45) NULL,

`saleOperatorUserName` VARCHAR(45) NULL,

`saleOperatorPassword` VARCHAR(45) NULL,

`company\_idcompany` INT NOT NULL,

PRIMARY KEY (`idsaleOperator`),

INDEX `fk\_saleOperator\_company1\_idx` (`company\_idcompany` ASC) VISIBLE,

CONSTRAINT `fk\_saleOperator\_company1`

FOREIGN KEY (`company\_idcompany`)

REFERENCES `SaleDB`.`company` (`idcompany`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`customer` (

`idcustomer` INT NOT NULL AUTO\_INCREMENT,

`customerName` VARCHAR(45) NULL,

`customerAddress` VARCHAR(45) NULL,

`customerIDNum` VARCHAR(45) NULL,

PRIMARY KEY (`idcustomer`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `SaleDB`.`Sale`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `SaleDB`.`Sale` (

`idSale` INT NOT NULL AUTO\_INCREMENT,

`saleDate` DATE NULL,

`car\_idCar` INT NOT NULL,

`saleOperator\_idsaleOperator` INT NOT NULL,

`customer\_idcustomer` INT NOT NULL,

PRIMARY KEY (`idSale`),

INDEX `fk\_Sale\_car1\_idx` (`car\_idCar` ASC) VISIBLE,

INDEX `fk\_Sale\_saleOperator1\_idx` (`saleOperator\_idsaleOperator` ASC) VISIBLE,

INDEX `fk\_Sale\_customer1\_idx` (`customer\_idcustomer` ASC) VISIBLE,

CONSTRAINT `fk\_Sale\_car1`

FOREIGN KEY (`car\_idCar`)

REFERENCES `SaleDB`.`car` (`idCar`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Sale\_saleOperator1`

FOREIGN KEY (`saleOperator\_idsaleOperator`)

REFERENCES `SaleDB`.`saleOperator` (`idsaleOperator`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Sale\_customer1`

FOREIGN KEY (`customer\_idcustomer`)

REFERENCES `SaleDB`.`customer` (`idcustomer`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

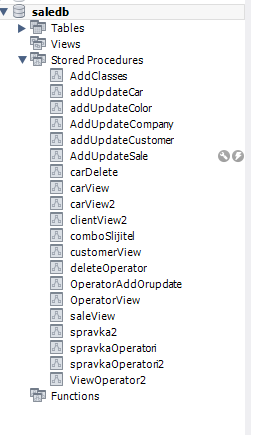
ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Процедури :**



**Процедура за добавяне на клас автомобил:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `AddClasses`(

\_idclasses int,

\_classType VARCHAR(45)

)

BEGIN

INSERT INTO classes (idclasses ,classType)

VALUES (\_idclasses , \_classType);

END

**Процедура за добавяне и/или редактиране на автомобил:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `addUpdateCar`(

\_idCar INT,

\_carModel VARCHAR(45),

\_carCondition VARCHAR(45),

\_carPrice DOUBLE,

\_carStatus VARCHAR(45),

\_carFuelType VARCHAR(45),

\_carKM VARCHAR(45),

\_classes\_idclasses VARCHAR(45),

\_color\_idcolor VARCHAR(45)

)

BEGIN

DECLARE color INT;

DECLARE classes INT;

SELECT idcolor INTO color FROM color WHERE color.color=\_color\_idcolor;

SELECT idclasses INTO classes FROM classes WHERE classes.classType=\_classes\_idclasses;

IF \_idCar = 0 THEN

INSERT INTO car (idCar ,carModel,carCondition,carPrice,carStatus,carFuelType,carKM,classes\_idclasses,color\_idcolor)

VALUES (\_idCar , \_carModel,\_carCondition,\_carPrice,\_carStatus,\_carFuelType,\_carKM,classes,color);

else

UPDATE car

SET

carModel = \_carModel,

carCondition = \_carCondition,

carPrice = \_carPrice,

carStatus=\_carStatus,

carFuelType=\_carFuelType,

carKM=\_carKM,

classes\_idclasses=classes,

color\_idcolor=color

WHERE idCar = \_idCar;

END IF;

END

**Процедура за добавяне и/или редактиране на цвят на автомобил:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `addUpdateColor`(

\_idcolor INT,

\_color VARCHAR(45)

)

BEGIN

IF \_idcolor = 0 THEN

INSERT INTO color (idcolor ,color)

VALUES (\_idcolor , \_color);

else

UPDATE color

SET

idcolor = \_idcolor,

color = \_color

WHERE \_idcolor = idcolor;

END IF;

END

**Процедура за добавяне и/или редактиране на компанията**

CREATE DEFINER=`root`@`localhost` PROCEDURE `AddUpdateCompany`(

\_idcompany INT,

\_companyName VARCHAR(45),

\_companyAddress VARCHAR(45)

)

BEGIN

IF \_idcompany = 0 THEN

INSERT INTO company (idcompany ,companyName,companyAddress)

VALUES (\_idcompany , \_companyName,\_companyAddress);

else

UPDATE company

SET

idcompany = \_idcompany,

companyName = \_companyName,

companyAddress = \_companyAddress

WHERE \_idcompany = idcompany;

END IF;

END

**Процедура за добавяне и/или редактиране на клиент**

CREATE DEFINER=`root`@`localhost` PROCEDURE `addUpdateCustomer`(

\_idcustomer INT,

\_customerName VARCHAR(45),

\_customerAddress VARCHAR(45),

\_customerIDNum VARCHAR(45)

)

BEGIN

IF \_idcustomer = 0 THEN

INSERT INTO customer (idcustomer ,customerName,customerAddress,customerIDNum)

VALUES (\_idcustomer , \_customerName,\_customerAddress,\_customerIDNum);

else

UPDATE customer

SET

idcustomer = \_idcustomer,

customerName = \_customerName,

customerAddress = \_customerAddress,

customerIDNum=\_customerIDNum

WHERE idcustomer = \_idcustomer;

END IF;

END

**Процедура за добавяне на продажба**

CREATE DEFINER=`root`@`localhost` PROCEDURE `AddUpdateSale`(

\_idSale INT,

\_saleDate datetime,

\_car\_idCar INT,

\_saleOperator\_idsaleOperator INT,

\_customer\_idcustomer INT

)

BEGIN

INSERT INTO sale (idSale ,saleDate,car\_idCar,saleOperator\_idsaleOperator,customer\_idcustomer)

VALUES (\_idSale , \_saleDate,\_car\_idCar,\_saleOperator\_idsaleOperator,\_customer\_idcustomer);

END

**Процедура за изтриване на кола**

CREATE DEFINER=`root`@`localhost` PROCEDURE `carDelete`(\_idCar INT)

BEGIN

DELETE FROM car

WHERE idCar = \_idCar;

END

**Процедура за извеждане на кола в таблица**

CREATE DEFINER=`root`@`localhost` PROCEDURE `carView`()

BEGIN

SELECT carModel , carCondition ,carPrice , carStatus , carFuelType , carKM, classes.classType , color.color , idcar

FROM car

INNER JOIN color ON color.idcolor = car.color\_idcolor

INNER JOIN classes ON classes.idclasses = car.classes\_idclasses;

END

**Процедура за извеждане на кола в таблица2**

CREATE DEFINER=`root`@`localhost` PROCEDURE `carView2`()

BEGIN

SELECT \*

FROM car;

END

**Процедура за извеждане на клиент в таблица**

CREATE DEFINER=`root`@`localhost` PROCEDURE `clientView2`()

BEGIN

SELECT \*

FROM customer;

END

**Процедура за извеждане на служител в комобокс**

CREATE DEFINER=`root`@`localhost` PROCEDURE `comboSlijitel`()

BEGIN

SELECT saleoperatorName , idsaleoperator

FROM saleoperator;

END

**Процедура за извеждане на клиент в таблица**

CREATE DEFINER=`root`@`localhost` PROCEDURE `customerView`()

BEGIN

SELECT \*

FROM customer;

END

**Процедура за изтриване на оператор**

CREATE DEFINER=`root`@`localhost` PROCEDURE `deleteOperator`(\_idsaleOperator INT)

BEGIN

DELETE FROM saleoperator

WHERE idsaleOperator = \_idsaleOperator ;

END

**Процедура за добавяне и/или редактиране на оператор**

CREATE DEFINER=`root`@`localhost` PROCEDURE `OperatorAddOrupdate`(

\_idsaleOperator INT ,

\_saleOperatorName Varchar(45),

\_saleOperatorUserName Varchar(45),

\_saleOperatorPassword Varchar(45),

\_idcompany VARCHAR(45)

)

BEGIN

DECLARE company INT;

SELECT idcompany INTO company FROM company WHERE company.companyName=\_idcompany;

IF \_idsaleOperator = 0 THEN

INSERT INTO saleoperator (saleOperatorName ,saleOperatorUserName,saleOperatorPassword,company\_idcompany)

VALUES (\_saleOperatorName , \_saleOperatorUserName,\_saleOperatorPassword,company);

else

UPDATE saleoperator

SET

saleOperatorName = \_saleOperatorName,

saleOperatorUserName = \_saleOperatorUserName,

saleOperatorPassword = \_saleOperatorPassword,

company\_idcompany = company

WHERE idsaleOperator = \_idsaleOperator;

END IF;

END

**Процедура за извеждане на оператор в таблица**

CREATE DEFINER=`root`@`localhost` PROCEDURE `OperatorView`()

BEGIN

SELECT saleoperatorName , saleOperatorUserName, saleOperatorPassword ,company.companyName, idsaleoperator

FROM saleoperator

INNER JOIN company ON company.idcompany = saleoperator.company\_idcompany;

END

**Процедура за извеждане на продажба в таблица**

CREATE DEFINER=`root`@`localhost` PROCEDURE `saleView`()

BEGIN

SELECT idSale , saleDate ,car.carModel , car.carPrice , car.carFuelType , saleoperator.saleOperatorName, customer.customerName

FROM sale

INNER JOIN customer ON customer.idcustomer = sale.customer\_idcustomer

INNER JOIN car ON car.idCar = sale.car\_idCar

INNER JOIN saleoperator ON saleoperator.idsaleOperator = sale.saleOperator\_idsaleOperator;

END

**Процедура за справка по компания**

CREATE DEFINER=`root`@`localhost` PROCEDURE `spravka2`(

\_idCompany INT

)

BEGIN

SELECT idSale , saleDate ,car.carModel , car.carPrice , car.carFuelType , saleoperator.saleOperatorName, customer.customerName,saleoperator.company\_idcompany

FROM sale

INNER JOIN customer ON customer.idcustomer = sale.customer\_idcustomer

INNER JOIN car ON car.idCar = sale.car\_idCar

INNER JOIN saleoperator ON saleoperator.idsaleOperator = sale.saleOperator\_idsaleOperator

WHERE saleoperator.company\_idcompany=\_idCompany;

END

**Процедура за справка по оператор**

CREATE DEFINER=`root`@`localhost` PROCEDURE `spravkaOperatori2`(

\_idsaleOperator INT

)

BEGIN

SELECT idSale , saleDate ,car.carModel , car.carPrice , car.carFuelType , saleoperator.saleOperatorName, customer.customerName,saleoperator.idsaleOperator

FROM sale

INNER JOIN customer ON customer.idcustomer = sale.customer\_idcustomer

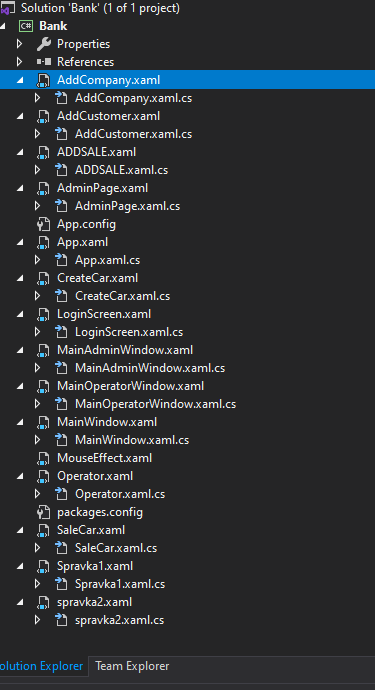
INNER JOIN car ON car.idCar = sale.car\_idCar

INNER JOIN saleoperator ON saleoperator.idsaleOperator = sale.saleOperator\_idsaleOperator

WHERE idsaleOperator=\_idsaleOperator;

END

**C# WPF код**



**Add company C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for AddCompany.xaml

/// </summary>

public partial class AddCompany : Page

{

int idcompany = 0;

int idcolor = 0;

int idclasses = 0;

string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public AddCompany()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("AddUpdateCompany", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idcompany", idcompany);

mySqlCmd.Parameters.AddWithValue("\_companyName", tbCompanyName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_companyAddress", tbCompanyAddress.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("addUpdateColor", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idcolor", idcolor);

mySqlCmd.Parameters.AddWithValue("\_color", tbColor.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("AddClasses", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idclasses", idclasses);

mySqlCmd.Parameters.AddWithValue("\_classType", tbCarClass.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

}

}

}

}

**Add company WPF interface code:**

<Page x:Class="Bank.AddCompany"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddCompany">

<Grid>

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<TextBox x:Name="tbCompanyName" HorizontalAlignment="Left" Height="37" Margin="292,89,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="151"/>

<TextBox x:Name="tbCompanyAddress" HorizontalAlignment="Left" Height="39" Margin="292,165,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="151"/>

<Label FontSize="18" Content="Име на компанията" HorizontalAlignment="Left" Margin="59,89,0,0" VerticalAlignment="Top"/>

<Label FontSize="18" Content="Адрес на компанията" HorizontalAlignment="Left" Margin="59,165,0,0" VerticalAlignment="Top"/>

<Button Content="Добави компания" HorizontalAlignment="Left" Margin="292,252,0,0" VerticalAlignment="Top" Width="90" Height="38" Click="Button\_Click"/>

<Label FontSize="18" Content="Цвят" HorizontalAlignment="Left" Margin="59,309,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbColor" HorizontalAlignment="Left" Height="37" Margin="292,306,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="151"/>

<Button Content="Добави цвят" HorizontalAlignment="Left" Margin="292,376,0,0" VerticalAlignment="Top" Width="90" Height="38" Click="Button\_Click\_1" />

<Button Content="Добави клас" HorizontalAlignment="Left" Margin="594,167,0,0" VerticalAlignment="Top" Width="94" Height="37" Click="Button\_Click\_2"/>

<TextBox x:Name="tbCarClass" HorizontalAlignment="Left" Height="33" Margin="565,128,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="154"/>

<Label FontSize="18" Content="Въведете клас на автомобила" HorizontalAlignment="Left" Margin="527,89,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.891,-2.225"/>

</Grid>

</Page>

**Add Customer C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for AddCustomer.xaml

/// </summary>

public partial class AddCustomer : Page

{

int idcustomer = 0;

string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public AddCustomer()

{

InitializeComponent();

gridfill();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("addUpdateCustomer", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idcustomer", idcustomer);

mySqlCmd.Parameters.AddWithValue("\_customerName", tbClientName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_customerAddress", tbCustomerAddress.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_customerIDNum", tbCustomerID.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

void gridfill()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("customerView", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtClients = new DataTable();

mySqlDa.Fill(dtClients);

gvClients.DataContext = dtClients;

}

}

}

}

**Add Customer WPF interface code:**

<Page x:Class="Bank.AddCustomer"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddCustomer">

<Grid Margin="0,0,-588,-107">

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Label FontSize="22" Content="Име на клиент" HorizontalAlignment="Left" Margin="15,80,0,0" VerticalAlignment="Top" RenderTransformOrigin="-1.943,0.422"/>

<Label FontSize="22" Content="Адрес на клиент" HorizontalAlignment="Left" Margin="15,147,0,0" VerticalAlignment="Top" RenderTransformOrigin="-1.943,0.422"/>

<Label FontSize="22" Content="№ лична карта" HorizontalAlignment="Left" Margin="15,212,0,0" VerticalAlignment="Top" RenderTransformOrigin="-1.943,0.422"/>

<TextBox FontSize="22" x:Name="tbClientName" HorizontalAlignment="Left" Height="40" Margin="221,77,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="168"/>

<TextBox FontSize="22" x:Name="tbCustomerAddress" HorizontalAlignment="Left" Height="40" Margin="221,146,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="168"/>

<TextBox FontSize="22" x:Name="tbCustomerID" HorizontalAlignment="Left" Height="40" Margin="221,211,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="168"/>

<DataGrid ItemsSource="{Binding}" x:Name="gvClients" HorizontalAlignment="Left" Height="223" Margin="530,54,0,0" VerticalAlignment="Top" Width="679"/>

<Button Content="Добави клиент" HorizontalAlignment="Left" Margin="530,359,0,0" VerticalAlignment="Top" Width="118" Height="35" Click="Button\_Click"/>

</Grid>

</Page>

**ADDSALE C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for ADDSALE.xaml

/// </summary>

public partial class ADDSALE : Page

{

public string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public int idSale=0;

public ADDSALE()

{

InitializeComponent();

gridfillClients();

gridfillCars();

comboxoxOperator();

}

public void gridfillCars()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("carView2", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtcarss = new DataTable();

mySqlDa.Fill(dtcarss);

gvCars.DataContext = dtcarss;

mysqlCon.Close();

}

}

public void gridfillClients()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("clientView2", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtClientss = new DataTable();

mySqlDa.Fill(dtClientss);

gvClients.DataContext = dtClientss;

mysqlCon.Close();

}

}

private void comboxoxOperator()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM saleOperator", mysqlCon);

DataSet ds = new DataSet();

category\_data.Fill(ds, "saleOperator");

Cboperator.DataContext = ds.Tables["saleOperator"].DefaultView;

Cboperator.DisplayMemberPath = "idsaleOperator";

Cboperator.SelectedIndex = 0;

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row1 = (DataRowView)gvCars.SelectedItem;

int selectedIDFromCar = (int)row1["idCar"];

DataRowView row3 = (DataRowView)gvClients.SelectedItem;

int selectedIDFromCustomers = (int)row3["idcustomer"];

var date = Convert.ToDateTime(dpDate.Text).ToString("yyyy/MM/dd");

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("AddUpdateSale", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idSale", idSale);

mySqlCmd.Parameters.AddWithValue("\_SaleDate", date);

mySqlCmd.Parameters.AddWithValue("\_car\_idCar", selectedIDFromCar);

mySqlCmd.Parameters.AddWithValue("\_saleOperator\_idsaleOperator", Cboperator.Text);

mySqlCmd.Parameters.AddWithValue("\_customer\_idcustomer", selectedIDFromCustomers);

this.Dispatcher.Invoke(() =>

{

mySqlCmd.ExecuteNonQuery();

});

MessageBox.Show("Submitted Successfully!");

mysqlCon.Close();

}

}

}

}

**ADDSALE WPF interface code:**

<Page x:Class="Bank.ADDSALE"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ADDSALE">

<Grid>

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<DataGrid ItemsSource="{Binding}" x:Name="gvCars" HorizontalAlignment="Left" Height="100" Margin="511,96,0,0" VerticalAlignment="Top" Width="100"/>

<DataGrid ItemsSource="{Binding}" x:Name="gvClients" HorizontalAlignment="Left" Height="100" Margin="290,96,0,0" VerticalAlignment="Top" Width="100"/>

<Button Content="Button" HorizontalAlignment="Left" Margin="555,326,0,0" VerticalAlignment="Top" Width="75" Click="Button\_Click"/>

<ComboBox ItemsSource="{Binding}" Name="Cboperator" HorizontalAlignment="Left" Margin="78,122,0,0" VerticalAlignment="Top" Width="120"/>

<DatePicker Name="dpDate" HorizontalAlignment="Left" Margin="198,276,0,0" VerticalAlignment="Top"/>

</Grid>

</Page>

**AdminPage C# code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for AdminPage.xaml

/// </summary>

public partial class AdminPage : Window

{

public AdminPage()

{

InitializeComponent();

}

}

}

**AdminPage WPF interface code:**

<Window x:Class="Bank.AdminPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

Title="AdminPage" Height="450" Width="800">

<Window.Resources>

<Style x:Key="menuParentStyle" TargetType="MenuItem">

<Setter Property="Background" Value="Black"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<SolidColorBrush x:Key="Menu.Static.Foreground" Color="White"/>

<SolidColorBrush x:Key="Menu.Static.Background" Color="Black"/>

<SolidColorBrush x:Key="Menu.Static.Border" Color="Transparent"/>

<Style x:Key="MenuItemStyle1" TargetType="{x:Type MenuItem}">

<Setter Property="HorizontalContentAlignment" Value="{Binding HorizontalContentAlignment, RelativeSource={RelativeSource AncestorType={x:Type ItemsControl}}}"/>

<Setter Property="VerticalContentAlignment" Value="{Binding VerticalContentAlignment, RelativeSource={RelativeSource AncestorType={x:Type ItemsControl}}}"/>

<Setter Property="Background" Value="Black"/>

<Setter Property="BorderBrush" Value="Transparent"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="ScrollViewer.PanningMode" Value="Both"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template" Value="{DynamicResource MenuItemControlTemplate1}"/>

<Style.Triggers>

<Trigger Property="Role" Value="TopLevelHeader">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="BorderBrush" Value="Transparent"/>

<Setter Property="Foreground" Value="{StaticResource Menu.Static.Foreground}"/>

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=TopLevelHeaderTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

<Setter Property="Padding" Value="6,0"/>

</Trigger>

<Trigger Property="Role" Value="TopLevelItem">

<Setter Property="Background" Value="{StaticResource Menu.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource Menu.Static.Border}"/>

<Setter Property="Foreground" Value="{StaticResource Menu.Static.Foreground}"/>

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=TopLevelItemTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

<Setter Property="Padding" Value="6,0"/>

</Trigger>

<Trigger Property="Role" Value="SubmenuHeader">

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=SubmenuHeaderTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Background" Value="Pink" />

</Trigger>

<Trigger Property="IsKeyboardFocusWithin" Value="true">

<Setter Property="Background" Value="#0a99f3" />

</Trigger>

</Style.Triggers>

</Style>

<ControlTemplate x:Key="MenuItemControlTemplate1" TargetType="{x:Type MenuItem}">

<Border x:Name="templateRoot" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<Grid Margin="-1">

<Grid.ColumnDefinitions>

<ColumnDefinition MinWidth="22" SharedSizeGroup="MenuItemIconColumnGroup" Width="Auto"/>

<ColumnDefinition Width="13"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="30"/>

<ColumnDefinition SharedSizeGroup="MenuItemIGTColumnGroup" Width="Auto"/>

<ColumnDefinition Width="20"/>

</Grid.ColumnDefinitions>

<ContentPresenter x:Name="Icon" Content="{TemplateBinding Icon}" ContentSource="Icon" HorizontalAlignment="Center" Height="16" Margin="3" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center" Width="16"/>

<Border x:Name="GlyphPanel" BorderBrush="#FF26A0DA" BorderThickness="1" Background="#3D26A0DA" ClipToBounds="False" HorizontalAlignment="Center" Height="22" Margin="-1,0,0,0" Visibility="Hidden" VerticalAlignment="Center" Width="22">

<Path x:Name="Glyph" Data="F1M10,1.2L4.7,9.1 4.5,9.1 0,5.2 1.3,3.5 4.3,6.1 8.3,0 10,1.2z" Fill="#FF212121" FlowDirection="LeftToRight" Height="11" Width="10"/>

</Border>

<ContentPresenter x:Name="menuHeaderContainer" ContentTemplate="{TemplateBinding HeaderTemplate}" Content="{TemplateBinding Header}" Grid.Column="2" ContentStringFormat="{TemplateBinding HeaderStringFormat}" ContentSource="Header" HorizontalAlignment="Left" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center"/>

<TextBlock x:Name="menuGestureText" Grid.Column="4" Margin="{TemplateBinding Padding}" Opacity="0.7" Text="{TemplateBinding InputGestureText}" VerticalAlignment="Center"/>

</Grid>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="Icon" Value="{x:Null}">

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsChecked" Value="True">

<Setter Property="Visibility" TargetName="GlyphPanel" Value="Visible"/>

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="templateRoot" Value="Blue"/>

<!--<Setter Property="BorderBrush" TargetName="templateRoot" Value="Yellow"/>-->

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="TextElement.Foreground" TargetName="templateRoot" Value="#FF707070"/>

<Setter Property="Fill" TargetName="Glyph" Value="#FF707070"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsHighlighted" Value="True"/>

<Condition Property="IsEnabled" Value="False"/>

</MultiTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="#0A000000"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="#21000000"/>

</MultiTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<ControlTemplate x:Key="MenuItemControlTemplate2" TargetType="{x:Type MenuItem}">

<Border x:Name="templateRoot" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<Grid VerticalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="Auto"/>

</Grid.ColumnDefinitions>

<ContentPresenter x:Name="Icon" Content="{TemplateBinding Icon}" ContentSource="Icon" HorizontalAlignment="Center" Height="16" Margin="3" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center" Width="16"/>

<Path x:Name="GlyphPanel" Data="F1M10,1.2L4.7,9.1 4.5,9.1 0,5.2 1.3,3.5 4.3,6.1 8.3,0 10,1.2z" Fill="{TemplateBinding Foreground}" FlowDirection="LeftToRight" Margin="3" Visibility="Collapsed" VerticalAlignment="Center"/>

<ContentPresenter ContentTemplate="{TemplateBinding HeaderTemplate}" Content="{TemplateBinding Header}" Grid.Column="1" ContentStringFormat="{TemplateBinding HeaderStringFormat}" ContentSource="Header" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

<Popup x:Name="PART\_Popup" AllowsTransparency="True" Focusable="False" IsOpen="{Binding IsSubmenuOpen, RelativeSource={RelativeSource TemplatedParent}}" PopupAnimation="{DynamicResource {x:Static SystemParameters.MenuPopupAnimationKey}}" Placement="Bottom">

<Border x:Name="SubMenuBorder" BorderBrush="#FF999999" BorderThickness="1" Background="#FFF0F0F0" Padding="2" >

<ScrollViewer x:Name="SubMenuScrollViewer" Style="{DynamicResource {ComponentResourceKey ResourceId=MenuScrollViewer, TypeInTargetAssembly={x:Type FrameworkElement}}}">

<Grid RenderOptions.ClearTypeHint="Enabled">

<Canvas HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="OpaqueRect" Fill="{Binding Background, ElementName=SubMenuBorder}" Height="{Binding ActualHeight, ElementName=SubMenuBorder}" Width="{Binding ActualWidth, ElementName=SubMenuBorder}"/>

</Canvas>

<Rectangle Fill="#FFD7D7D7" HorizontalAlignment="Left" Margin="29,2,0,2" Width="1"/>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Cycle" Grid.IsSharedSizeScope="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" KeyboardNavigation.TabNavigation="Cycle"/>

</Grid>

</ScrollViewer>

</Border>

</Popup>

</Grid>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsSuspendingPopupAnimation" Value="True">

<Setter Property="PopupAnimation" TargetName="PART\_Popup" Value="None"/>

</Trigger>

<Trigger Property="Icon" Value="{x:Null}">

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsChecked" Value="True">

<Setter Property="Visibility" TargetName="GlyphPanel" Value="Visible"/>

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="templateRoot" Value="Blue"/>

<!--<Setter Property="BorderBrush" TargetName="templateRoot" Value="#FF26A0DA"/>-->

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="TextElement.Foreground" TargetName="templateRoot" Value="#FF707070"/>

<Setter Property="Fill" TargetName="GlyphPanel" Value="#FF707070"/>

</Trigger>

<Trigger Property="CanContentScroll" SourceName="SubMenuScrollViewer" Value="False">

<Setter Property="Canvas.Top" TargetName="OpaqueRect" Value="{Binding VerticalOffset, ElementName=SubMenuScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="OpaqueRect" Value="{Binding HorizontalOffset, ElementName=SubMenuScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Window.Resources>

<StackPanel>

<Menu>

<MenuItem Header="\_File" Style="{StaticResource menuParentStyle}" ItemContainerStyle="{DynamicResource MenuItemStyle1}" Template="{DynamicResource MenuItemControlTemplate2}" >

<MenuItem Header="\_New" />

<MenuItem Header="\_Save"/>

<MenuItem Header="\_Quit" />

</MenuItem>

<MenuItem Header="\_Edit" Style="{StaticResource menuParentStyle}" ItemContainerStyle="{DynamicResource MenuItemStyle1}" Template="{DynamicResource MenuItemControlTemplate2}">

<MenuItem Header="\_Copy"/>

<MenuItem Header="\_Cut" />

<MenuItem Header="\_Paste" />

</MenuItem>

</Menu>

<Grid Height="395" >

<Border CornerRadius="10" Grid.RowSpan="2">

<Border.Background>

<LinearGradientBrush>

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Border.Background>

</Border>

</Grid>

</StackPanel>

</Window>

**CreateCar C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for CreateCar.xaml

/// </summary>

public partial class CreateCar : Page

{

public CreateCar()

{

InitializeComponent();

comboxox();

comboxox2();

gridfill();

}

int idCar = 0;

string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

void gridfill()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("carView", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtcars = new DataTable();

mySqlDa.Fill(dtcars);

gvCars.DataContext = dtcars;

}

}

private void comboxox()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM color", mysqlCon);

DataSet ds = new DataSet();

category\_data.Fill(ds, "color");

cbColor.DataContext = ds.Tables["color"].DefaultView;

cbColor.DisplayMemberPath = "color";

cbColor.SelectedIndex = 0;

}

}

private void comboxox2()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM classes", mysqlCon);

DataSet ds = new DataSet();

category\_data.Fill(ds, "classes");

cbCarClass.DataContext = ds.Tables["classes"].DefaultView;

cbCarClass.DisplayMemberPath = "classType";

cbCarClass.SelectedIndex = 0;

}

}

private void btsave\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("addUpdateCar", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idCar", idCar);

mySqlCmd.Parameters.AddWithValue("\_carModel", tbModel.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carCondition", tbCondition.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carPrice", tbPrice.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carStatus", tbAvailability.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carFuelType", tbEngine.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carKM", tbCurrKM.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_classes\_idclasses", cbCarClass.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_color\_idcolor", cbColor.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void DataGrid\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

DataRowView rowView = gvCars.SelectedItem as DataRowView;

tbModel.Text = rowView.Row[0].ToString();

tbCondition.Text = rowView.Row[1].ToString();

tbPrice.Text = rowView.Row[2].ToString();

tbAvailability.Text = rowView.Row[3].ToString();

tbEngine.Text = rowView.Row[4].ToString();

tbCurrKM.Text = rowView.Row[5].ToString();

}

private void btChange\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row = (DataRowView)gvCars.SelectedItem;

int selectedIDFromDataGrid = (int)row["idCar"];

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("addUpdateCar", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idCar", selectedIDFromDataGrid);

mySqlCmd.Parameters.AddWithValue("\_carModel", tbModel.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carCondition", tbCondition.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carPrice", tbPrice.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carStatus", tbAvailability.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carFuelType", tbEngine.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_carKM", tbCurrKM.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_classes\_idclasses", cbCarClass.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_color\_idcolor", cbColor.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row = (DataRowView)gvCars.SelectedItem;

int selectedIDFromDataGrid = (int)row["idCar"];

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("carDelete", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idCar", selectedIDFromDataGrid);

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

}

}

**CreateCar WPF interface code:**

<Page x:Class="Bank.CreateCar"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="1600"

Title="CreateCar">

<Grid Margin="-405,0,-294,-354" RenderTransformOrigin="0.491,0.505">

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<TextBox x:Name="tbModel" HorizontalAlignment="Left" Height="43" Margin="820,8,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Модел на автомобил" HorizontalAlignment="Left" Margin="461,12,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbCondition" HorizontalAlignment="Left" Height="40" Margin="820,71,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Състояние на автомобила" HorizontalAlignment="Left" Margin="461,72,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbPrice" HorizontalAlignment="Left" Height="43" Margin="821,137,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Цена на автомобил" HorizontalAlignment="Left" Margin="462,141,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbAvailability" HorizontalAlignment="Left" Height="43" Margin="819,209,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Наличност" HorizontalAlignment="Left" Margin="460,213,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbEngine" HorizontalAlignment="Left" Height="46" Margin="820,288,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Тип на двигателя" HorizontalAlignment="Left" Margin="461,288,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="tbCurrKM" HorizontalAlignment="Left" Height="43" Margin="819,371,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="206" RenderTransformOrigin="0.493,1.02"/>

<Label FontSize="22" Content="Текущи километри" HorizontalAlignment="Left" Margin="460,375,0,0" VerticalAlignment="Top"/>

<Label FontSize="22" Content="Клас на автомобила" HorizontalAlignment="Left" Margin="460,448,0,0" VerticalAlignment="Top"/>

<Label FontSize="22" Content="Цвят на автомобила" HorizontalAlignment="Left" Margin="468,521,0,0" VerticalAlignment="Top"/>

<DataGrid ItemsSource="{Binding}" x:Name="gvCars" HorizontalAlignment="Left" Margin="1139,12,0,663" Width="715" MouseDoubleClick="DataGrid\_MouseDoubleClick"/>

<Button x:Name="btsave" Content="Запис" HorizontalAlignment="Left" Margin="1159,465,0,0" VerticalAlignment="Top" Width="187" Click="btsave\_Click" Height="45"/>

<ComboBox ItemsSource="{Binding}" FontSize="22" Name="cbCarClass" HorizontalAlignment="Left" Margin="819,448,0,0" VerticalAlignment="Top" Width="206" Height="43"/>

<ComboBox ItemsSource="{Binding}" FontSize="22" Name="cbColor" HorizontalAlignment="Left" Margin="818,511,0,0" VerticalAlignment="Top" Width="206" Height="49"/>

<Button x:Name="btChange" Content="Промени" HorizontalAlignment="Left" Margin="1378,465,0,0" VerticalAlignment="Top" Width="183" Click="btChange\_Click" Height="45"/>

<Button Content="Премахни" HorizontalAlignment="Left" Margin="1661,465,0,0" VerticalAlignment="Top" Width="193" Click="Button\_Click" Height="45"/>

</Grid>

</Page>

**LoginCreen C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.OleDb;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for LoginScreen.xaml

/// </summary>

public partial class LoginScreen : Window

{

public LoginScreen()

{

InitializeComponent();

}

Operator op;

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

String username1 = username.Text.ToString();

String password1 = password.Password.ToString();

//connection.Open();

string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

MySqlConnection conn = new MySqlConnection(myconnstrng);

try

{

//SQL Query to check login

//string sql = "SELECT userName , password FROM operator WHERE userName=@username AND password=@password ";

string sql = " SELECT saleOperatorUserName , saleOperatorPassword from saleoperator WHERE saleOperatorUserName='" + username1 + "'and saleOperatorPassword='" + password1 + "'";

//Creating SQL Command to pass value

MySqlCommand cmd = new MySqlCommand(sql, conn);

//cmd.Parameters.AddWithValue("@username", username1);

//cmd.Parameters.AddWithValue("@password", password1);

MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

//Checking The rows in DataTable

if (dt.Rows.Count > 0)

{

//Login Sucessful

MainOperatorWindow mainoperatorwindow = new MainOperatorWindow();

MessageBox.Show("Sucsess login!");

mainoperatorwindow.Show();

this.Close();

}

else if (username1.Equals("admin") && password1.Equals("admin"))

{

MainWin adminpage = new MainWin();

MessageBox.Show("Sucsess login! Welcome to administrator account");

adminpage.Show();

this.Close();

}

else

{

//Login Failed

MessageBox.Show("Invalid Login please check username and password");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

conn.Close();

}

}

}

}

**LoginScreen WPF interface code:**

<Window x:Class="Bank.LoginScreen"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="380"

AllowsTransparency="True" Background="Transparent"

WindowStyle="None" ResizeMode="NoResize" MouseDown="Window\_MouseDown">

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Border CornerRadius="10" Grid.RowSpan="2">

<Border.Background>

<LinearGradientBrush>

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Border.Background>

</Border>

<StackPanel VerticalAlignment="Center">

<Image Source="C:\Users\Viko\_Tuzara\Desktop\SBDProject\Bank\v%20lo.png" Width="80"/>

<TextBlock Text="Cars"

FontWeight="Light"

FontFamily="helvetica"

FontSize="22"

Foreground="White"

HorizontalAlignment="Center"/>

</StackPanel>

<StackPanel Grid.Row="1" >

<StackPanel Orientation="Horizontal">

<TextBox FontFamily="Helvetica"

Name="username"

FontWeight="Light"

Text="Username"

FontSize="20"

HorizontalAlignment="Center"

Foreground="White"

Background="Transparent"

BorderThickness="0"

Width="235"

HorizontalContentAlignment="Left"

Opacity="0.5"

Height="25"

Margin="63,0,0,0"/>

</StackPanel>

<Border Width="250"

Height="2"

Background="White"

Opacity="0.5"/>

<StackPanel Orientation="Horizontal" Margin="0,20,0,0">

<PasswordBox FontFamily="Helvetica"

Name="password"

FontWeight="Light"

Password="Password"

FontSize="20"

HorizontalAlignment="Center"

Foreground="White"

Background="Transparent"

BorderThickness="0"

Width="235"

HorizontalContentAlignment="Left"

Opacity="0.5"

Height="25"

Margin="63,0,0,0"/>

</StackPanel>

<Border Width="250"

Height="2"

Background="White"

Opacity="0.5"/>

<StackPanel Orientation="Horizontal" Margin="0,50,0,0">

<Button Width="100" Height="40" Content="LOGIN" Margin="60,0,60,0" Click="Button\_Click"/>

<Button Width="100" Height="40" Content="REGISTER"/>

</StackPanel>

</StackPanel>

</Grid>

</Grid>

</Window>

**MainAdminWindow C# code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for MainWin.xaml

/// </summary>

public partial class MainWin : Window

{

public MainWin()

{

InitializeComponent();

}

private void adminMainFrame\_Navigated(object sender, System.Windows.Navigation.NavigationEventArgs e)

{

}

private void Operator(object sender, RoutedEventArgs e)

{

}

private void MenuItem\_Click(object sender, RoutedEventArgs e)

{

adminMainFrame.Navigate(new Operator());

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void MenuItem\_Click\_1(object sender, RoutedEventArgs e)

{

LoginScreen login = new LoginScreen();

login.Show();

this.Close();

}

private void MenuItem\_Click\_2(object sender, RoutedEventArgs e)

{

System.Environment.Exit(0);

}

private void MenuItem\_Click\_3(object sender, RoutedEventArgs e)

{

adminMainFrame.Navigate(new AddCompany());

}

}

}

**MainAdminWindow WPF interface code:**

<Window x:Class="Bank.MainWin"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

Title="MainWin" Height="600" Width="1400" WindowStyle="None" ResizeMode="NoResize" MouseDown="Window\_MouseDown">

<Window.Resources>

<BooleanToVisibilityConverter x:Key="btv" />

<!-- style a top level menu item -->

<ControlTemplate x:Key="VsMenuTop" TargetType="MenuItem">

<StackPanel TextBlock.FontSize="15px" Height="40">

<!-- label, icons, etc. -->

<Label Content="{TemplateBinding Header}" Margin="5" Foreground="White" />

<!-- sub items -->

<Popup IsOpen="{TemplateBinding IsSubmenuOpen}" AllowsTransparency="True" Focusable="False">

<Border BorderThickness="1" Background="White" BorderBrush="red">

<StackPanel IsItemsHost="True" />

</Border>

</Popup>

<StackPanel.Style>

<Style TargetType="StackPanel">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="Gray" />

</Trigger>

</Style.Triggers>

</Style>

</StackPanel.Style>

</StackPanel>

</ControlTemplate>

<!-- style a non-top level menu item -->

<ControlTemplate x:Key="VsMenuSub" TargetType="MenuItem">

<DockPanel TextBlock.FontSize="15px" x:Name="panel">

<!-- label, icons, etc. -->

<Image Source="{Binding Icon, RelativeSource={RelativeSource TemplatedParent}}" Width="20" Margin="5,0" />

<Label Content="{TemplateBinding Header}" Foreground="Black" Margin="0,5,5,5" />

<!-- sub items -->

<Popup IsOpen="{TemplateBinding IsSubmenuOpen}" AllowsTransparency="True" Focusable="False" Placement="Right" >

<Border BorderThickness="1" Background="White" BorderBrush="SlateGray">

<StackPanel IsItemsHost="True" />

</Border>

</Popup>

</DockPanel>

<ControlTemplate.Triggers>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="panel" Value="SlateGray" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<!-- some base stuff -->

<Style TargetType="Menu">

<Setter Property="Background" Value="LightSlateGray" />

<Setter Property="Height" Value="40px" />

</Style>

</Window.Resources>

<!-- the real app and real menu -->

<StackPanel>

<Menu IsMainMenu="True" >

<MenuItem Header="\_Меню" Template="{StaticResource VsMenuTop}" >

<MenuItem Header="\_Изход от администраторския акаунт" Template="{StaticResource VsMenuSub}" Icon="images/log-out.png" Click="MenuItem\_Click\_1" />

<MenuItem Header="\_Изход от приложението" Template="{StaticResource VsMenuSub}" Icon="images/log-out.png" Click="MenuItem\_Click\_2" />

</MenuItem>

<MenuItem Header="\_Добави" Template="{StaticResource VsMenuTop}">

<MenuItem Header="\_Добави/Редактирай/Изтрий Оператор" Template="{StaticResource VsMenuSub}" Icon="images/nurse.png" Click="MenuItem\_Click" />

<MenuItem Header="\_Добави нова компания/нов цвят/нов клас" Template="{StaticResource VsMenuSub}" Icon="images/test-tubes.png" Click="MenuItem\_Click\_3" />

</MenuItem>

</Menu>

<Frame Name="adminMainFrame" Height="759" NavigationUIVisibility="Hidden"/>

</StackPanel>

</Window>

**MainOperatorWindow C# code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for MainOperatorWindow.xaml

/// </summary>

public partial class MainOperatorWindow : Window

{

public MainOperatorWindow()

{

InitializeComponent();

}

private void Window\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.LeftButton == MouseButtonState.Pressed)

DragMove();

}

private void MenuItem\_Click(object sender, RoutedEventArgs e)

{

OperatorMainFrame.Navigate(new CreateCar());

}

private void MenuItem\_Click\_1(object sender, RoutedEventArgs e)

{

LoginScreen login = new LoginScreen();

login.Show();

this.Close();

}

private void MenuItem\_Click\_2(object sender, RoutedEventArgs e)

{

System.Environment.Exit(0);

}

private void MenuItem\_Click\_3(object sender, RoutedEventArgs e)

{

OperatorMainFrame.Navigate(new AddCustomer());

}

private void MenuItem\_Click\_4(object sender, RoutedEventArgs e)

{

OperatorMainFrame.Navigate(new SaleCar());

}

private void MenuItem\_Click5(object sender, RoutedEventArgs e)

{

OperatorMainFrame.Navigate(new Spravka1());

}

private void MenuItem\_Click6(object sender, RoutedEventArgs e)

{

OperatorMainFrame.Navigate(new spravka2());

}

}

}

**MainOperatorWindow WPF interface code:**

<Window x:Class="Bank.MainOperatorWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

Title="MainWin" Height="600" Width="1400" WindowStyle="None" ResizeMode="NoResize" MouseDown="Window\_MouseDown">

<Window.Resources>

<BooleanToVisibilityConverter x:Key="btv" />

<!-- style a top level menu item -->

<ControlTemplate x:Key="VsMenuTop" TargetType="MenuItem">

<StackPanel TextBlock.FontSize="15px" Height="40">

<!-- label, icons, etc. -->

<Label Content="{TemplateBinding Header}" Margin="5" Foreground="White" />

<!-- sub items -->

<Popup IsOpen="{TemplateBinding IsSubmenuOpen}" AllowsTransparency="True" Focusable="False">

<Border BorderThickness="1" Background="White" BorderBrush="red">

<StackPanel IsItemsHost="True" />

</Border>

</Popup>

<StackPanel.Style>

<Style TargetType="StackPanel">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="Gray" />

</Trigger>

</Style.Triggers>

</Style>

</StackPanel.Style>

</StackPanel>

</ControlTemplate>

<!-- style a non-top level menu item -->

<ControlTemplate x:Key="VsMenuSub" TargetType="MenuItem">

<DockPanel TextBlock.FontSize="15px" x:Name="panel">

<!-- label, icons, etc. -->

<Image Source="{Binding Icon, RelativeSource={RelativeSource TemplatedParent}}" Width="20" Margin="5,0" />

<Label Content="{TemplateBinding Header}" Foreground="Black" Margin="0,5,5,5" />

<!-- sub items -->

<Popup IsOpen="{TemplateBinding IsSubmenuOpen}" AllowsTransparency="True" Focusable="False" Placement="Right" >

<Border BorderThickness="1" Background="White" BorderBrush="SlateGray">

<StackPanel IsItemsHost="True" />

</Border>

</Popup>

</DockPanel>

<ControlTemplate.Triggers>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="panel" Value="SlateGray" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<!-- some base stuff -->

<Style TargetType="Menu">

<Setter Property="Background" Value="LightSlateGray" />

<Setter Property="Height" Value="40px" />

</Style>

</Window.Resources>

<!-- the real app and real menu -->

<StackPanel>

<Menu IsMainMenu="True" >

<MenuItem Header="\_Меню" Template="{StaticResource VsMenuTop}" >

<MenuItem Header="\_Изход от акаунт" Template="{StaticResource VsMenuSub}" Icon="images/log-out.png" Click="MenuItem\_Click\_1" />

<MenuItem Header="\_Изход от приложението" Template="{StaticResource VsMenuSub}" Icon="images/log-out.png" Click="MenuItem\_Click\_2" />

</MenuItem>

<MenuItem Header="\_Добави" Template="{StaticResource VsMenuTop}">

<MenuItem Header="\_Добави кола" Template="{StaticResource VsMenuSub}" Icon="images/nurse.png" Click="MenuItem\_Click" />

<MenuItem Header="\_Добави Клиент" Template="{StaticResource VsMenuSub}" Icon="images/test-tubes.png" Click="MenuItem\_Click\_3" />

<MenuItem Header="\_Продай Кола" Template="{StaticResource VsMenuSub}" Icon="images/hospital.png" Click="MenuItem\_Click\_4"/>

</MenuItem>

<MenuItem Header="Справки" Template="{StaticResource VsMenuTop}">

<MenuItem Header="\_Справка за коли продадени от служител" Template="{StaticResource VsMenuSub}" Icon="images/nurse.png" Click="MenuItem\_Click5" />

<MenuItem Header="\_Справка за коли от дадена компания" Template="{StaticResource VsMenuSub}" Icon="images/nurse.png" Click="MenuItem\_Click6" />

</MenuItem>

</Menu>

<Frame Name="OperatorMainFrame" Height="760" NavigationUIVisibility="Hidden" />

</StackPanel>

</Window>

**MainWindow C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

string connectionString = "server=localhost;user=root;database=rentacardb;port=3306;password=123456";

MySqlConnection connection = new MySqlConnection(connectionString);

MySqlCommand cmd = new MySqlCommand("select \* from client", connection);

connection.Open();

DataTable dt = new DataTable();

dt.Load(cmd.ExecuteReader());

connection.Close();

//dbGrid.DataContext = dt;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

}

private void MenuItem\_Click(object sender, RoutedEventArgs e)

{

}

}

}

**MainWindow WPF interface code:**

<Window x:Class="Bank.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Window.Resources>

<Style x:Key="menuParentStyle" TargetType="MenuItem">

<Setter Property="Background" Value="Black"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<SolidColorBrush x:Key="Menu.Static.Foreground" Color="White"/>

<SolidColorBrush x:Key="Menu.Static.Background" Color="Black"/>

<SolidColorBrush x:Key="Menu.Static.Border" Color="Transparent"/>

<Style x:Key="MenuItemStyle1" TargetType="{x:Type MenuItem}">

<Setter Property="HorizontalContentAlignment" Value="{Binding HorizontalContentAlignment, RelativeSource={RelativeSource AncestorType={x:Type ItemsControl}}}"/>

<Setter Property="VerticalContentAlignment" Value="{Binding VerticalContentAlignment, RelativeSource={RelativeSource AncestorType={x:Type ItemsControl}}}"/>

<Setter Property="Background" Value="Black"/>

<Setter Property="BorderBrush" Value="Transparent"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="ScrollViewer.PanningMode" Value="Both"/>

<Setter Property="Stylus.IsFlicksEnabled" Value="False"/>

<Setter Property="Template" Value="{DynamicResource MenuItemControlTemplate1}"/>

<Style.Triggers>

<Trigger Property="Role" Value="TopLevelHeader">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="BorderBrush" Value="Transparent"/>

<Setter Property="Foreground" Value="{StaticResource Menu.Static.Foreground}"/>

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=TopLevelHeaderTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

<Setter Property="Padding" Value="6,0"/>

</Trigger>

<Trigger Property="Role" Value="TopLevelItem">

<Setter Property="Background" Value="{StaticResource Menu.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource Menu.Static.Border}"/>

<Setter Property="Foreground" Value="{StaticResource Menu.Static.Foreground}"/>

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=TopLevelItemTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

<Setter Property="Padding" Value="6,0"/>

</Trigger>

<Trigger Property="Role" Value="SubmenuHeader">

<Setter Property="Template" Value="{DynamicResource {ComponentResourceKey ResourceId=SubmenuHeaderTemplateKey, TypeInTargetAssembly={x:Type MenuItem}}}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="true">

<Setter Property="Background" Value="Pink" />

</Trigger>

<Trigger Property="IsKeyboardFocusWithin" Value="true">

<Setter Property="Background" Value="#0a99f3" />

</Trigger>

</Style.Triggers>

</Style>

<ControlTemplate x:Key="MenuItemControlTemplate1" TargetType="{x:Type MenuItem}">

<Border x:Name="templateRoot" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<Grid Margin="-1">

<Grid.ColumnDefinitions>

<ColumnDefinition MinWidth="22" SharedSizeGroup="MenuItemIconColumnGroup" Width="Auto"/>

<ColumnDefinition Width="13"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="30"/>

<ColumnDefinition SharedSizeGroup="MenuItemIGTColumnGroup" Width="Auto"/>

<ColumnDefinition Width="20"/>

</Grid.ColumnDefinitions>

<ContentPresenter x:Name="Icon" Content="{TemplateBinding Icon}" ContentSource="Icon" HorizontalAlignment="Center" Height="16" Margin="3" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center" Width="16"/>

<Border x:Name="GlyphPanel" BorderBrush="#FF26A0DA" BorderThickness="1" Background="#3D26A0DA" ClipToBounds="False" HorizontalAlignment="Center" Height="22" Margin="-1,0,0,0" Visibility="Hidden" VerticalAlignment="Center" Width="22">

<Path x:Name="Glyph" Data="F1M10,1.2L4.7,9.1 4.5,9.1 0,5.2 1.3,3.5 4.3,6.1 8.3,0 10,1.2z" Fill="#FF212121" FlowDirection="LeftToRight" Height="11" Width="10"/>

</Border>

<ContentPresenter x:Name="menuHeaderContainer" ContentTemplate="{TemplateBinding HeaderTemplate}" Content="{TemplateBinding Header}" Grid.Column="2" ContentStringFormat="{TemplateBinding HeaderStringFormat}" ContentSource="Header" HorizontalAlignment="Left" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center"/>

<TextBlock x:Name="menuGestureText" Grid.Column="4" Margin="{TemplateBinding Padding}" Opacity="0.7" Text="{TemplateBinding InputGestureText}" VerticalAlignment="Center"/>

</Grid>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="Icon" Value="{x:Null}">

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsChecked" Value="True">

<Setter Property="Visibility" TargetName="GlyphPanel" Value="Visible"/>

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="templateRoot" Value="Blue"/>

<!--<Setter Property="BorderBrush" TargetName="templateRoot" Value="Yellow"/>-->

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="TextElement.Foreground" TargetName="templateRoot" Value="#FF707070"/>

<Setter Property="Fill" TargetName="Glyph" Value="#FF707070"/>

</Trigger>

<MultiTrigger>

<MultiTrigger.Conditions>

<Condition Property="IsHighlighted" Value="True"/>

<Condition Property="IsEnabled" Value="False"/>

</MultiTrigger.Conditions>

<Setter Property="Background" TargetName="templateRoot" Value="#0A000000"/>

<Setter Property="BorderBrush" TargetName="templateRoot" Value="#21000000"/>

</MultiTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

<ControlTemplate x:Key="MenuItemControlTemplate2" TargetType="{x:Type MenuItem}">

<Border x:Name="templateRoot" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">

<Grid VerticalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="Auto"/>

</Grid.ColumnDefinitions>

<ContentPresenter x:Name="Icon" Content="{TemplateBinding Icon}" ContentSource="Icon" HorizontalAlignment="Center" Height="16" Margin="3" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="Center" Width="16"/>

<Path x:Name="GlyphPanel" Data="F1M10,1.2L4.7,9.1 4.5,9.1 0,5.2 1.3,3.5 4.3,6.1 8.3,0 10,1.2z" Fill="{TemplateBinding Foreground}" FlowDirection="LeftToRight" Margin="3" Visibility="Collapsed" VerticalAlignment="Center"/>

<ContentPresenter ContentTemplate="{TemplateBinding HeaderTemplate}" Content="{TemplateBinding Header}" Grid.Column="1" ContentStringFormat="{TemplateBinding HeaderStringFormat}" ContentSource="Header" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>

<Popup x:Name="PART\_Popup" AllowsTransparency="True" Focusable="False" IsOpen="{Binding IsSubmenuOpen, RelativeSource={RelativeSource TemplatedParent}}" PopupAnimation="{DynamicResource {x:Static SystemParameters.MenuPopupAnimationKey}}" Placement="Bottom">

<Border x:Name="SubMenuBorder" BorderBrush="#FF999999" BorderThickness="1" Background="#FFF0F0F0" Padding="2">

<ScrollViewer x:Name="SubMenuScrollViewer" Style="{DynamicResource {ComponentResourceKey ResourceId=MenuScrollViewer, TypeInTargetAssembly={x:Type FrameworkElement}}}">

<Grid RenderOptions.ClearTypeHint="Enabled">

<Canvas HorizontalAlignment="Left" Height="0" VerticalAlignment="Top" Width="0">

<Rectangle x:Name="OpaqueRect" Fill="{Binding Background, ElementName=SubMenuBorder}" Height="{Binding ActualHeight, ElementName=SubMenuBorder}" Width="{Binding ActualWidth, ElementName=SubMenuBorder}"/>

</Canvas>

<Rectangle Fill="#FFD7D7D7" HorizontalAlignment="Left" Margin="29,2,0,2" Width="1"/>

<ItemsPresenter x:Name="ItemsPresenter" KeyboardNavigation.DirectionalNavigation="Cycle" Grid.IsSharedSizeScope="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" KeyboardNavigation.TabNavigation="Cycle"/>

</Grid>

</ScrollViewer>

</Border>

</Popup>

</Grid>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsSuspendingPopupAnimation" Value="True">

<Setter Property="PopupAnimation" TargetName="PART\_Popup" Value="None"/>

</Trigger>

<Trigger Property="Icon" Value="{x:Null}">

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsChecked" Value="True">

<Setter Property="Visibility" TargetName="GlyphPanel" Value="Visible"/>

<Setter Property="Visibility" TargetName="Icon" Value="Collapsed"/>

</Trigger>

<Trigger Property="IsHighlighted" Value="True">

<Setter Property="Background" TargetName="templateRoot" Value="Blue"/>

<!--<Setter Property="BorderBrush" TargetName="templateRoot" Value="#FF26A0DA"/>-->

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="TextElement.Foreground" TargetName="templateRoot" Value="#FF707070"/>

<Setter Property="Fill" TargetName="GlyphPanel" Value="#FF707070"/>

</Trigger>

<Trigger Property="CanContentScroll" SourceName="SubMenuScrollViewer" Value="False">

<Setter Property="Canvas.Top" TargetName="OpaqueRect" Value="{Binding VerticalOffset, ElementName=SubMenuScrollViewer}"/>

<Setter Property="Canvas.Left" TargetName="OpaqueRect" Value="{Binding HorizontalOffset, ElementName=SubMenuScrollViewer}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Window.Resources>

<StackPanel>

<Menu>

<MenuItem Header="\_File" Style="{StaticResource menuParentStyle}" ItemContainerStyle="{DynamicResource MenuItemStyle1}" Template="{DynamicResource MenuItemControlTemplate2}" >

<MenuItem Header="\_New" Click="MenuItem\_Click" />

<MenuItem Header="\_Save"/>

<MenuItem Header="\_Quit" />

</MenuItem>

<MenuItem Header="\_Edit" Style="{StaticResource menuParentStyle}" ItemContainerStyle="{DynamicResource MenuItemStyle1}" Template="{DynamicResource MenuItemControlTemplate2}">

<MenuItem Header="\_Copy"/>

<MenuItem Header="\_Cut" />

<MenuItem Header="\_Paste" />

</MenuItem>

</Menu>

</StackPanel>

</Window>

**Operator C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for Operator.xaml

/// </summary>

public partial class Operator : Page

{

int idsaleOperator = 0;

string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public Operator()

{

InitializeComponent();

comboxox();

gridfill();

}

private void comboxox()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM company", mysqlCon);

DataSet ds = new DataSet();

category\_data.Fill(ds, "company");

CbCompany.DataContext = ds.Tables["company"].DefaultView;

CbCompany.DisplayMemberPath = "companyName";

CbCompany.SelectedIndex = 0;

}

}

void gridfill()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("OperatorView", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtoperators = new DataTable();

mySqlDa.Fill(dtoperators);

grView.DataContext = dtoperators;

// grView.DataContext = dtoperators.DefaultView;

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("OperatorAddOrupdate", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idsaleOperator", idsaleOperator);

mySqlCmd.Parameters.AddWithValue("\_saleOperatorName", name.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorUserName", userName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorPassword", pass.Password.Trim());

mySqlCmd.Parameters.AddWithValue("\_idcompany", CbCompany.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void grView\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

// name.Text = grView.SelectedCells[1].ToString();

// pass.Password = grView.SelectedCells[2].ToString();

DataRowView rowView = grView.SelectedItem as DataRowView;

name.Text = rowView.Row[0].ToString();

userName.Text = rowView.Row[1].ToString();

pass.Password = rowView.Row[2].ToString();

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row = (DataRowView)grView.SelectedItem;

int selectedIDFromDataGrid = (int)row["idsaleOperator"];

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("deleteOperator", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idsaleOperator", selectedIDFromDataGrid);

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row = (DataRowView)grView.SelectedItem;

int selectedIDFromDataGrid = (int)row["idsaleOperator"];

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("OperatorAddOrupdate", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idsaleOperator", selectedIDFromDataGrid);

mySqlCmd.Parameters.AddWithValue("\_saleOperatorName", name.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorUserName", userName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorPassword", pass.Password.Trim());

mySqlCmd.Parameters.AddWithValue("\_idcompany", CbCompany.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("OperatorAddOrupdate", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idsaleOperator", idsaleOperator);

mySqlCmd.Parameters.AddWithValue("\_saleOperatorName", name.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorUserName", userName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_saleOperatorPassword", pass.Password.Trim());

mySqlCmd.Parameters.AddWithValue("\_idcompany", CbCompany.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Submitted Successfully!");

gridfill();

}

}

private void CbCompany\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

}

**Operator WPF interface code:**

<Page x:Class="Bank.Operator"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Operator">

<Grid Margin="0,0,-588,-107">

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<TextBox Name="name" HorizontalAlignment="Left" Height="44" Margin="288,23,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="180"/>

<Label FontSize="22" Content="Име" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="30,28,0,0"/>

<TextBox x:Name="userName" HorizontalAlignment="Left" Height="40" Margin="288,81,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="180" />

<Label FontSize="22" Content="Потребителско име" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="30,77,0,0" Height="64" Width="234"/>

<Label FontSize="22" Content="Парола" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="30,141,0,0" Height="46" Width="104"/>

<ComboBox ItemsSource="{Binding}" Name="CbCompany" HorizontalAlignment="Left" Margin="288,221,0,0" VerticalAlignment="Top" Width="180" Height="49" SelectionChanged="CbCompany\_SelectionChanged" />

<Label FontSize="22" Content="Компания" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="30,221,0,0" Height="49" Width="131"/>

<Button Background="Green" Opacity="0.7" x:Name="bt1" Content="Запази" HorizontalAlignment="Left" Margin="73,394,0,0" VerticalAlignment="Top" Width="120" Click="Button\_Click" Height="38"/>

<PasswordBox x:Name="pass" HorizontalAlignment="Left" Margin="288,149,0,0" VerticalAlignment="Top" Width="180" Height="38"/>

<DataGrid ItemsSource="{Binding}" x:Name="grView" HorizontalAlignment="Left" Height="367" VerticalAlignment="Top" Width="616" Margin="772,28,0,0" MouseDoubleClick="grView\_MouseDoubleClick" CanUserAddRows="False" IsReadOnly="True"/>

<Button Background="Red" Opacity="0.5" Content="Изтриване" HorizontalAlignment="Left" Margin="505,397,0,0" VerticalAlignment="Top" Width="126" Height="35" Click="Button\_Click\_2"/>

<Button Background="Blue" Opacity="0.5" Content="Обнови" HorizontalAlignment="Left" Margin="243,396,0,0" VerticalAlignment="Top" Width="111" Height="36" Click="Button\_Click\_3"/>

</Grid>

</Page>

**SaleCar C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for SaleCar.xaml

/// </summary>

public partial class SaleCar : Page

{

public int idSale = 0;

public string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public SaleCar()

{

InitializeComponent();

gridfillClients();

gridfillCars();

gridfillOperator();

gridfillSale();

}

public void gridfillCars()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("carView", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtcarss = new DataTable();

mySqlDa.Fill(dtcarss);

gvCars.DataContext = dtcarss;

mysqlCon.Close();

}

}

public void gridfillClients()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("clientView2", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtClientss = new DataTable();

mySqlDa.Fill(dtClientss);

gvClients.DataContext = dtClientss;

mysqlCon.Close();

}

}

public void gridfillOperator()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("ViewOperator2", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtoperatorss = new DataTable();

mySqlDa.Fill(dtoperatorss);

gvOperators.DataContext = dtoperatorss;

// grView.DataContext = dtoperators.DefaultView;

mysqlCon.Close();

}

}

public void gridfillSale()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("saleView", mysqlCon);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtsale = new DataTable();

mySqlDa.Fill(dtsale);

gvSale.DataContext = dtsale;

// grView.DataContext = dtoperators.DefaultView;

mysqlCon.Close();

}

}

private void TextBox\_KeyDown(object sender, KeyEventArgs e)

{

(gvCars.DataContext as DataTable).DefaultView.RowFilter = string.Format("carModel LIKE '%{0}%'", tbSearch.Text);

}

private void tbSearchClient\_KeyDown(object sender, KeyEventArgs e)

{

(gvClients.DataContext as DataTable).DefaultView.RowFilter = string.Format("customerIDNum LIKE '%{0}%'", tbSearchClient.Text);

}

private void tbSearchOperator\_KeyDown(object sender, KeyEventArgs e)

{

(gvOperators.DataContext as DataTable).DefaultView.RowFilter = string.Format("saleOperatorName LIKE '%{0}%'", tbSearchOperator.Text);

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

DataRowView row1 = (DataRowView)gvCars.SelectedItem;

int selectedIDFromCar = (int)row1["idCar"];

DataRowView row2 = (DataRowView)gvOperators.SelectedItem;

int selectedIDFromDataGrid = (int)row2["idsaleOperator"];

DataRowView row3 = (DataRowView)gvClients.SelectedItem;

int selectedIDFromCustomers = (int)row3["idcustomer"];

//MessageBox.Show(""+selectedIDFromCar+selectedIDFromDataGrid+ selectedIDFromCustomers+"");

var date = Convert.ToDateTime(dpDate.Text).ToString("yyyy/MM/dd");

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("AddUpdateSale", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_idSale", idSale);

mySqlCmd.Parameters.AddWithValue("\_SaleDate", date);

mySqlCmd.Parameters.AddWithValue("\_car\_idCar", selectedIDFromCar);

mySqlCmd.Parameters.AddWithValue("\_saleOperator\_idsaleOperator", selectedIDFromDataGrid);

mySqlCmd.Parameters.AddWithValue("\_customer\_idcustomer", selectedIDFromCustomers);

this.Dispatcher.Invoke(() =>

{

mySqlCmd.ExecuteNonQuery();

});

MessageBox.Show("Submitted Successfully!");

gridfillSale();

mysqlCon.Close();

}

}

}

}

**SaleCar WPF interface code:**

<Page x:Class="Bank.SaleCar"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="623.466"

Title="SaleCar" Width="1664.146">

<Grid Margin="0,0,-588,-107">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="677\*"/>

<ColumnDefinition Width="449\*"/>

</Grid.ColumnDefinitions>

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<DataGrid ItemsSource="{Binding}" x:Name="gvCars" HorizontalAlignment="Left" Height="187" Margin="827,85,0,0" VerticalAlignment="Top" Width="488"/>

<TextBox x:Name="tbSearch" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="986,56,0,0" KeyDown="TextBox\_KeyDown"/>

<Label FontSize="22" Content="Филтрирай марка" HorizontalAlignment="Left" Margin="953,17,0,0" VerticalAlignment="Top" Height="41" Width="229"/>

<Label FontSize="22" Content="Въведете дата на продажба" HorizontalAlignment="Left" Margin="28,311,0,0" VerticalAlignment="Top" RenderTransformOrigin="-4.514,-3.887" Height="66" Width="294"/>

<DataGrid ItemsSource="{Binding}" x:Name="gvClients" HorizontalAlignment="Left" Height="187" Margin="384,89,0,0" VerticalAlignment="Top" Width="381"/>

<Label FontSize="22" Content="Филтрирай клиент" HorizontalAlignment="Left" Margin="472,15,0,0" VerticalAlignment="Top" Height="41" Width="211"/>

<TextBox x:Name="tbSearchClient" HorizontalAlignment="Left" Height="23" Margin="517,56,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" KeyDown="tbSearchClient\_KeyDown"/>

<DataGrid ItemsSource="{Binding}" x:Name="gvOperators" HorizontalAlignment="Left" Height="187" Margin="10,89,0,0" VerticalAlignment="Top" Width="355"/>

<TextBox x:Name="tbSearchOperator" HorizontalAlignment="Left" Height="23" Margin="141,56,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" KeyDown="tbSearchOperator\_KeyDown"/>

<Label FontSize="22" Content="Филтрирай Оператор" HorizontalAlignment="Left" Margin="84,15,0,0" VerticalAlignment="Top" Height="41" Width="238"/>

<Button FontSize="22" Content="Продай" HorizontalAlignment="Left" Margin="112,477,0,0" VerticalAlignment="Top" Width="121" Click="Button\_Click" Height="36"/>

<DatePicker FontSize="18" x:Name="dpDate" SelectedDate="{Binding myVideModelProperty}"

Height="25" HorizontalAlignment="Left" Margin="84,352,0,0" VerticalAlignment="Top" Width="238">

<DatePicker.Resources>

<Style TargetType="{x:Type DatePickerTextBox}">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<TextBox x:Name="PART\_TextBox"

Text="{Binding Path=SelectedDate, RelativeSource={RelativeSource AncestorType={x:Type DatePicker}}, StringFormat={}{0:yyyy/MM/dd}}" />

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</DatePicker.Resources>

</DatePicker>

<DataGrid ItemsSource="{Binding}" x:Name="gvSale" HorizontalAlignment="Left" Height="199" Margin="401,347,0,0" VerticalAlignment="Top" Width="705"/>

</Grid>

</Page>

**Spravka1 C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for Spravka1.xaml

/// </summary>

public partial class Spravka1 : Page

{

public DataSet ds = new DataSet();

public Spravka1()

{

InitializeComponent();

//gridfillCars();

comboxox();

}

public string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public int selectedUser = 0;

public void gridfillCars()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

try{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("spravkaOperatori2", mysqlCon);

selectedUser = CbCompany.SelectedIndex+1;

mySqlDa.SelectCommand.Parameters.AddWithValue("\_idsaleOperator", selectedUser);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtcarss = new DataTable();

mySqlDa.Fill(dtcarss);

gvCars.DataContext = dtcarss;

mysqlCon.Close();

// String id = (String)ds.Tables[1].Rows[CbCompany.SelectedIndex]["idsaleOperator"];

// Console.WriteLine(ds);

}

catch (Exception e)

{

}

}

}

private void comboxox()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM saleoperator", mysqlCon);

category\_data.Fill(ds, "saleoperator");

CbCompany.DataContext = ds.Tables["saleoperator"].DefaultView;

CbCompany.DisplayMemberPath = "saleOperatorName";

Console.WriteLine(ds);

CbCompany.SelectedIndex = 0;

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

PrintDialog printDialog = new PrintDialog();

if (printDialog.ShowDialog() == true)

{

printDialog.PrintVisual(gvCars, "My First Print Job");

}

}

private void CbCompany\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

gridfillCars();

}

}

}

**Spravka2 WPF interface code:**

<Page x:Class="Bank.Spravka1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Spravka1">

<Grid>

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button FontSize="22" Content="Принтирай!" HorizontalAlignment="Left" Margin="321,336,0,0" VerticalAlignment="Top" Width="121" Click="Button\_Click" Height="36"/>

<ComboBox ItemsSource="{Binding}" Name="CbCompany" HorizontalAlignment="Left" Margin="295,58,0,0" VerticalAlignment="Top" Width="180" Height="49" SelectionChanged="CbCompany\_SelectionChanged" />

<DataGrid ItemsSource="{Binding}" x:Name="gvCars" HorizontalAlignment="Left" Height="187" Margin="153,126,0,0" VerticalAlignment="Top" Width="488"/>

</Grid>

</Page>

**Spravka2 C# code:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Bank

{

/// <summary>

/// Interaction logic for spravka2.xaml

/// </summary>

public partial class spravka2 : Page

{

public spravka2()

{

InitializeComponent();

//gridfillCars();

comboxox();

}

public string myconnstrng = "server=localhost;user=root;database=saledb;port=3306;password=123456";

public int selectedUser = 0;

public DataSet ds = new DataSet();

public void gridfillCars()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

try

{

mysqlCon.Open();

MySqlDataAdapter mySqlDa = new MySqlDataAdapter("spravka2", mysqlCon);

selectedUser = CbCompany.SelectedIndex + 1;

mySqlDa.SelectCommand.Parameters.AddWithValue("\_idCompany", selectedUser);

mySqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtcarss = new DataTable();

mySqlDa.Fill(dtcarss);

gvCars.DataContext = dtcarss;

mysqlCon.Close();

// String id = (String)ds.Tables[1].Rows[CbCompany.SelectedIndex]["idsaleOperator"];

// Console.WriteLine(ds);

}

catch (Exception e)

{

}

}

}

private void comboxox()

{

using (MySqlConnection mysqlCon = new MySqlConnection(myconnstrng))

{

mysqlCon.Open();

MySqlDataAdapter category\_data = new MySqlDataAdapter("SELECT \* FROM company", mysqlCon);

category\_data.Fill(ds, "company");

CbCompany.DataContext = ds.Tables["company"].DefaultView;

CbCompany.DisplayMemberPath = "companyName";

CbCompany.SelectedIndex = 0;

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

PrintDialog printDialog = new PrintDialog();

if (printDialog.ShowDialog() == true)

{

printDialog.PrintVisual(gvCars, "My First Print Job");

}

}

private void CbCompany\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

gridfillCars();

}

}

}

**Spravka2 WPF interface code:**

<Page x:Class="Bank.spravka2"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Bank"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="spravka2">

<Grid>

<Grid.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="#7be9f6" Offset="0.0"/>

<GradientStop Color="#f083da" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button FontSize="22" Content="Принтирай!" HorizontalAlignment="Left" Margin="69,336,0,0" VerticalAlignment="Top" Width="121" Click="Button\_Click" Height="36"/>

<ComboBox ItemsSource="{Binding}" Name="CbCompany" HorizontalAlignment="Left" Margin="49,127,0,0" VerticalAlignment="Top" Width="180" Height="49" SelectionChanged="CbCompany\_SelectionChanged" />

<DataGrid ItemsSource="{Binding}" x:Name="gvCars" HorizontalAlignment="Left" Height="187" Margin="270,94,0,0" VerticalAlignment="Top" Width="488"/>

</Grid>

</Page>