MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Centrul de Excelenţă în Informatică şi Tehnologii Informaţionale

RAPORT

LA PRACTICA TEHNOLOGICĂ

SPECIALITATEA "Programare şi analiza produselor program"

TEMA : "Учет продажи техники в компьютерном магазине *Centurion*"

A elaborat elevul

Боцан Николай, P-2033r

Conducătorul practicii

Şarapanovscaia Irina

Chişinău 2023

Содержание

Введение.

Общее описание практики:

Практика проходила в колледже, в 104 кабинете, в очень теплой дружилебной атмосфере. Мне необходимо было спроектировать и нормализовать базу данных компьютерного магазина, написать для неё приложение на C# с возможностью авторизации, что мне вполне удалось.

Сначала мне было необходимо составить словесное описание предметной области, по которой в последствии я составил ER-диаграмму. Далее эта ER-диаграмма была использована для создания реляционной модели и, уже после её нормализации, я приступил к составлению физицеской модели, а именно кода.

После того, как База Данных была составлена я приступил к написанию программы на C#. Сначала я создал главную форму, на которой есть возможность считывать данные из таблиц базы данных (далее БД), вносить, удалать и изменять её. После этого, естественно последовали формочки для работы с этими таблицами (добавление, удаление и редактирование), после чего я преступил к построению итчетов, и в итоге, создал формочку авторизации, и полностью настроил го функционал. Данные о пользователях хранятся в самой БД, а в самой формочке есть возможность войти в аккаунт или зарегистрировать его. Также мною были добавлены возможность поиска, фильтрации информации таблиц и итоги...

Технологическая практика преследует достижение общих целей:

- накопление студентами практических навыков и опыта создания общей структуры визуального приложения;
- настройка компонентов визуального приложения;
- управление компонентами визуального приложения;
- разработка визуального приложения в соответствии с техническими условиями;
- поддержка визуальных приложений;
- использование визуального программирования для разработки приложений;
- изучение механизма подключения базы данных к программируемому приложению;
- применение методов восстановления данных и создание резервных копий;
- соблюдение стандартов, касающихся способа оформления документов.

Период практики

Со 2-го по 26-е мая 2023 года

Время практики

С 8:00 до 11:00

Перечисление использованных инструментов (Visual Studio 2022, SQL Manager Server Studio, Word, Excel, Smart Install Maker, Microsoft Report Viewer)

Постановка задачи

Словесное описание

Клиент приходит в компьютерный магазин с целью купить тот или иной товар. В комьютерном магазине продаются следующие товары: **компьютеры** (**ПК** или **нонтбуки**), **телефоны** (и **планшеты**), **мониторы** и **принтеры**. Данные **товары** обладают следующими характеристиками: название, производитель, цена, гарантия, тип этого товара (ПК, ноутбук, телефот и т.д.), а также характерные этому товару технические характеристики. **Товары** поставляют определенные **поставщики** определенных фирм (**производителей**), а все поставки заносятся в соответствующий журнал, в который заносятся следующие данные: поставщик, сам поставленный товар, его количество, производитель, а также №, дата и цена поставки... У **поставщика** есть имя, адрес, тип поставляемой продукции, телефон, а также тот производитель, чью продукцию он поставляет... **Производитель** же в свою очередь характеризуется именем и адресом. Когда клиент определился с выбором он подходит к кассе, выбирает **тип оплаты** (наличка, кредитная карта или кредит) и предоставляет свои **паспортные данные**, а именно свой адрес, ИДНП и ФИО. После завершения оплаты, покупателю выдается **чек**, в котором указываются тип оплаты, купленный товар, дата покупки и итоговая стоимость...

Техническое задание по проекту

Спроектилвать Базу Данных “учета продажи техноко в компьютерном магазине”

Нормализовать БД

Написать скрипт БД и создать её диаграмму

Создать приложение на C#

Добавить главную форму, с возможностью чтения, добавления и изменения информации таблиц БД

Установить соединение с БД

Создать отчеты по проекту

Создать поск, фильтацию и итоги

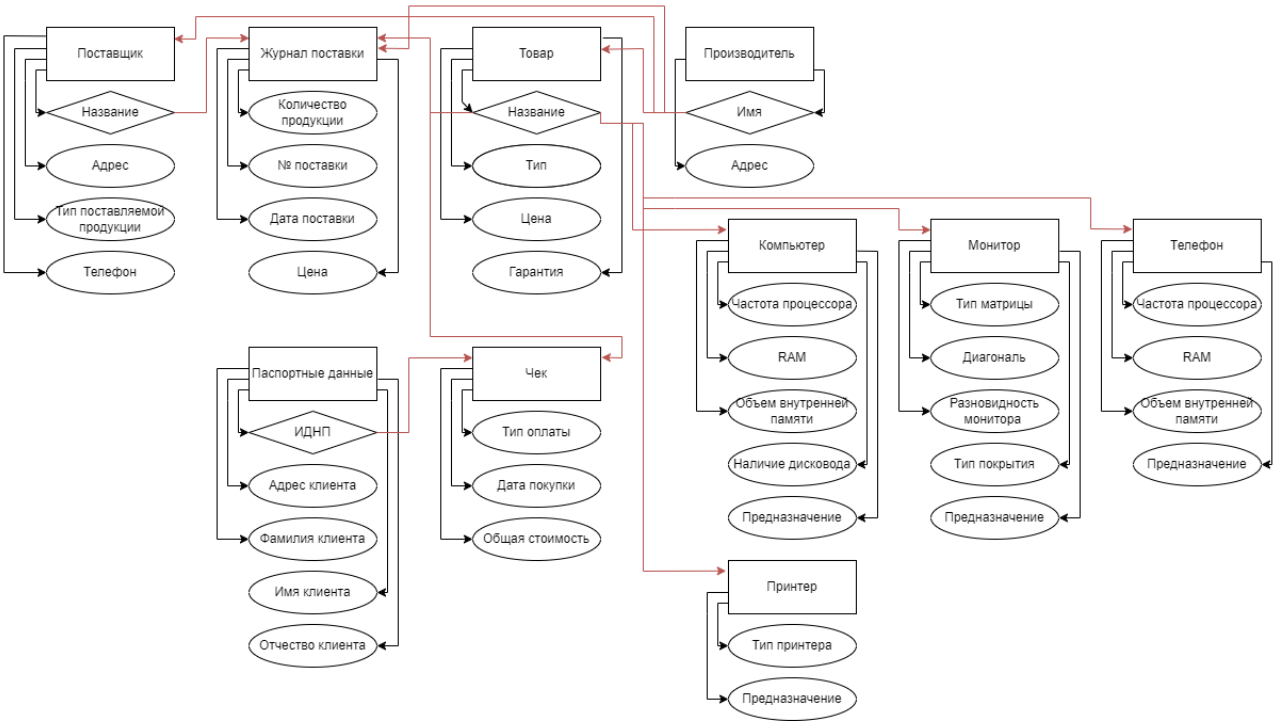
Создать формочку регистрации

Создать возможность создавать резервные копии и восстановления БД

Создать установщик ПО

Проектирование базы данных

ER диаграмма



Реляционная модель

До нормализации (не нормализованная БД)

Parent-Children
Поставщик
Название
Имя производителя
Адрес
Тип поставляемой продукции
Телефон

Children
Журнал поставки
Название поставщика
Имя производителя
Название продукции
Кол-во продукции
№ Поставки
Дата поставки
Цена

Parent
Производитель
Имя
Адрес

Children
Компьютер
Название товара
Частота процессора
RAM
Объем внутренней памяти
Наличие дисковода
Предназначение

Parent-Children
Товар
Имя производителя
Название
Тип товара
Цена
Гарантия

Children
Телефон
Название товара
Частота процессора
RAM
Объем внутренней памяти
Предназначение

Parent
Паспортные данные
ИДНП
Адрес клиента
Фамилия клиента
Имя клиента
Отчество клиента

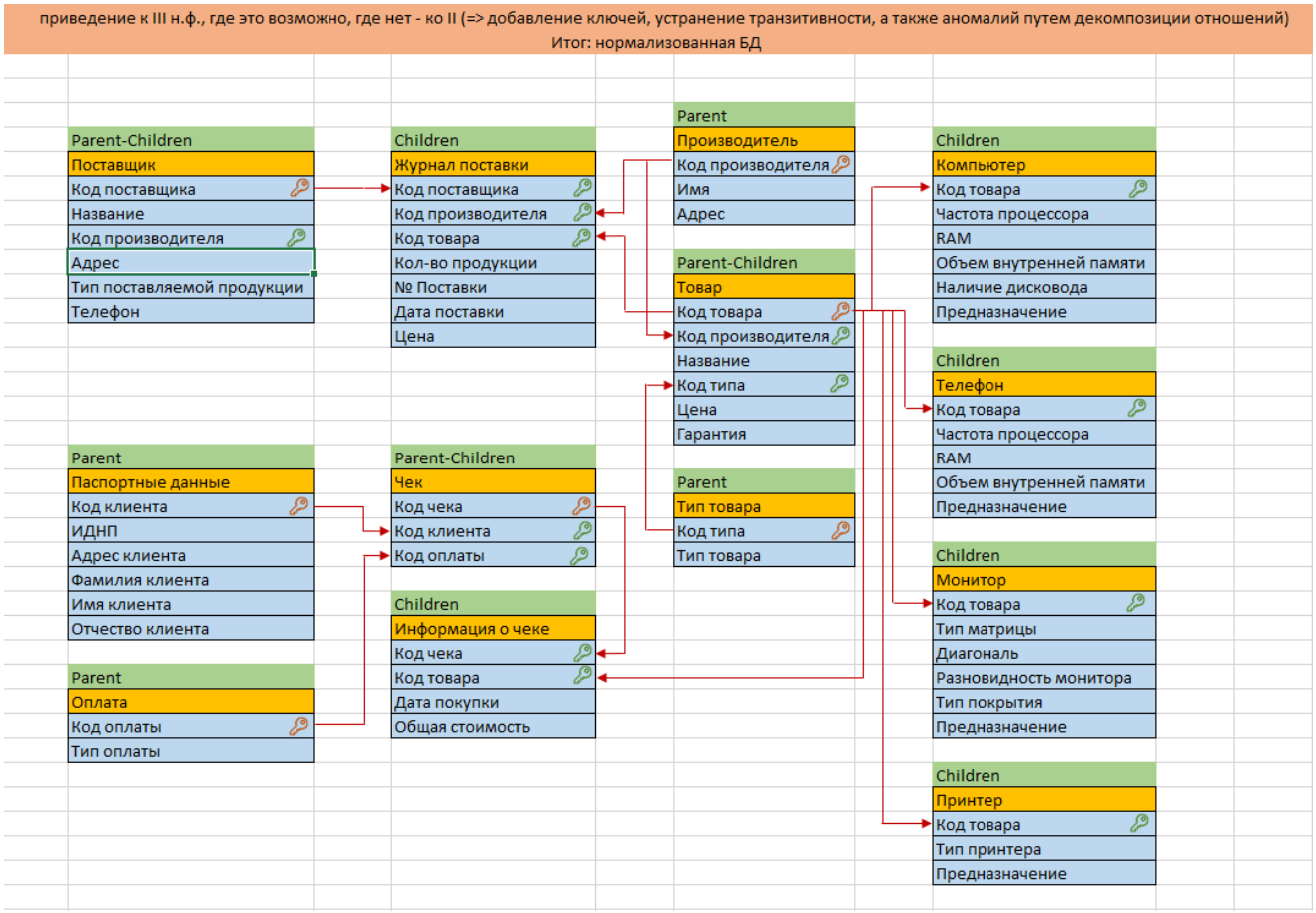
Children
Чек
ИДНП
Название товара
Тип оплаты
Дата покупки
Общая стоимость

Children
Монитор
Название товара
Тип матрицы
Диагональ
Разновидность монитора
Тип покрытия
Предназначение

Children
Принтер
Название товара
Тип принтера
Предназначение

Нормализация

Нормализауия заключается в приведении к III н.ф., где это возможно, где нет - ко II (=> добавление ключей, устранении транзитивности (все атрибуты должны зависить только от первичного ключа, без каких-либо коссвенных связей), а также аномалий путем декомпозиции отношений)



Физическое описание

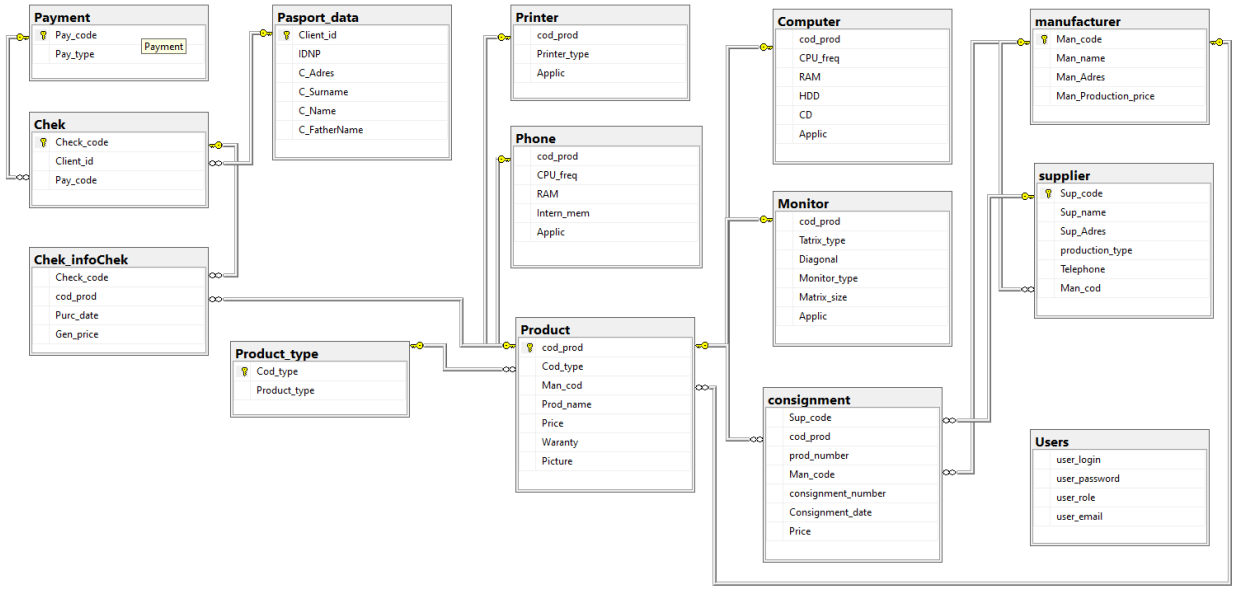
Отношение	Первичный ключ	Вторичный ключ	Атрибуты
Паспортные Данные	Код клиента	---	➢ Код клиента ➢ ИДНП ➢ Адрес клиента ➢ Фамилия клиента ➢ Имя клиента ➢ Отчество клиента
Оплата	Код оплаты	---	➢ Код оплаты ➢ Тип оплаты
Производитель	Код производителя	---	➢ Код производителя ➢ Имя (производителя) ➢ Адрес (производителя)

Тип товара	Код типа (товара)	---	➢ Код типа ➢ Тип товара
Поставщик	Код поставщика	Код производителя	➢ Код поставщика ➢ Название ➢ Код производителя ➢ Адрес ➢ Тип поставляемой продукции ➢ Телефон
Чек	Код чека	Код клиента, Код оплаты	➢ Код чека ➢ Код клиента ➢ Код оплаты
Товар	Код товара	Код производителя, Код типа	➢ Код товара ➢ Код производителя ➢ Название ➢ Код типа ➢ Цена ➢ Гарантия ➢ Изображение
Журнал поставки	(№ Поставки)	Код поставщика, Код производителя, Код товара	➢ Код поставщика ➢ Код производителя ➢ Код товара ➢ Количество продукции ➢ № Поставки ➢ Дата поставки ➢ Цена
Информация о чеке	---	Код чека, Код товара	➢ Код чека ➢ Код товара ➢ Дата покупки ➢ Общая стоимость
Компьютер	---	Код товара	➢ Код товара ➢ Частота процессора ➢ RAM ➢ Объем внутренней памяти ➢ Наличие дисковода ➢ Преднозначение
Телефон	---	Код товара	➢ Код товара ➢ Частота процессора ➢ RAM ➢ Объем внутренней

			памяти ➢ Преднозначение
Монитор	---	Код товара	➢ Код товара ➢ Тип матрицы ➢ Диагональ ➢ Разновидность монитора ➢ Тип покрытия (экрана монитора) ➢ Преднозначение
Принтер	---	Код товара	➢ Код товара ➢ Тип принтера ➢ Предназначение

Создание Базы данных

Диаграмма Базы данных



Таблицы Базы данных

Паспортные Данные

Results Messages

	Client_id	IDNP	C_Adres	C_Sumame	C_Name	C_FatherName
1	1001	1111111111111	str. Puschin 6/7	Balconschii	Andrei	Nicolai
2	1002	2222222222222	str. Nicolae Milescu Spatarul 21/1	Bezuhov	Pier	Anatol
3	1003	3333333333333	str. Sadoveanu 13/5	Rostov	Nicolai	Ilie
4	1004	4444444444444	str. Petru Zadnipru 28/8	Boliakin	Andrei	Nikifor
5	1005	5555555555555	str. Muncesti 76/9	Shiskin	Alexandr	Igor

Оплата

Results Messages

	Pay_code	Pay_type
1	101	credit
2	102	credit card
3	103	cash

Производитель

Results Messages

	Man_code	Man_name	Man_Adres	Man_Production_price
1	4001	Xiaomi	str. Petru Movila 8	2000
2	4002	Samsung	str. Uzinelor 10	3000
3	4003	LG	str. Muncesti 7	3000
4	4004	HP	str. Gogol 36	2000
5	4005	Hyper PC	str. Mihail Sadoveanu	4000

Тип товара

Results Messages

	Cod_type	Product_type
1	2000	PC
2	2010	Laptop
3	2020	Monitor
4	2030	Phone
5	2040	Tablet PC
6	2050	Printer

Поставщик

Results Messages

	Sup_code	Sup_name	Sup_Adres	production_type	Telephone	Man_cod
1	3001	Comp_Moldova	str. Stefan cel mare 21/5	PCs, Laptops, Monitors, Phones	+37367354672	4003
2	3002	Hyper Supply	str. Gogol 34/4	PCs, Monitors, Printers, Laptops	+37369984519	4005
3	3003	GMSupply	str. Columna 5	PCs, Laptops	+37378534765	4003
4	3004	TechMD	str. Uzinelor 7	Monitors, Phones, Laptops	+37379945543	4002
5	3005	XMoldova	str. Petru Movila 7	Laptops, Phones, Printers, Monitors	+37360606607	4001
6	3006	HPSupply	str. Mihail Sadoveanu 3	PCs, Monitors, Printers	+37369984519	4004

Чек

Results Messages

	Check_code	Client_id	Pay_code
1	1013	1001	101
2	1014	1003	102
3	1017	1005	103
4	1020	1002	103
5	1022	1004	102

Товар

Results Messages

	cod_prod	Cod_type	Man_cod	Prod_name	Price	Waranty	Picture
1	5000	2000	4005	Blye Yeti	20000	5	../../resources/DataBase/images/blueYeti.png
2	5010	2010	4003	Nitro	25000	3	../../resources/DataBase/images/Nitro.png
3	5020	2020	4003	Spin 5	5000	1	../../resources/DataBase/images/spin5.png
4	5021	2020	4004	Hp one	2000	1	../../resources/DataBase/images/HPOne.png
5	5030	2030	4002	Redmi 5 plus	4000	1	../../resources/DataBase/images/redmi5plus.png
6	5040	2040	4001	Prestige Pro 3	4700	2	../../resources/DataBase/images/Prestige.png
7	5050	2050	4004	All Vew Pro	2000	3	../../resources/DataBase/images/SamsungAllView.png
8	5051	2050	4001	Clear Sky	3000	4	../../resources/DataBase/images/CanonClearSky.png
9	5052	2050	4001	Colors	2500	2	../../resources/DataBase/images/EpsonColors.png

Журнал поставки

Results Messages

	Sup_code	cod_prod	prod_number	Man_code	consignment_number	Consignment_date	Price
1	3002	5000	5	4005	10456	2021-12-21	17000
2	3003	5010	3	4003	10432	2022-01-04	14000
3	3001	5020	4	4003	10487	2021-10-10	3000
4	3006	5021	7	4004	10499	2022-01-09	1000
5	3004	5030	10	4002	10444	2022-02-27	2500
6	3005	5040	8	4001	10411	2022-01-01	3000
7	3006	5050	11	4004	10410	2022-02-04	1500
8	3005	5051	15	4001	10440	2022-02-02	2000
9	3005	5052	12	4001	10404	2021-12-13	1700

Информация о чеке

Results Messages

	Check_code	cod_prod	Purc_date	Gen_price
1	1013	5000	2021-12-31	20000
2	1014	5052	2021-12-31	2500
3	1017	5040	2022-01-08	4700
4	1020	5021	2022-01-09	2000
5	1022	5030	2022-02-28	4000

Компьютер

Results Messages

	cod_prod	CPU_freq	RAM	HDD	CD	Applic
1	5000	4200	16000	1000	false	Gaming
2	5010	3900	8000	2500	true	Gaming

Телефон

Results Messages

	cod_prod	CPU_freq	RAM	Intern_mem	Applic
1	5030	2000	3000	32000	Daily
2	5040	3000	4000	32000	Graphic design

Монитор

Results Messages

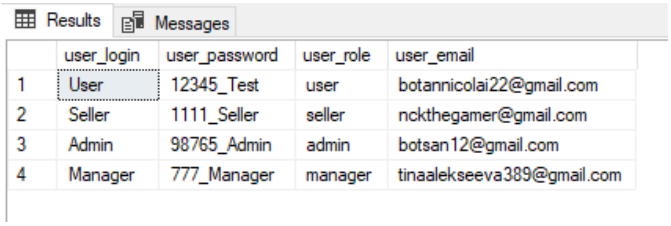
	cod_prod	Tatrix_type	Diagonal	Monitor_type	Matrix_size	Applic
1	5020	IPS	21	with rotation	mate	Gaming
2	5021	TN	18	without rotation	glossy	Office

Принтер

Results Messages

	cod_prod	Printer_type	Applic
1	5050	laser	Office
2	5051	color	Photos
3	5052	color	Photos

Пользователи



Results | Messages

	user_login	user_password	user_role	user_email
1	User	12345_Test	user	botannicolai22@gmail.com
2	Seller	1111_Seller	seller	nckthegamer@gmail.com
3	Admin	98765_Admin	admin	botsan12@gmail.com
4	Manager	777_Manager	manager	tinaalekseeva389@gmail.com

Команды SQL создания БД и таблиц

```
--Создание контейнера базы данных--
create database Computer_magazine
go

--Активизирование контейнера базы данных--
use Computer_magazine
go

--пользовательские типы данных--
create type Adres from varchar(70);
go

create type applicationS from varchar(100);
go

create type TNumber from char(12);
go


--Создание отношений--

--Родительские--

--Тип продкута--
create table Product_type(Cod_type smallint primary key,
                                                Product_type varchar(20))
go

insert into Product_type(Cod_type, Product_type)
values (2000, 'PC'),
          (2010, 'Laptop'),
          (2020, 'Monitor'),
          (2030, 'Phone'),
          (2040, 'Tablet PC'),
          (2050, 'Printer')
go


--Производитель--
create table manufacturer(Man_code int primary key,
                                                Man_name varchar(20),
                                                Man_Adres Adres,
                                                Man_Production_price smallint)
go

insert into manufacturer(Man_code, Man_name, Man_Adres, Man_Production_price)
values  (4001, 'Xiaomi', 'str. Petru Movila 8', 2000),
              (4002, 'Samsung', 'str. Uzinelor 10', 3000),
              (4003, 'LG', 'str. Muncesti 7', 3000),
              (4004, 'HP', 'str. Gogol 36', 2000),
              (4005, 'Hyper PC', 'str. Mihail Sadoveanu', 4000)
go


--Оплата--
create table Payment(Pay_code int primary key,
```

```
                                    Pay_type varchar(15) default 'cash')
go

insert into Payment(Pay_code, Pay_type)
values  (101, 'credit'),
              (102, 'credit card'),
              (103, 'cash')
go


--Паспортные данные--
create table Pasport_data(Client_id int primary key,
                                                IDNP char(13) unique not null,
                                                C_Adres Adres,
                                                C_Surname varchar(12),
                                                C_Name varchar(12),
                                                C_FatherName varchar(12))
go

insert into Pasport_data(Client_id, IDNP, C_Adres, C_Surname, C_Name, C_FatherName)
values (1001, '1111111111111', 'str. Puschin 6/7', 'Balconschii', 'Andrei', 'Nicolai'),
          (1002, '2222222222222', 'str. Nicolae Milescu Spatarul 21/1', 'Bezuhov', 'Pier',
'Anatol'),
          (1003, '3333333333333','str. Sadoveanu 13/5', 'Rostov', 'Nicolai', 'Ilie'),
          (1004, '4444444444444','str. Petru Zadnipru 28/8', 'Boliakin', 'Andrei', 'Nikifor'),
          (1005, '5555555555555','str. Muncesti 76/9', 'Shiskin', 'Alexandr', 'Igor')
go



--Родительски-дочерние--

--Поставщик--
create table supplier(Sup_code int primary key,
                                    Sup_name varchar(20),
                                    Sup_Adres Adres,
                                    production_type varchar(40),
                                    Telephone TNumber,
                                    Man_cod int foreign key references manufacturer(Man_code)
not null)
go

insert into supplier(Sup_code, Sup_name, Man_cod, Sup_Adres, production_type, Telephone)
values  (3001, 'Comp_Moldova', 4003, 'str. Stefan cel mare 21/5', 'PCs, Laptops, Monitors,
Phones', '+37367354672'),
              (3002, 'Hyper Supply', 4005, 'str. Gogol 34/4', 'PCs, Monitors, Printers, Laptops',
'+37369984519'),
              (3003, 'GMSupply', 4003, 'str. Columna 5', 'PCs, Laptops', '+37378534765'),
              (3004, 'TechMD', 4002, 'str. Uzinelor 7', 'Monitors, Phones, Laptops',
'+37379945543'),
              (3005, 'XMoldova', 4001, 'str. Petru Movila 7', 'Laptops, Phones, Printers,
Monitors', '+37360606607'),
              (3006, 'HPSupply', 4004, 'str. Mihail Sadoveanu 3', 'PCs, Monitors, Printers',
'+37369984519')
go

--Чек--
create table Chek(Check_code int primary key,
                            Client_id int foreign key references Pasport_data(Client_id) not
null,
                            Pay_code int foreign key references Payment(Pay_code) not null)
go

insert into Chek(Check_code, Client_id, Pay_code)
values  (1013, 1001, 101),
              (1014, 1003, 102),
              (1017, 1005, 103),
              (1020, 1002, 103),
              (1022, 1004, 102)
```

```
go


--Товар--
create table Product(cod_prod int primary key,
                                    Cod_type smallint foreign key references
Product_type(Cod_type) not null,
                                    Man_cod int foreign key references manufacturer(Man_code)
not null,
                                    Prod_name varchar(20),
                                    Price smallint,
                                    Waranty smallint check(Waranty <= 5) default 1,
                                    Picture varchar(100) not null unique)
go

insert into Product(cod_prod, Cod_type, Man_cod, Prod_name, Price, Waranty, Picture)
values  (5000, 2000, 4005, 'Blye Yeti', 20000, 5,
'../../resources/DataBase/images/blueYeti.png'),
              (5010, 2010, 4003, 'Nitro', 25000 ,3, '../../resources/DataBase/images/Nitro.png'),
              (5020, 2020, 4003, 'Spin 5', 5000, 1, '../../resources/DataBase/images/spin5.png'),
              (5021, 2020, 4004, 'Hp one', 2000, 1, '../../resources/DataBase/images/HPOne.png'),
              (5030, 2030, 4002, 'Redmi 5 plus', 4000, 1,
'../../resources/DataBase/images/redmi5plus.png'),
              (5040, 2040, 4001, 'Prestige Pro 3', 4700, 2,
'../../resources/DataBase/images/Prestige.png'),
              (5050, 2050, 4004, 'All Vew Pro', 2000, 3,
'../../resources/DataBase/images/SamsungAllView.png'),
              (5051, 2050, 4001, 'Clear Sky', 3000, 4,
'../../resources/DataBase/images/CanonClearSky.png'),
              (5052, 2050, 4001, 'Colors', 2500, 2,
'../../resources/DataBase/images/EpsonColors.png')

go


--Дочерние--

--Компьютер--
create table Computer(cod_prod int foreign key references Product(cod_prod) unique not null, --
связь 1 к 1--
                                     CPU_freq smallint default 2000,
                                     RAM smallint default 1000,
                                     HDD smallint default 250,
                                     CD varchar(5) check((CD = 'true') or (CD = 'false'))
default 'false',
                                     Applic applicationS)
go

insert into Computer(cod_prod, CPU_freq, RAM, HDD, CD, Applic)
values  (5000, 4200, 16000, 1000, 'false', 'Gaming'),
              (5010, 3900, 8000, 2500, 'true', 'Gaming')

go


--Монитор--
create table Monitor(cod_prod int foreign key references Product(cod_prod) unique not null,
                                    Tatrix_type varchar(5) default 'TN',
                                    Diagonal smallint default 21,
                                    Monitor_type varchar(20) default 'without rotation',
                                    Matrix_size varchar(10) default 'glossy',
                                    Applic applicationS)
go

insert into Monitor(cod_prod, Tatrix_type, Diagonal, Monitor_type, Matrix_size, Applic)
values (5020, 'IPS', 21, 'with rotation', 'mate', 'Gaming'),
          (5021, 'TN', 18, 'without rotation', 'glossy','Office')

go
```

```
--Телефон--
create table Phone(cod_prod int foreign key references Product(cod_prod) unique not null,
                                CPU_freq smallint default 1000,
                                RAM smallint default 1000,
                                Intern_mem smallint default 5000,
                                Applic applicationS)
go

insert into Phone(cod_prod, CPU_freq, RAM, Intern_mem, Applic)
values (5030, 2000, 3000, 32000, 'Daily'),
          (5040, 3000, 4000, 32000, 'Graphic design')

go


--Принтер--
create table Printer(cod_prod int foreign key references Product(cod_prod) unique not null,
                                    Printer_type varchar(6) check((Printer_type = 'color') or
(Printer_type = 'laser')) default 'laser',
                                    Applic applicationS)
go

insert into Printer(cod_prod, Printer_type, Applic)
values  (5050, 'laser', 'Office'),
              (5051, 'color', 'Photos'),
              (5052, 'color', 'Photos')

go


--Журнал поставки--
create table consignment(Sup_code int foreign key references supplier(Sup_code) not null,
                                        cod_prod int foreign key references
Product(cod_prod) not null,
                                        prod_number smallint,
                                        Man_code int foreign key references
manufacturer(Man_code) not null,
                                        consignment_number int,
                                        Consignment_date date,
                                        Price int)
go

insert into consignment(Sup_code, cod_prod, prod_number, Man_code, consignment_number,
Consignment_date, Price)
values  (3002, 5000, 5, 4005, 10456, '2021-12-21', 17000),
              (3003, 5010, 3, 4003, 10432, '2022-01-04', 14000),
              (3001, 5020, 4, 4003, 10487, '2021-10-10', 3000),
              (3006, 5021, 7, 4004, 10499, '2022-01-09', 1000),
              (3004, 5030, 10, 4002, 10444, '2022-02-27', 2500),
              (3005, 5040, 8, 4001, 10411, '2022-01-01', 3000),
              (3006, 5050, 11, 4004, 10410, '2022-02-04', 1500),
              (3005, 5051, 15, 4001, 10440, '2022-02-02', 2000),
              (3005, 5052, 12, 4001, 10404, '2021-12-13', 1700)

go


--Информация в чеке--
create table Chek_infoChek(Check_code int foreign key references Chek(Check_code) not null,
                                          cod_prod int foreign key references
Product(cod_prod) not null,
                                          Purc_date date,
                                          Gen_price smallint)
go

insert into Chek_infoChek(Check_code, cod_prod, Purc_date, Gen_price)
values  (1013, 5000, '2021-12-31', 20000),
```

```
                (1014, 5052, '2021-12-31', 2500),
                (1017, 5040, '2022-01-08', 4700),
                (1020, 5021, '2022-01-09', 2000),
                (1022, 5030, '2022-02-28', 4000)

go


--Отношение для пользователей приложения--
create table Users(user_login varchar(15) unique not null,
                                user_password varchar(16) not null,
                                user_role varchar(7) not null check(user_role = 'user' or
user_role='admin' or user_role = 'manager' or user_role='seller'),
                                user_email varchar(30) not null unique)
go

insert into Users values('User', '12345_Test', 'user', 'botannicolai22@gmail.com'),
                                                ('Seller', '1111_Seller', 'seller',
'nckthegamer@gmail.com'),
                                                ('Admin', '98765_Admin', 'admin',
'botsan12@gmail.com'),
                                                ('Manager', '777_Manager', 'manager',
'tinaalekseeva389@gmail.com')

go
```

Запросы

выборка данных

```
select * from Pasport_data
go

select * from Payment
go

select * from manufacturer
go

select * from Product_type
go

select * from supplier
go

select * from Chek
go

select * from Product
go

select * from consignment
go

select * from Chek_infoChek
go

select * from Computer
go

select * from Phone
go

select * from Monitor
```

```
go

select * from Printer
go

select * from Users
go
```

сложные запросы

```
--все компьютеры частота процессоров которых > 3900--
select * from Computer
where CPU_freq > 3900
go

--все поставки в периоде с 21.12.2021 по 04.02.2022--
select *
from consignment inner join Product on consignment.cod_prod = Product.cod_prod
where Consignment_date >= '2021-12-21' and Consignment_date <= '2022-02-04'
go

--все компьютеры и ноутбуки среди товаров--
select Product_type as 'Product type', Man_cod as 'Manufacturer code', Prod_name as 'Product
name', Price, Waranty
from Product inner join Product_type on Product.Cod_type = Product_type.Cod_type
where Product_type = 'PC' or Product_type = 'Laptop'
go

--Вывести всех поставщиков, поставляющих телефоны--
select *
from supplier
where production_type like '%Phones%'
go
```

подзапросы

```
--названия и дату всех купленных телефонов--
select Prod_name as 'Phone', Purc_date as 'Purcase date'
from Chek_infoChek inner join Product on Chek_infoChek.cod_prod = Product.cod_prod
where Cod_type = (select Cod_type from Product_type
                              where Product_type = 'Phone'
                              )
go


--вывести покупателей, коротые расплатились за товар наличными--
select IDNP, C_Adres as 'Client adres', C_Surname as 'Client surname', C_Name as 'Client name',
C_FatherName as 'Client father name'
from Pasport_data
where Client_id in (select Client_id
                                  from Chek
                                  where Pay_code = (select Pay_code
                                                                  from Payment
                                                                  where Pay_type = 'cash'
                                                                  )
                                  )
go
```

```
--Вывести тип поставленной продукции за 2021-12-21 число--
select Product_type
from Product_type
where Cod_type in (select Cod_type
                                from Product
                                where cod_prod in (select cod_prod
                                                                from consignment
                                                                where Consignment_date = '2021-
12-21'
                                                                )
                                )
Go
```

запросы добавления информации

```
insert into Product_type(Cod_type, Product_type)
values (2000, 'PC'),
          (2010, 'Laptop'),
          (2020, 'Monitor'),
          (2030, 'Phone'),
          (2040, 'Tablet PC'),
          (2050, 'Printer')
go
```

```
insert into supplier(Sup_code, Sup_name, Man_cod, Sup_Adres, production_type, Telephone)
values  (3001, 'Comp_Moldova', 4003, 'str. Stefan cel mare 21/5', 'PCs, Laptops, Monitors,
Phones', '+37367354672'),
              (3002, 'Hyper Supply', 4005, 'str. Gogol 34/4', 'PCs, Monitors, Printers, Laptops',
'+37369984519'),
              (3003, 'GMSupply', 4003, 'str. Columna 5', 'PCs, Laptops', '+37378534765'),
              (3004, 'TechMD', 4002, 'str. Uzinelor 7', 'Monitors, Phones, Laptops',
'+37379945543'),
              (3005, 'XMoldova', 4001, 'str. Petru Movila 7', 'Laptops, Phones, Printers,
Monitors', '+37360606607'),
              (3006, 'HPSupply', 4004, 'str. Mihail Sadoveanu 3', 'PCs, Monitors, Printers',
'+37369984519')
Go
```

```
insert into consignment(Sup_code, cod_prod, prod_number, Man_code, consignment_number,
Consignment_date, Price)
values  (3002, 5000, 5, 4005, 10456, '2021-12-21', 17000),
              (3003, 5010, 3, 4003, 10432, '2022-01-04', 14000),
              (3001, 5020, 4, 4003, 10487, '2021-10-10', 3000),
              (3006, 5021, 7, 4004, 10499, '2022-01-09', 1000),
              (3004, 5030, 10, 4002, 10444, '2022-02-27', 2500),
              (3005, 5040, 8, 4001, 10411, '2022-01-01', 3000),
              (3006, 5050, 11, 4004, 10410, '2022-02-04', 1500),
              (3005, 5051, 15, 4001, 10440, '2022-02-02', 2000),
              (3005, 5052, 12, 4001, 10404, '2021-12-13', 1700)

go
```

запросы удаления информации

```
--manufacturer(Man_name)--
declare @CManName varchar(20);
set @CManName = 'Test Manufacturer';

delete from manufacturer where Man_name = @CManName
go



--Pasport_data(IDNP)--
declare @CIDNP char(13);
set @CIDNP = '1234567890987';

delete from Pasport_data where IDNP = @CIDNP
go



--supplier(Sup_name)
declare @CSupName varchar(20);
set @CSupName = 'Test supplier';

delete from supplier where Sup_name = @CSupName
go
```

запросы изменения информации

```
update supplier
set Sup_name = 'SamsungS_MD'
where Sup_name = 'TechMD'
go



update manufacturer
set Man_name = 'Lenovo'
where Man_name = 'HP'
go



update Printer
set Printer_type = 'laser'
where cod_prod = 5052
go
```

Представления

```
--Представления--

--Создаём представление для журнала поставок--
create view Sup_Cons_view as
select Sup_name, cod_prod, prod_number, Man_code, consignment_number, Consignment_date, Price
```

```
from supplier inner join consignment on supplier.Sup_code = consignment.Sup_code
go

create view Prod_Cons_view as
select Sup_name, Prod_name, prod_number, Man_code, consignment_number, Consignment_date,
Sup_Cons_view.Price
from Product inner join Sup_Cons_view on Product.cod_prod = Sup_Cons_view.cod_prod
go

create view consignment_view as
select Sup_name, Prod_name, prod_number, Man_name, consignment_number, Consignment_date,
Prod_Cons_view.Price
from manufacturer inner join Prod_Cons_view on manufacturer.Man_code = Prod_Cons_view.Man_code
go

select * from consignment_view
go


--Создаём представление для чека--
create view Chek_payment_view as
select Check_code, Client_id, Pay_type
from Payment inner join Chek on Payment.Pay_code = Chek.Pay_code
go

create view Chek_Pasport_data as
select Check_code, IDNP, Pay_type
from Pasport_data inner join Chek_payment_view on Pasport_data.Client_id =
Chek_payment_view.Client_id
go

create view Chek_Chek_info_payment_view as
select Chek_infoChek.Check_code, IDNP, Pay_type, cod_prod, Purc_date, Gen_price
from Chek_infoChek inner join Chek_Pasport_data on Chek_infoChek.Check_code =
Chek_Pasport_data.Check_code
go

create view Chek_view as
select Check_code, IDNP, Pay_type, Prod_name, Purc_date, Gen_price
from Product inner join Chek_Chek_info_payment_view on Product.cod_prod =
Chek_Chek_info_payment_view.cod_prod
go

select * from Chek_view
go


--Создаём представление для товаров--
create view Product_Type_view as
select cod_prod, Product_type, Man_cod, Prod_name, Price, Waranty
from Product inner join Product_type on Product.Cod_type = Product_type.Cod_type
go

create view Product_view as
select Product_type, Man_name, Prod_name, Price, Waranty
from Product_Type_view inner join manufacturer on Product_Type_view.Man_cod =
manufacturer.Man_code
go

select * from Product_view
go
```

Хранимые процедуры

```
--Процедура ИЗМЕНЕНИЯ названия продукта--
```

```
create procedure dbo.Change_product_name(@CProductName varchar(20), @CNewProductName varchar(20))
as
	update Product
	set Prod_name = @CNewProductName
	where Prod_name = @CProductName
go


declare @COldProduct varchar(20), @CNewProduct varchar(20);
set @COldProduct = 'Redmi 5 pro';
set @CNewProduct = 'Redmi note 11';


exec dbo.Change_product_name @COldProduct, @CNewProduct
go



update Product
set Prod_name = 'Redmi 5 pro'
where Prod_name = 'Redmi note 11'
go


select *
from Product
go




--создадим процедуру ДОБАВЛЕНИЯ информации о новом продукте--
create procedure dbo.Add_new_product(@ProdType varchar(20), @ManName varchar(20), @ProdName
varchar(20), @ProdCoefficient decimal(3,1), @Waranty smallint, @SupName varchar(20), @ProdAmount
smallint, @ConsDate date, @ConsCoef decimal(3, 1), @ProdPrice money) as


	if(@ProdType = (select Product_type
				from Product_type
				where Product_type = @ProdType))
	begin

		declare @ProdTypeCod smallint;
		set @ProdTypeCod = (select Cod_type
						from Product_type
						where Product_type = @ProdType)

		if(@ManName = (select Man_name
					from manufacturer
					where Man_name = @ManName))
		begin

			declare @ManCode int;
			set @ManCode = (select Man_code
						from manufacturer
						where Man_name = @ManName)


			if(@SupName in (select Sup_name
						from supplier
						where Sup_name = @SupName))
			begin

				declare @SupCode int;

				set @SupCode = (select Sup_code
							from supplier
							where Sup_name = @SupName)
```

```
                        if(@ProdTypeCod in (select Cod_type
                                            from supplier_products
                                            where Sup_code = @SupCode))
                        begin
                                declare @ProdCod int, @ConsNumb int;
                                set @ProdCod = (select max(cod_prod)
                                                from Product) + 1;

                                set @ConsNumb = (select max(consignment_number)
                                                 from consignment) + 1;


                                insert into Product
                                values (@ProdCod, @ProdTypeCod, @ProdName, @ProdCoefficient,
@Waranty);

                                insert into consignment
                                values(@SupCode, @ProdCod, @ProdAmount, @ManCode, @ConsNumb,
@ConsDate, @ConsCoef);

                                insert into manufacturer_pruduction_price
                                values (@ManCode, @ProdCod, @ProdPrice);


                        end
                        else
                        begin
                                print 'This supplier does not supply such type of
production!'
                                end
                end
                else
                begin
                        print 'Sorry, such supplier has not been found!'
                end
        end
        else
        begin
                print 'Sorry, such manufacturer has not been found!'
        end
end
else
begin
        print 'Sorry, such product type has not been found!'
end


go



declare @ProdType varchar(20), @ManName varchar(20), @ProdName varchar(20), @ProdCoefficient
decimal(3,1), @Waranty smallint, @SupName varchar(20), @ProdAmount smallint, @ConsDate date,
@ConsCoef decimal(3, 1), @ProductionPrice money;
--set @ProdType = 'Test';
set @ProdType = 'Laptop';
--set @ManName = 'Test';
set @ManName = 'Xiaomi';
set @ProdName = 'Test';
set @ProdCoefficient = 5;
set @Waranty = 5;
--set @SupName = 'Test111';
set @SupName = 'XMoldova';
set @ProdAmount = 5;
set @ConsDate = '2022-02-02';
set @ConsCoef = 6;
set @ProductionPrice = 9999;
```

```
exec dbo.Add_new_product @ProdType, @ManName, @ProdName, @ProdCoefficient, @Waranty, @SupName,
@ProdAmount, @ConsDate, @ConsCoef, @ProductionPrice
go



--Процедура УДАЛЕНИЯ принтеров по из типу (цветной или лазкрный)--
create procedure dbo.Delete_printers_by_type(@CPrinterType varchar(6)) as

delete from Product where cod_prod in (select cod_prod
                                       from Printer
                                       where Printer_type =
@CPrinterType
                                      )
go


declare @CPrinter varchar(6);
set @CPrinter = 'laser';

exec dbo.Delete_printers_by_type @CPrinter
go
```

Создание приложения

Авторизация

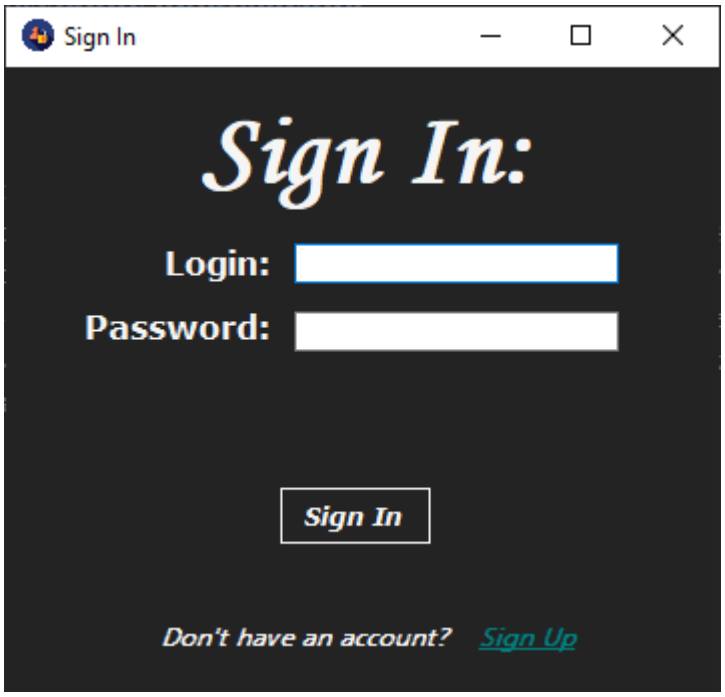
Описание:

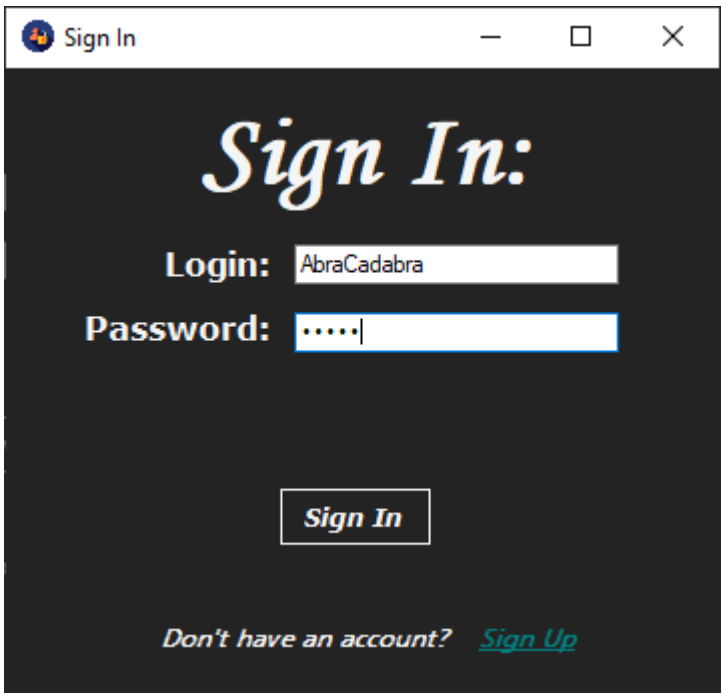
Формочка регистрации помогает контролировать распределение ролей между пользователями. Данные о пользователях хранятся в базе данных. Есть возможность войти в уже существующую учетную запись или зарегистрировать новую, приэтом ей присваивается роль обычного пользователя. При входе учитывается существование пользователя, корректность введеного пароля, а при создани нового пользователя, доступность логина, и совпадение введенного пароля и его подтверждения.

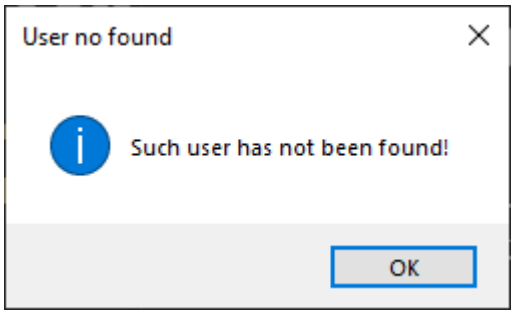
! Минимальная длинна пароля – 5 символов!

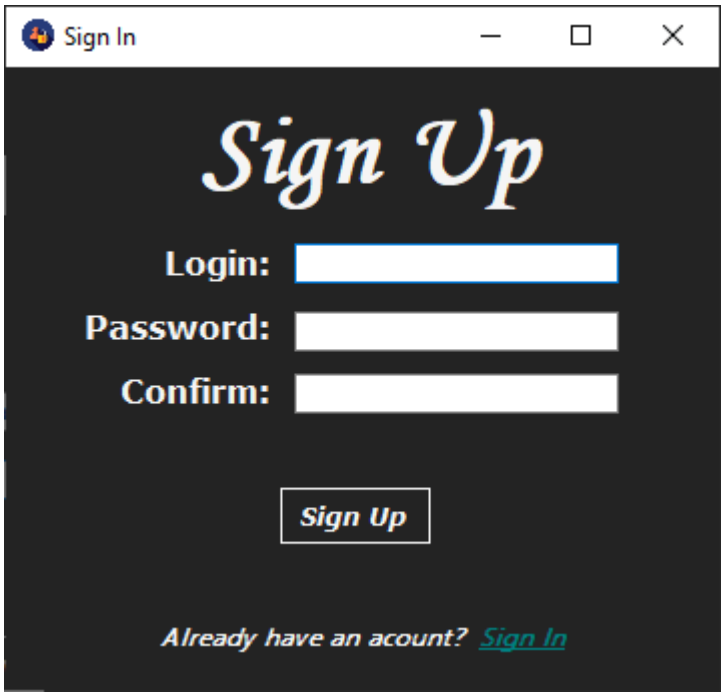
! Пароль можно восстановить!

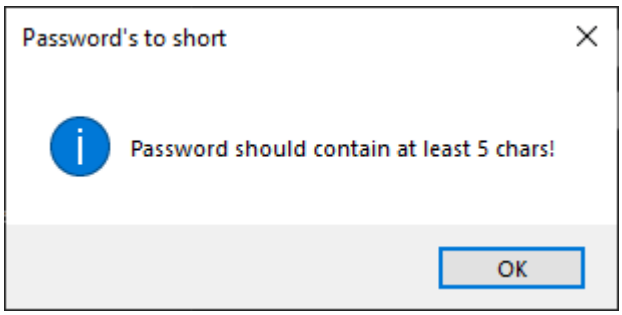
Скриншоты:

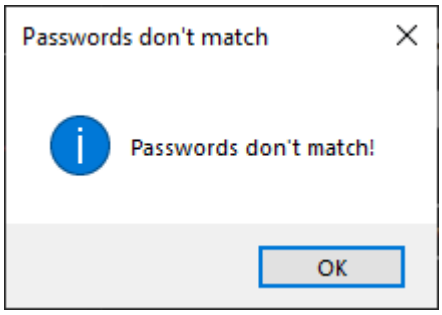


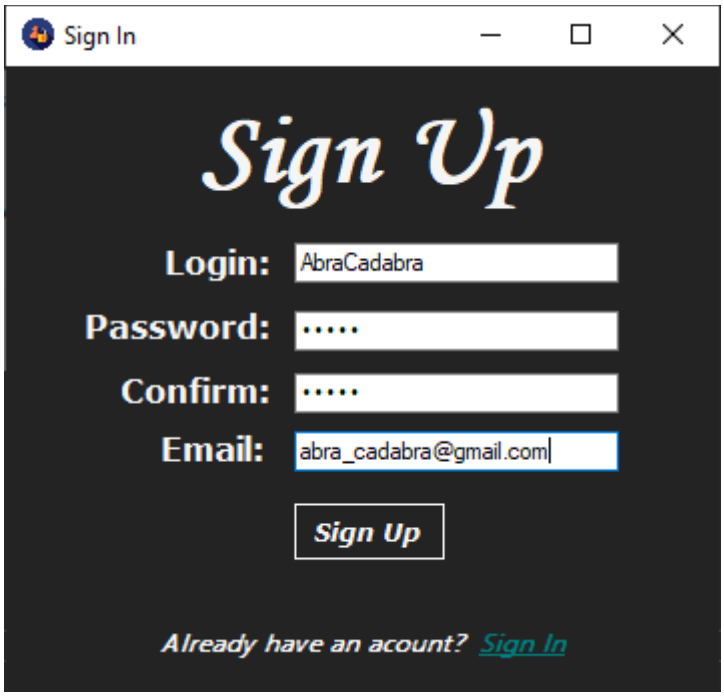

Sign In
Sign In:
Login:
AbraCadabra
Password:
Sign In
Don't have an account?
Sign Up

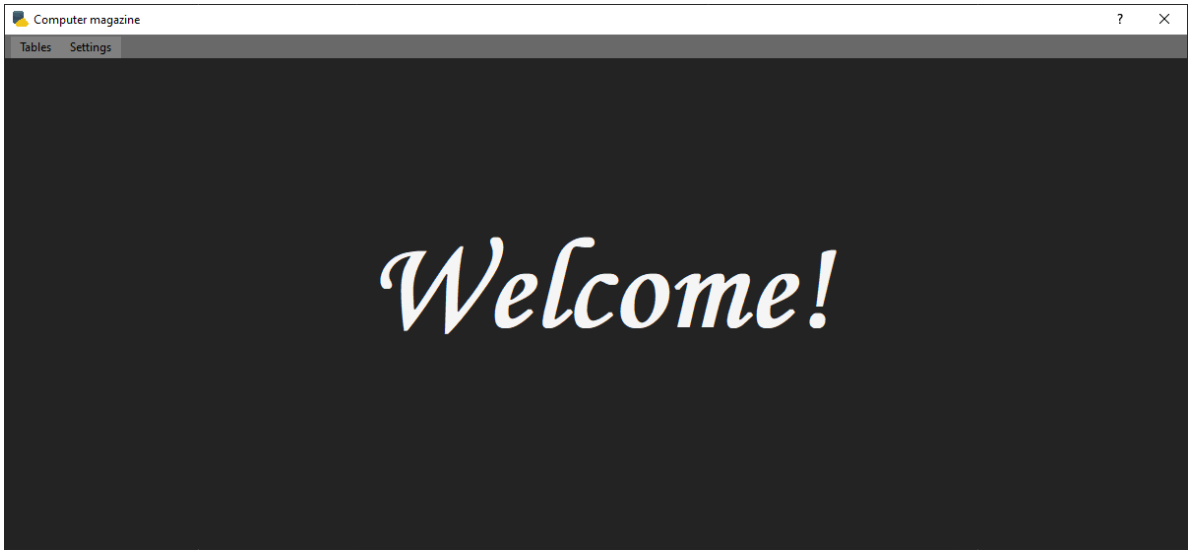

User no found
Such user has not been found!
OK


Sign In
Sign Up
Login:
Password:
Confirm:
Sign Up
Already have an acount?
Sign In

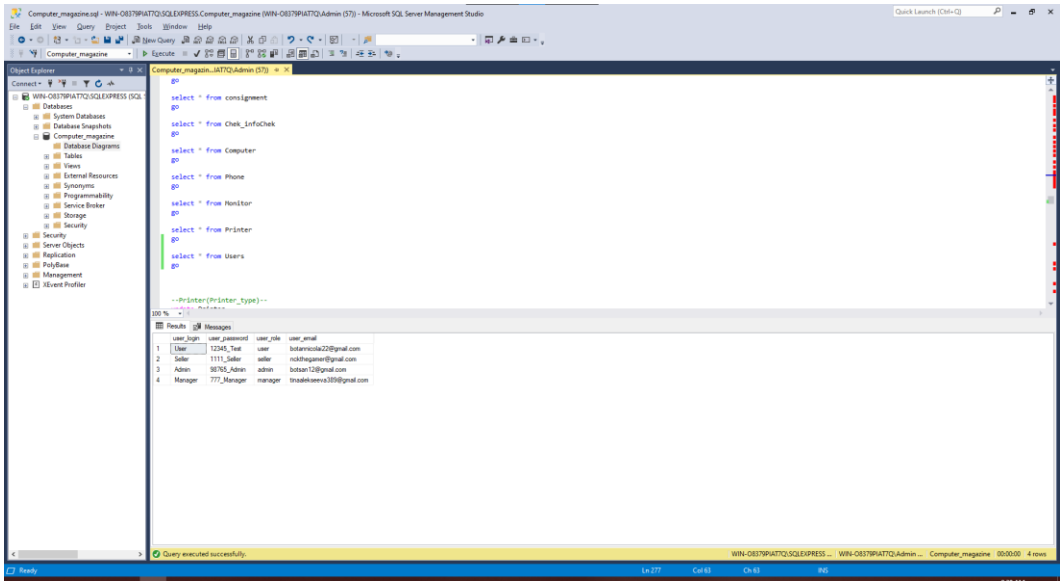

Password's to short
Password should contain at least 5 chars!
OK


Passwords don't match
Passwords don't match!
OK

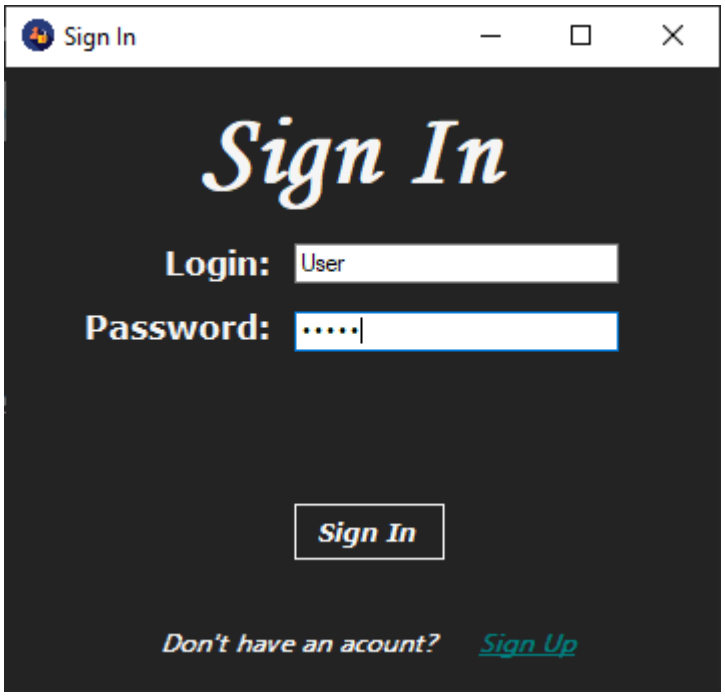

Sign In
Sign Up
Login:
AbraCadabra
Password:
Confirm:
Email:
abra_cadabra@gmail.com
Sign Up
Already have an acount? Sign In

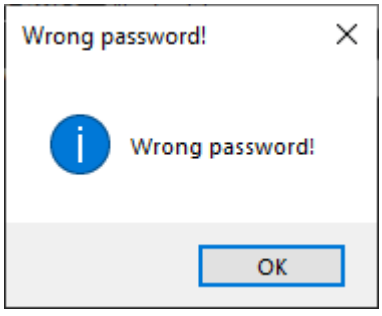

Computer magazine
Tables
Settings
Welcome!

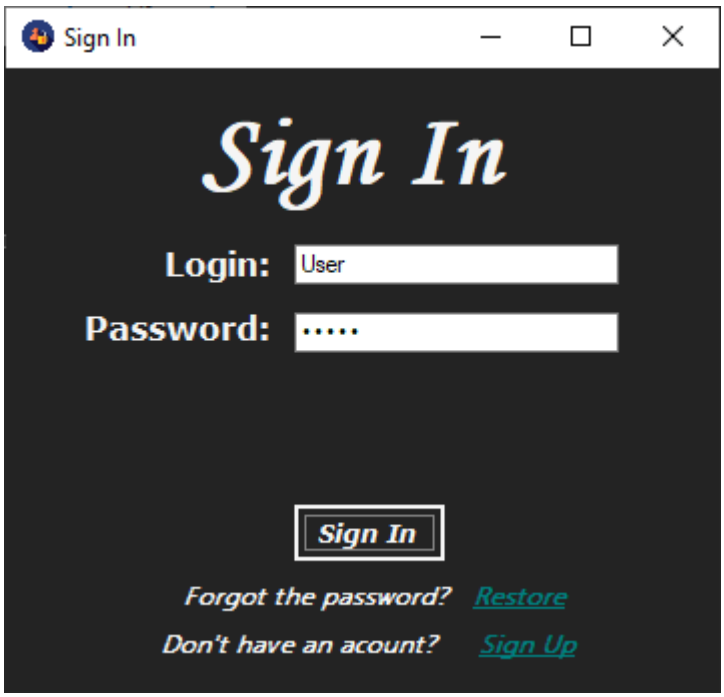
После создания пользователя он сохраняется в БД:



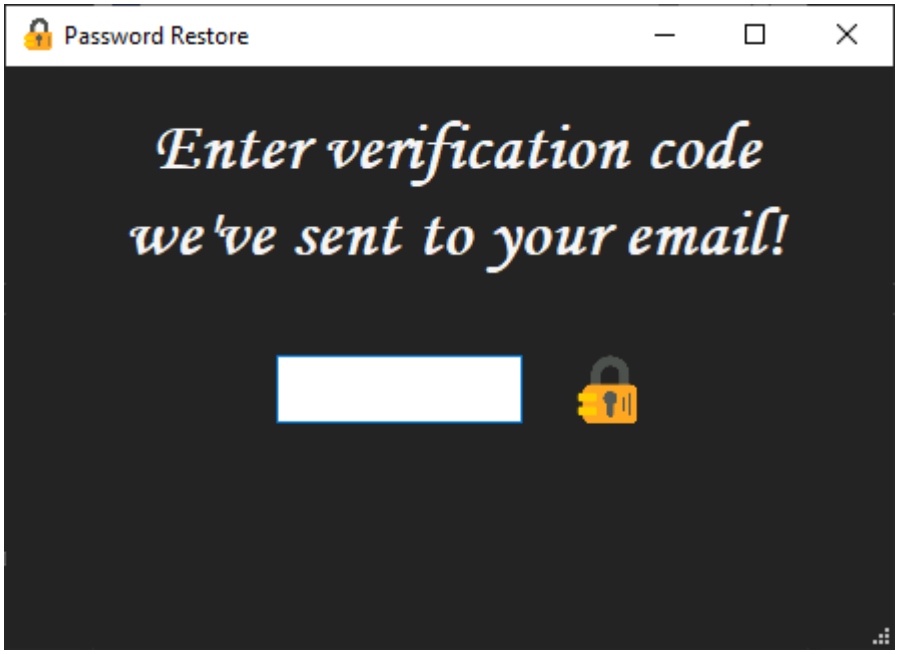
! Введем неверный пароль...



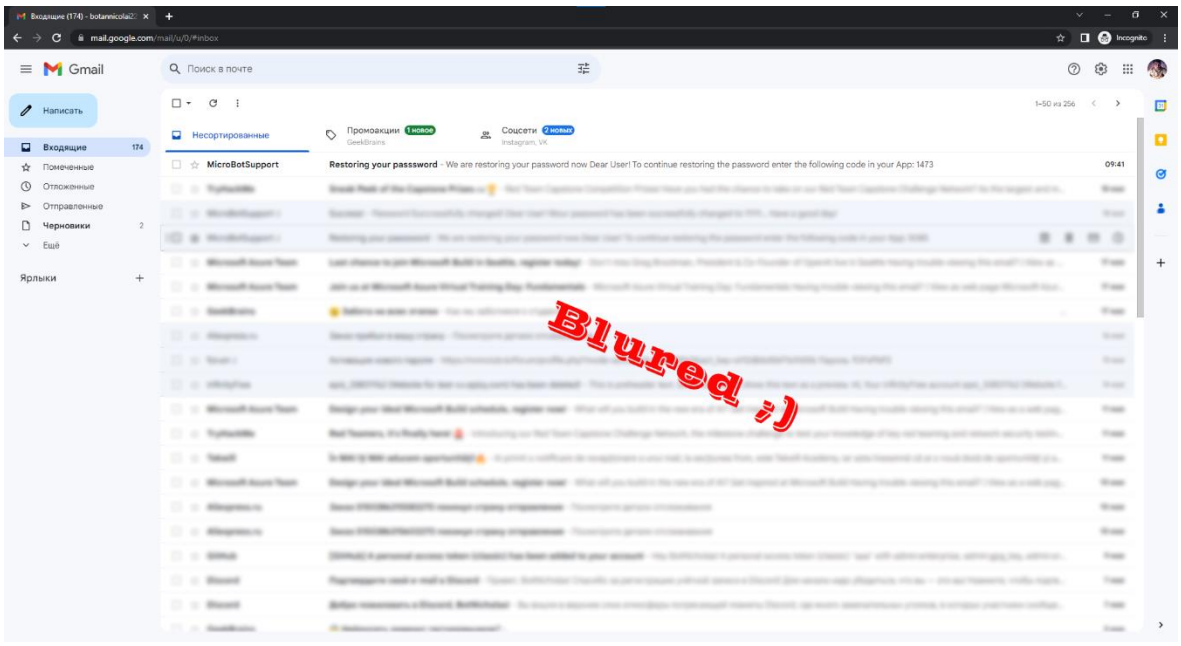


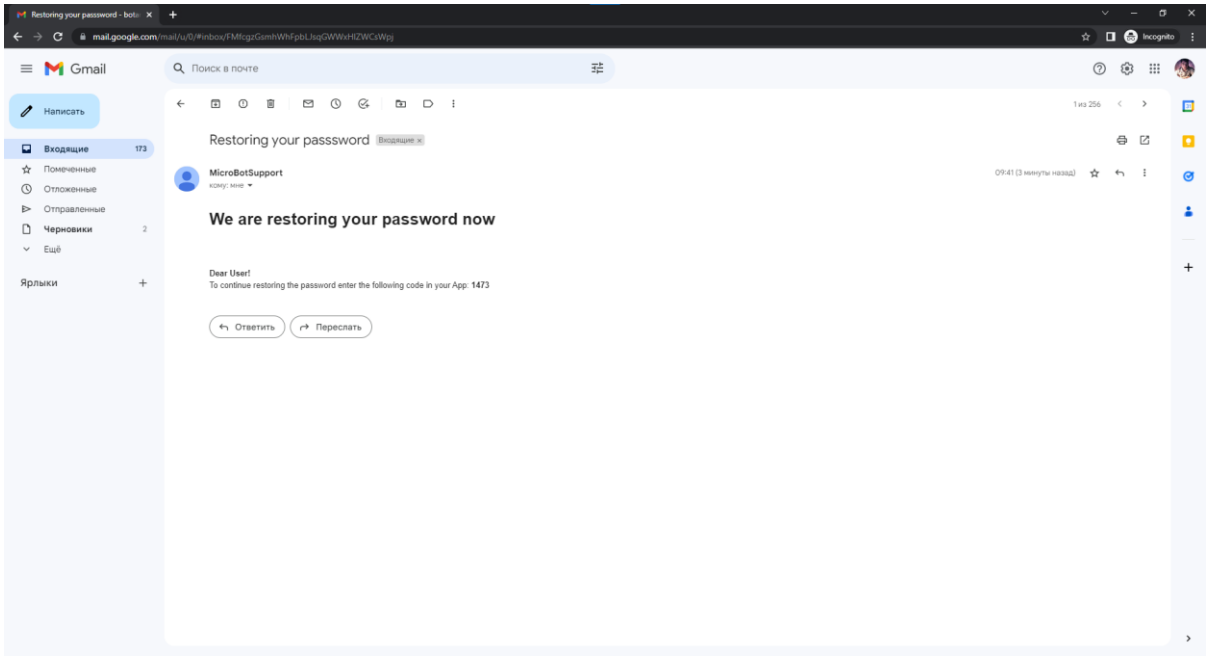


Мы можем восстановить пароль для данной учетной записи...

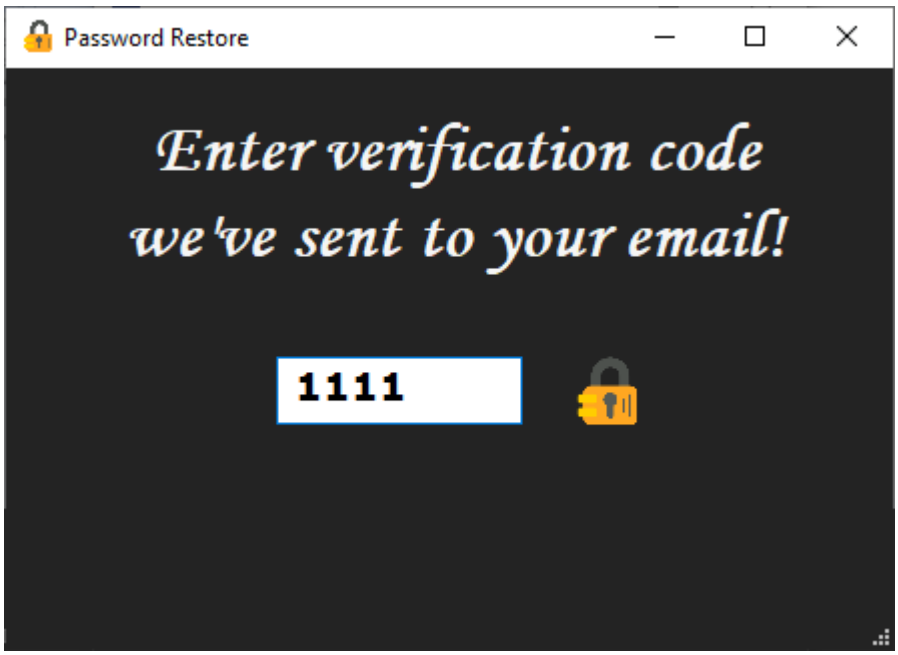


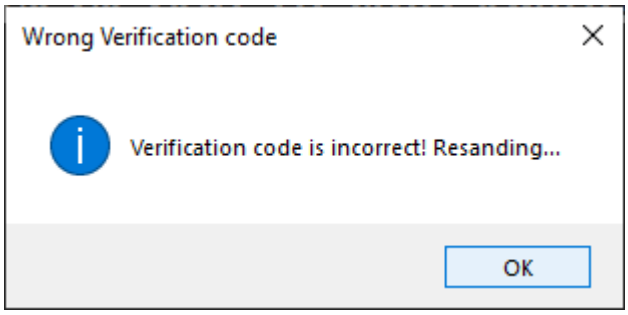
Писмо с кодомвосстановления отправляется на тот электронный адрес, который был указан при геристрации...

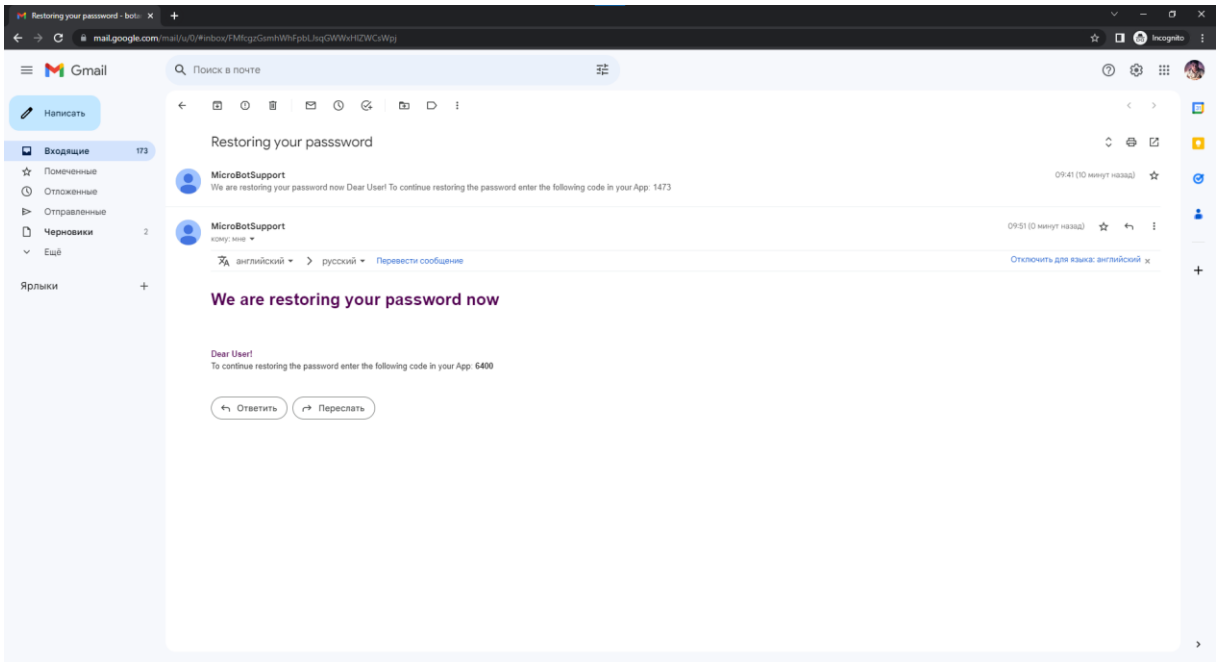


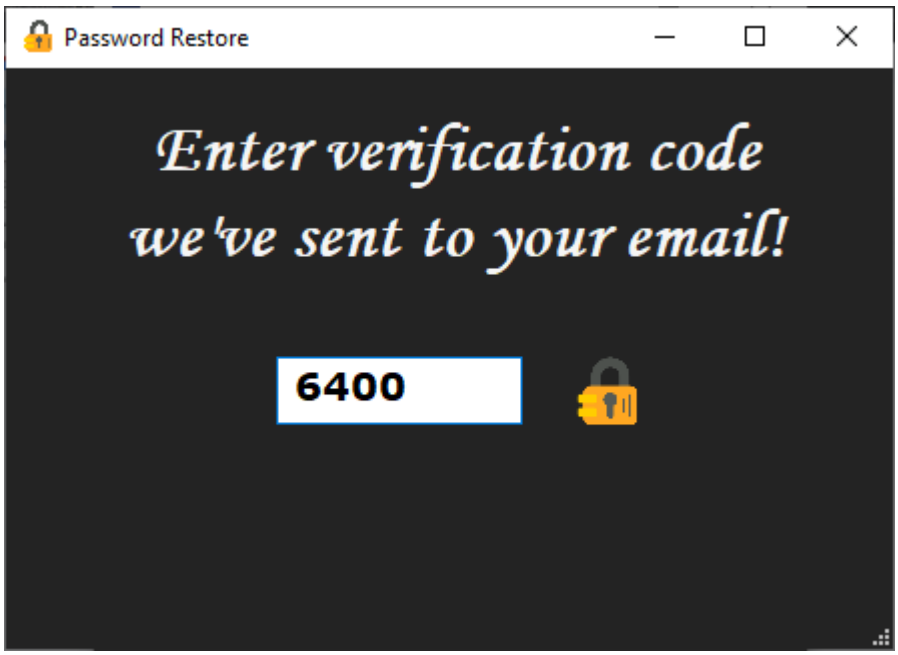


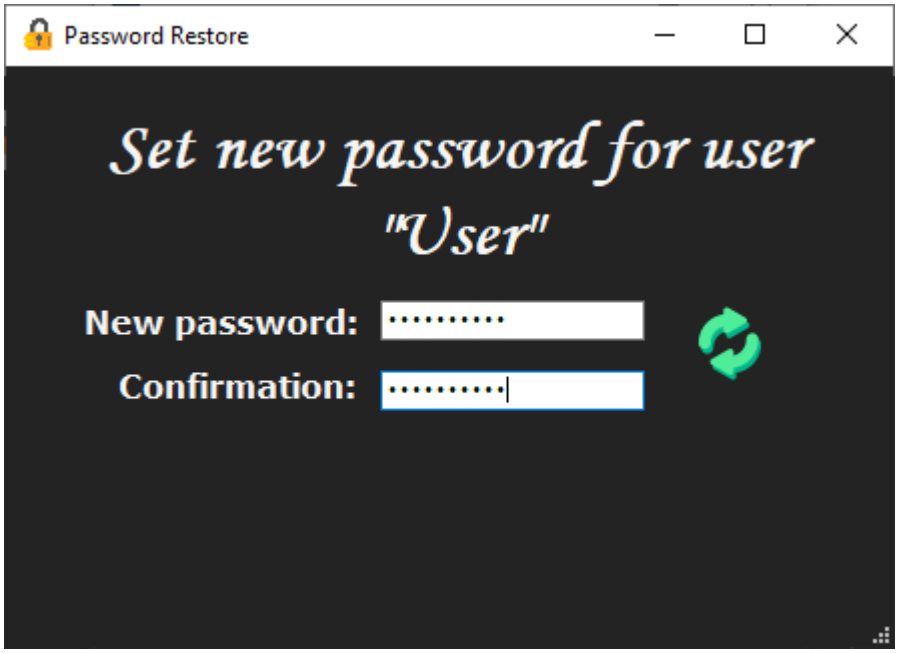
Попробуем ввести неправильный код

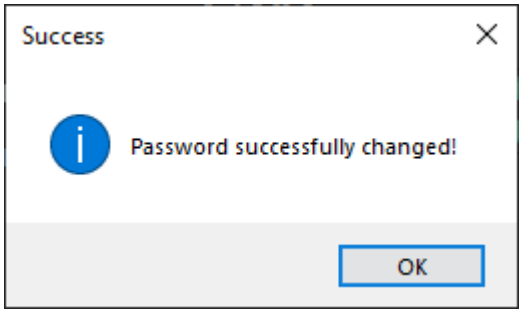

Password Restore
Enter verification code
we've sent to your email!
1111

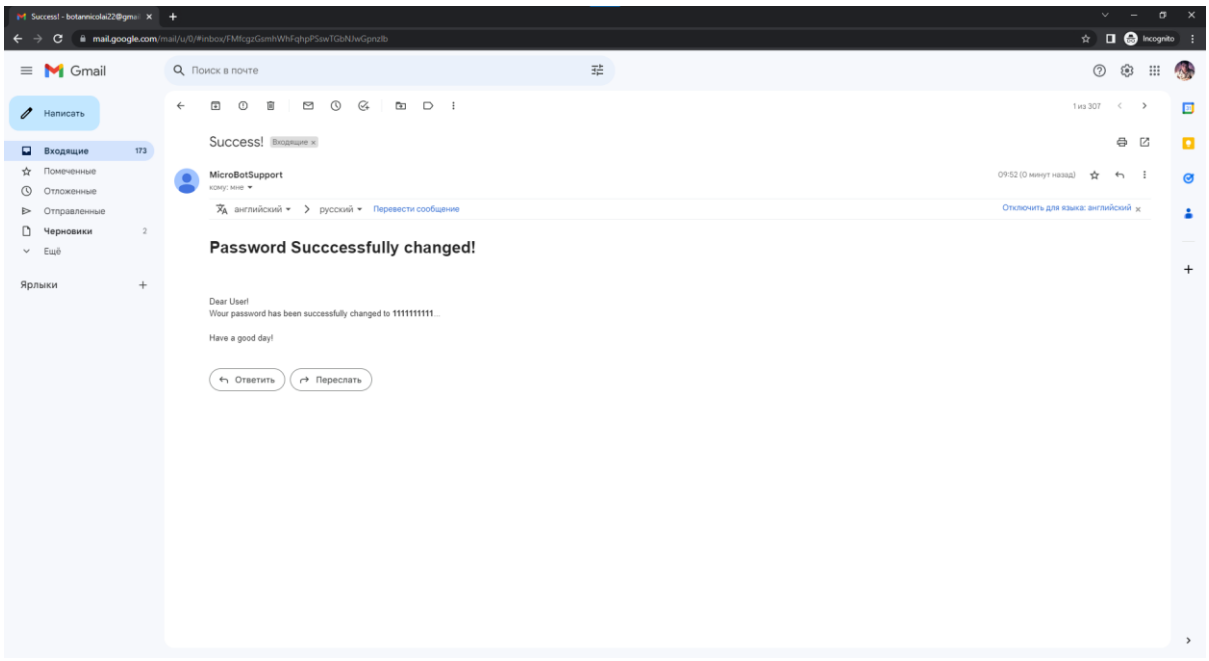

Wrong Verification code
Verification code is incorrect! Resanding...
OK


Gmail
Restoring your passsword
MicroBotSupport
We are restoring your password now Dear User! To continue restoring the password enter the following code in your App: 1473
MicroBotSupport
We are restoring your password now
Dear User!
To continue restoring the password enter the following code in your App: 6400

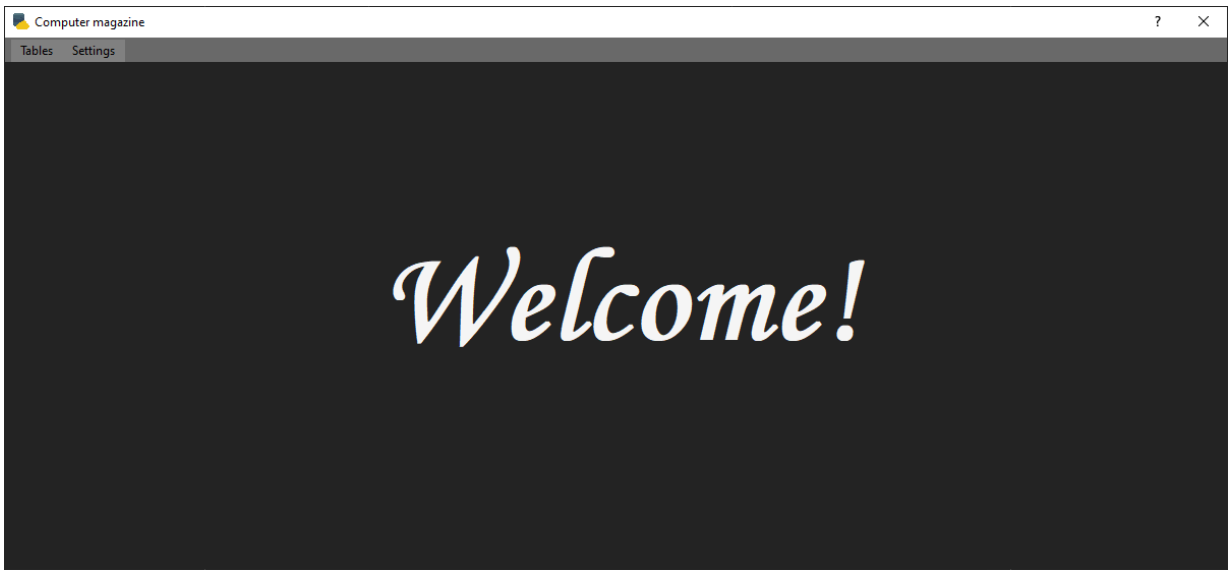

Password Restore
Enter verification code we've sent to your email!
6400


Password Restore
Set new password for user "User"
New password:
Confirmation:

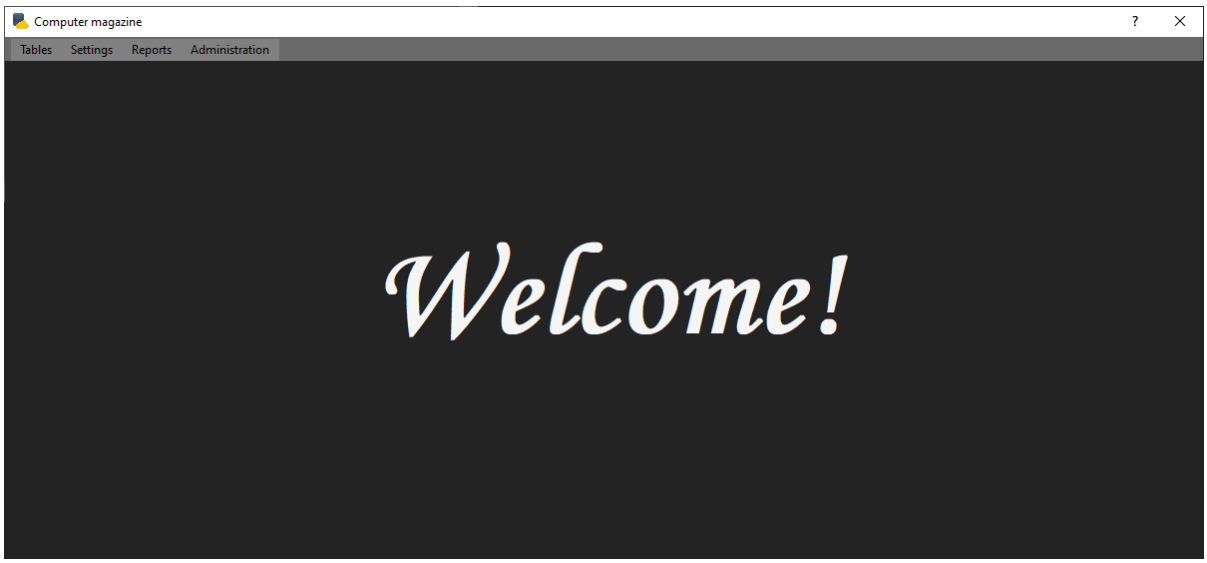

Success
Password successfully changed!
OK


Gmail
Поиск в почте
Написать
Входящие 173
Помеченные
Отложенные
Отправленные
Черновики 2
Ещё
Ярлыки
Success!
MicroBotSupport
кому: мне
09:52 (0 минут назад)
английский
русский
Перевести сообщение
Отключить для языка: английский
Password Succcessfully changed!
Dear User!
Wour password has been successfully changed to 11111111111...
Have a good day!
Ответить
Переслать


Sign In
Sign In
Login:
User
Password:
Sign In
Forgot the password?
Restore
Don't have an acount?
Sign Up



! Как вы могли заметить в БД уже есть зарегестрированные пользователи. Давайте попробуем зайти в БД под одним из них...



Фрагмент кода:

RegistrationForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class RegistrationForm : Form
    {
        Form caller;
        string mode = "signIn";
        public RegistrationForm(Form caller)
        {
            InitializeComponent();
            this.caller = caller;

            if(((Form1)caller).getStyle() == "white")
            {
                this.BackColor = Color.WhiteSmoke;

                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                label7.ForeColor = Color.Black;
                label9.ForeColor = Color.Black;
                button1.ForeColor = Color.Black;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);

                label1.ForeColor = Color.WhiteSmoke;
```

```
            label2.ForeColor = Color.WhiteSmoke;
            label3.ForeColor = Color.WhiteSmoke;
            label5.ForeColor = Color.WhiteSmoke;
            label6.ForeColor = Color.WhiteSmoke;
            label7.ForeColor = Color.WhiteSmoke;
            label9.ForeColor = Color.WhiteSmoke;
            button1.ForeColor = Color.WhiteSmoke;
        }
    }

    private void RegistrationForm_FormClosed(object sender, FormClosedEventArgs e)
    {
        caller.Close();
    }

    private void label4_Click(object sender, EventArgs e)
    {
        if (mode == "signIn")
        {
            mode = "signUp";
            label5.Text = "Sign Up";
            label3.Text = "Already have an acount?";
            label4.Text = "Sign In";
            button1.Text = "Sign Up";
            label6.Visible = true;
            textBox2.Visible = true;
            label7.Visible= true;
            textBox4.Visible = true;
            label8.Visible = false;
            label9.Visible = false;

            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";

        }
        else {
            mode = "signIn";
            label5.Text = "Sign In";
            label3.Text = "Don't have an acount?";
            label4.Text = "Sign Up";
            button1.Text = "Sign In";
            label6.Visible = false;
            textBox2.Visible = false;
            label7.Visible = false;
            textBox4.Visible = false;
            label8.Visible = false;
            label9.Visible = false;

            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            if (mode == "signIn")
            {
                using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                {
                    connection.Open();
```

```
                        SqlCommand command = new SqlCommand("select * from Users where user_login
=  @user_login collate Latin1_General_CS_AI", connection);
                        command.Parameters.Add("@user_login", textBox3.Text);

                        SqlDataReader reader = command.ExecuteReader();

                        //MessageBox.Show(reader.HasRows.ToString());

                        if (reader.HasRows)
                        {
                            reader.Read();
                            //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                            if(textBox1.Text == reader.GetValue(1).ToString())
                            {
                                //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                                ((Form1)caller).setUser(reader.GetValue(2).ToString());
                                ((Form1)caller).setUserLogin(reader.GetValue(0).ToString());
                                caller.Show();
                                this.Hide();
                            }
                            else
                            {
                                MessageBox.Show("Wrong password!", "Wrong password!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                label8.Visible = true;
                                label9.Visible = true;
                            }
                        }
                        else
                        {
                            MessageBox.Show("Such user has not been found!", "User no found",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }

                        reader.Close();
                    }
                }
                else
                {
                    if(textBox3.Text != "")
                    {
                        if (textBox1.Text.Length >= 5)
                        {
                            if (textBox1.Text == textBox2.Text)
                            {
                                if (Regex.IsMatch(textBox4.Text, "^.+\\@.+\\..+$"))
                                {
                                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                                    {
                                        connection.Open();

                                        SqlCommand command = new SqlCommand("select * from Users
where user_login = @user_login collate Latin1_General_CS_AI", connection);
                                        command.Parameters.Add("@user_login", textBox3.Text);

                                        SqlDataReader reader = command.ExecuteReader();

                                        //MessageBox.Show(reader.HasRows.ToString());

                                        if (!reader.HasRows)
                                        {
                                            reader.Close();

                                            command.CommandText = "select * from Users where
user_email = @user_email";
                                            command.Parameters.Clear();
```

```
                                        command.Parameters.Add("user_email", textBox4.Text);

                                        reader = command.ExecuteReader();

                                        if (!reader.HasRows)
                                        {
                                            reader.Close();

                                            command.CommandText = "insert into Users values
(@user_login, @user_password, 'user', @user_email)";
                                            command.Parameters.Clear();
                                            command.Parameters.Add("@user_login",
textBox3.Text);
                                            command.Parameters.Add("@user_password",
textBox1.Text);
                                            command.Parameters.Add("@user_email",
textBox4.Text);

                                            if (command.ExecuteNonQuery() > 0)
                                            {
                                                ((Form1)caller).setUser("user");
                                                ((Form1)caller).setUserLogin(textBox3.Text);
                                                this.Hide();
                                                caller.Show();
                                            }
                                            else
                                            {
                                                MessageBox.Show("An error occured!", "User
adding", MessageBoxButtons.OK, MessageBoxIcon.Error);
                                            }
                                        }
                                        else
                                        {
                                            MessageBox.Show("This email is already used!",
"Unavailable email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                        }
                                    }
                                    else
                                    {
                                        MessageBox.Show("Such user already exists!", "User
exists", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                    }

                                    reader.Close();
                                }
                            }
                            else
                            {
                                MessageBox.Show("Please enter the email in the folowing
format: example@gmail.com!", "Wrong email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                        }
                        else
                        {
                            MessageBox.Show("Passwords don't match!", "Passwords don't
match", MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    }
                    else
                    {
                        MessageBox.Show("Password should contain at least 5 chars!",
"Password's to short", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
                else
                {
                    MessageBox.Show("Enter User name!", "Empty user name",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
```

```
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void label8_Click(object sender, EventArgs e)
        {
            PasswordRestoreFrorm passwordRestoreFrorm = new PasswordRestoreFrorm(textBox3.Text,
((Form1)caller).getStyle(), this);
            passwordRestoreFrorm.ShowDialog();
        }



        public void setLogin(String login)
        {
            textBox3.Text = login;
        }

        public void setPassword(String password)
        {
            textBox1.Text = password;
        }
    }
}
```

PasswordRestoreForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class RegistrationForm : Form
    {
        Form caller;
        string mode = "signIn";
        public RegistrationForm(Form caller)
        {
            InitializeComponent();
            this.caller = caller;

            if(((Form1)caller).getStyle() == "white")
            {
                this.BackColor = Color.WhiteSmoke;

                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                label7.ForeColor = Color.Black;
                label9.ForeColor = Color.Black;
```

```
            button1.ForeColor = Color.Black;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);

            label1.ForeColor = Color.WhiteSmoke;
            label2.ForeColor = Color.WhiteSmoke;
            label3.ForeColor = Color.WhiteSmoke;
            label5.ForeColor = Color.WhiteSmoke;
            label6.ForeColor = Color.WhiteSmoke;
            label7.ForeColor = Color.WhiteSmoke;
            label9.ForeColor = Color.WhiteSmoke;
            button1.ForeColor = Color.WhiteSmoke;
        }
    }

    private void RegistrationForm_FormClosed(object sender, FormClosedEventArgs e)
    {
        caller.Close();
    }

    private void label4_Click(object sender, EventArgs e)
    {
        if (mode == "signIn")
        {
            mode = "signUp";
            label5.Text = "Sign Up";
            label3.Text = "Already have an acount?";
            label4.Text = "Sign In";
            button1.Text = "Sign Up";
            label6.Visible = true;
            textBox2.Visible = true;
            label7.Visible= true;
            textBox4.Visible = true;
            label8.Visible = false;
            label9.Visible = false;

            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";

        }
        else {
            mode = "signIn";
            label5.Text = "Sign In";
            label3.Text = "Don't have an acount?";
            label4.Text = "Sign Up";
            button1.Text = "Sign In";
            label6.Visible = false;
            textBox2.Visible = false;
            label7.Visible = false;
            textBox4.Visible = false;
            label8.Visible = false;
            label9.Visible = false;

            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            if (mode == "signIn")
```

```
                {
                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from Users where user_login
=  @user_login collate Latin1_General_CS_AI", connection);
                        command.Parameters.Add("@user_login", textBox3.Text);

                        SqlDataReader reader = command.ExecuteReader();

                        //MessageBox.Show(reader.HasRows.ToString());

                        if (reader.HasRows)
                        {
                            reader.Read();
                            //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                            if(textBox1.Text == reader.GetValue(1).ToString())
                            {
                                //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                                ((Form1)caller).setUser(reader.GetValue(2).ToString());
                                ((Form1)caller).setUserLogin(reader.GetValue(0).ToString());
                                caller.Show();
                                this.Hide();
                            }
                            else
                            {
                                MessageBox.Show("Wrong password!", "Wrong password!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                label8.Visible = true;
                                label9.Visible = true;
                            }
                        }
                        else
                        {
                            MessageBox.Show("Such user has not been found!", "User no found",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }

                        reader.Close();
                    }
                }
                else
                {
                    if(textBox3.Text != "")
                    {
                        if (textBox1.Text.Length >= 5)
                        {
                            if (textBox1.Text == textBox2.Text)
                            {
                                if (Regex.IsMatch(textBox4.Text, "^.+\\@.+\\..+$"))
                                {
                                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                                    {
                                        connection.Open();

                                        SqlCommand command = new SqlCommand("select * from Users
where user_login = @user_login collate Latin1_General_CS_AI", connection);
                                        command.Parameters.Add("@user_login", textBox3.Text);

                                        SqlDataReader reader = command.ExecuteReader();

                                        //MessageBox.Show(reader.HasRows.ToString());

                                        if (!reader.HasRows)
```

```
                                    {
                                        reader.Close();

                                        command.CommandText = "select * from Users where
user_email = @user_email";
                                        command.Parameters.Clear();
                                        command.Parameters.Add("user_email", textBox4.Text);

                                        reader = command.ExecuteReader();

                                        if (!reader.HasRows)
                                        {
                                            reader.Close();

                                            command.CommandText = "insert into Users values
(@user_login, @user_password, 'user', @user_email)";
                                            command.Parameters.Clear();
                                            command.Parameters.Add("@user_login",
textBox3.Text);
                                            command.Parameters.Add("@user_password",
textBox1.Text);
                                            command.Parameters.Add("@user_email",
textBox4.Text);

                                            if (command.ExecuteNonQuery() > 0)
                                            {
                                                ((Form1)caller).setUser("user");
                                                ((Form1)caller).setUserLogin(textBox3.Text);
                                                this.Hide();
                                                caller.Show();
                                            }
                                            else
                                            {
                                                MessageBox.Show("An error occured!", "User
adding", MessageBoxButtons.OK, MessageBoxIcon.Error);
                                            }
                                        }
                                        else
                                        {
                                            MessageBox.Show("This email is already used!",
"Unavailable email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                        }
                                    }
                                    else
                                    {
                                        MessageBox.Show("Such user already exists!", "User
exists", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                    }

                                    reader.Close();
                                }
                            }
                            else
                            {
                                MessageBox.Show("Please enter the email in the folowing
format: example@gmail.com!", "Wrong email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                        }
                        else
                        {
                            MessageBox.Show("Passwords don't match!", "Passwords don't
match", MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    }
                    else
                    {
                        MessageBox.Show("Password should contain at least 5 chars!",
"Password's to short", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
```

```
                    }
                    else
                    {
                        MessageBox.Show("Enter User name!", "Empty user name",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }

                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void label8_Click(object sender, EventArgs e)
        {
            PasswordRestoreFrorm passwordRestoreFrorm = new PasswordRestoreFrorm(textBox3.Text,
((Form1)caller).getStyle(), this);
            passwordRestoreFrorm.ShowDialog();
        }



        public void setLogin(String login)
        {
            textBox3.Text = login;
        }

        public void setPassword(String password)
        {
            textBox1.Text = password;
        }
    }
}
```

Описание интерфейса приложения

Описание какие операции (действия) в каждом пункте меню

В зависимости от зарегистрировавшегося пользователя некоровые возможности могут быть недоступны (скрыты). Так, в моём проекте предусмотрены следующие роли и возможности:

Обычный пользователь (наименьшие права) только просмотр таблиц и настройка внешнего вида (темы) приложения...

Продавец, который помимо просмотра таблиц и изменения внешнего вида приложения также может добавлять новые чеки, а также нового покупателя (паспортные данные).

Менеджер, которому добавились возможности редактировать информацию в чеках и информацию о пользователях, а также добавлять новые записи в жернал поставки и новые продкуты. Также у него появилась возможность создания отчетов.

Администратор (полный доступ). Он может редактировать информацию полностью во всех таблицах (добавлять, редактировать и удалять). У него есть возможность печати отчетов а также у него появилась возможность назначения ролей, редактирования логина и пороля пользователя. Может делать резервные копии БД и восстанавливать её...

Итак, меню:

Tables

Возможность просмотра, добавления и редактирования информации отношений. Всего отношений 13.

Settings

Возможность открыть настройки приложения, в которых, на данный момент, доступна лишь возможность изменения стиля приложения (темы).

! Все настройки сохраняютя в файл config.cfg в папке config проекта.

Reports

Возможность печати отчетов по проекту, например жунал поставки, чеки и товары (в товарах есть возможность просмотр графика типов товаров магазина)...

Administration

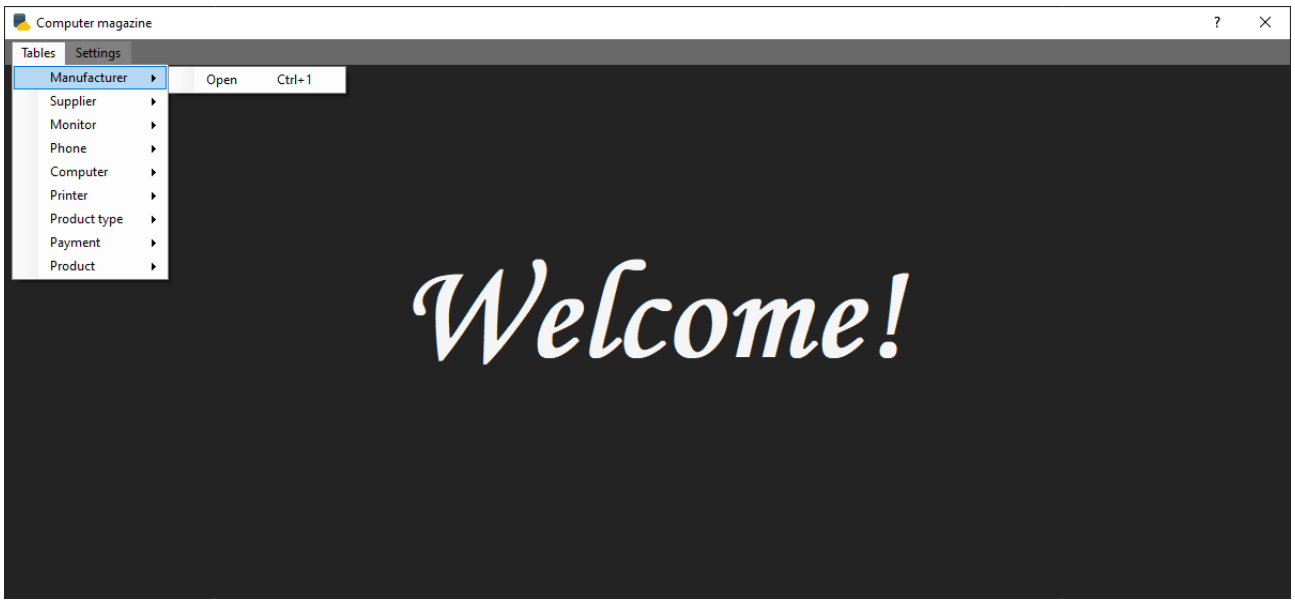
Это меню доступно только только для администраторов. Тут администратор может изменить роль пользователя (кроме своей), их логин и пароль. Также аднимистратору запрещается удалить себя, если в БД он остался один. При удалении админом самого себя, приложение закрывается.

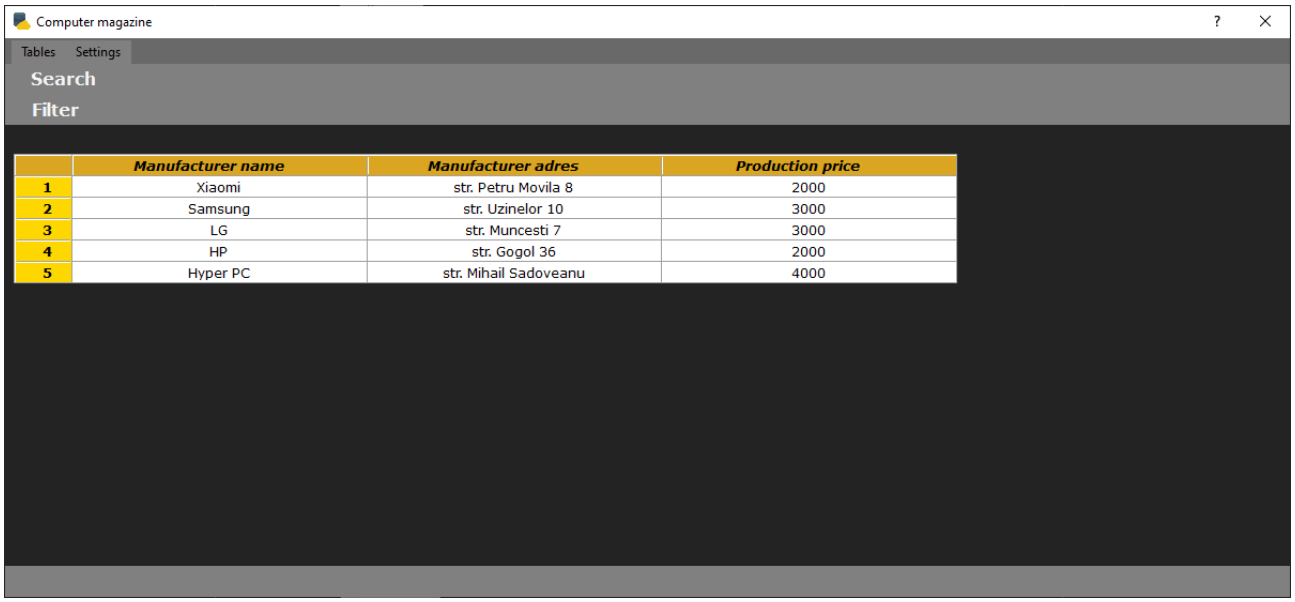
Итоги

Также для определенных отношений доступен поиск (например в *поставщике, поставках, информация в чеках, товарах, паспортных данных*), фильтрация (в *поставщике, мониторах, компьютерах, принтерах, поставках, товарах*) и итоги (сумма, максимальное и минимальное значение, и сренее значение)(в *мониторах, компьютерах, поставках, информация в чеках, товарах*)...

Скриншоты

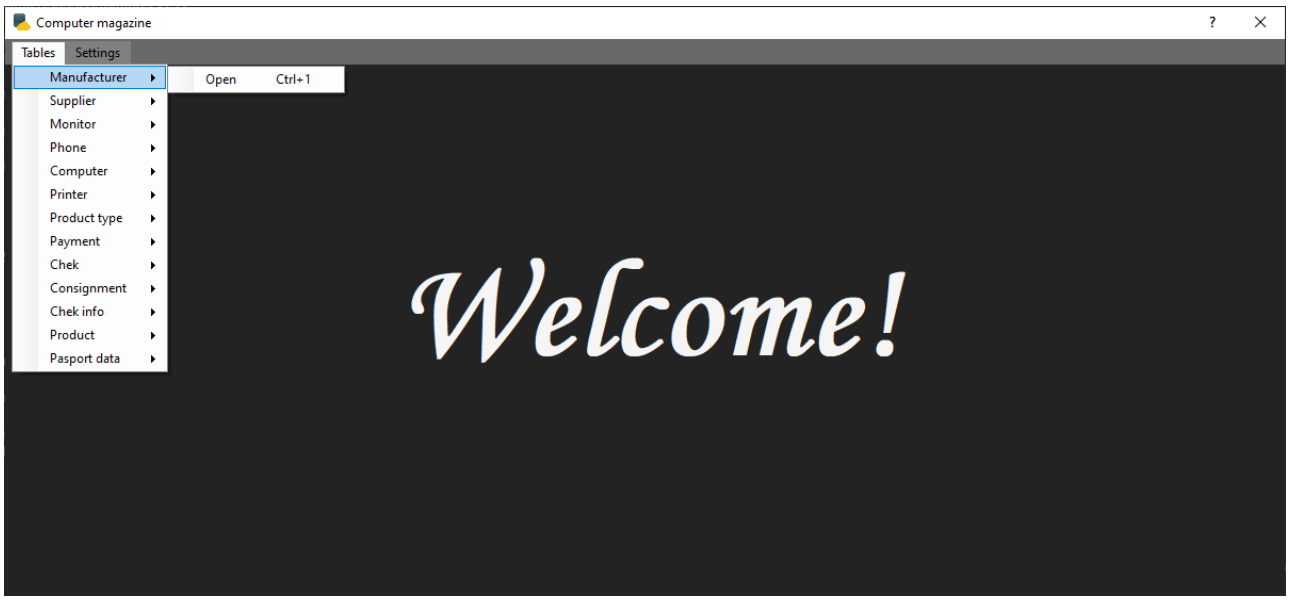
Обычный пользователь

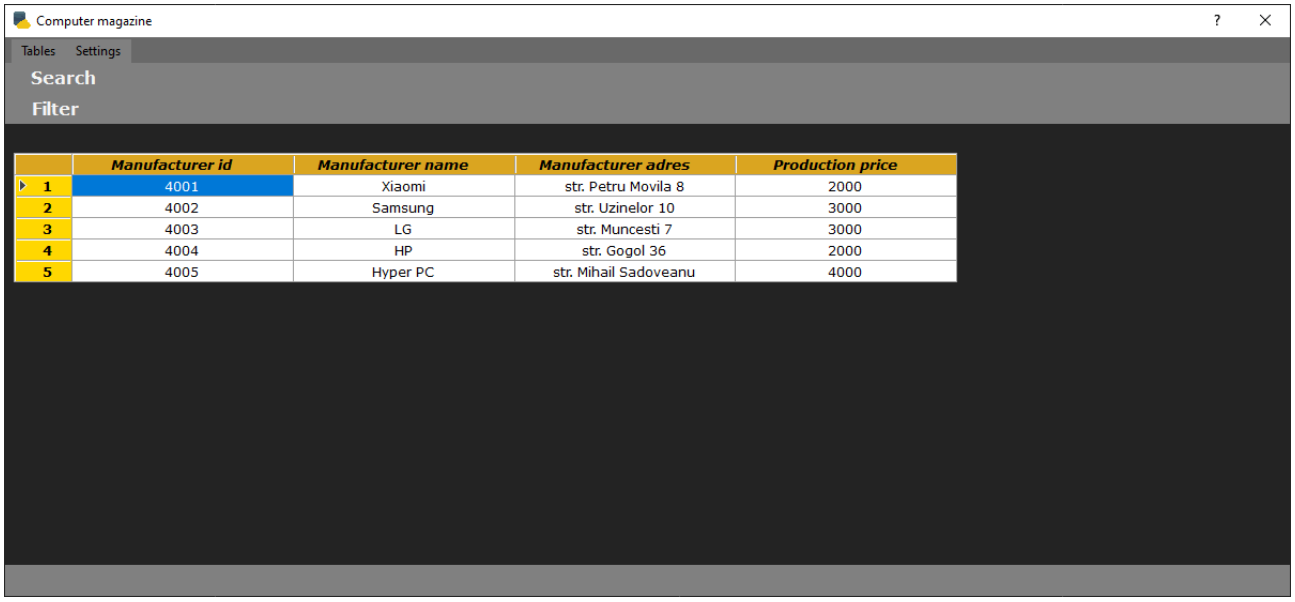


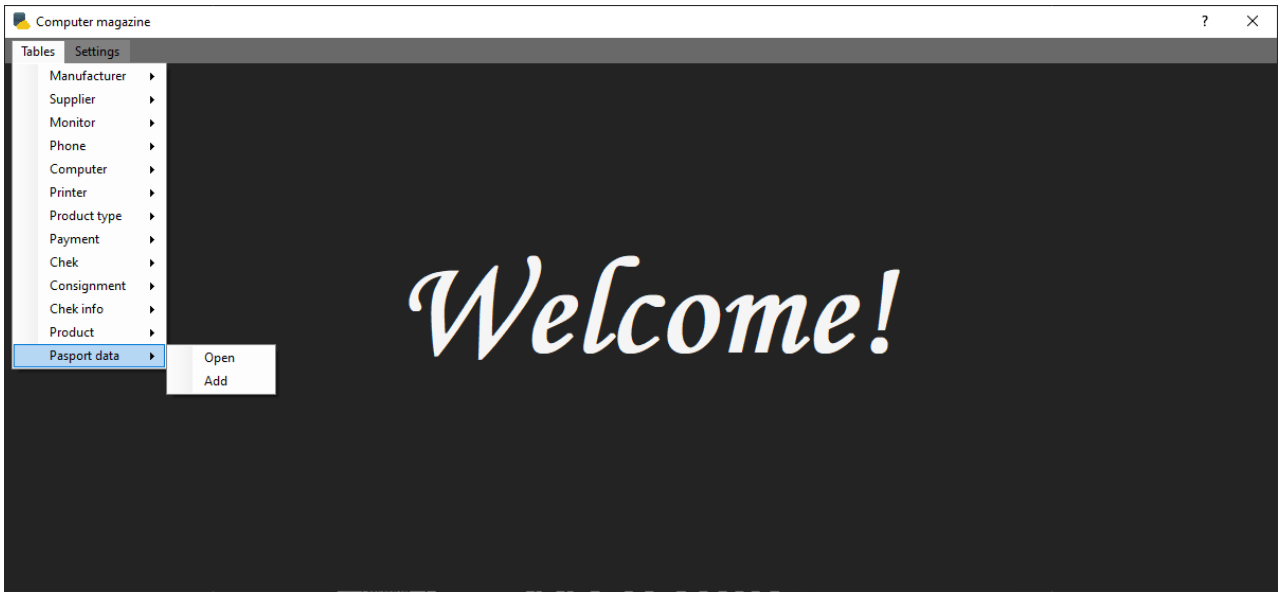


(Простому пользователю недоступны коды)

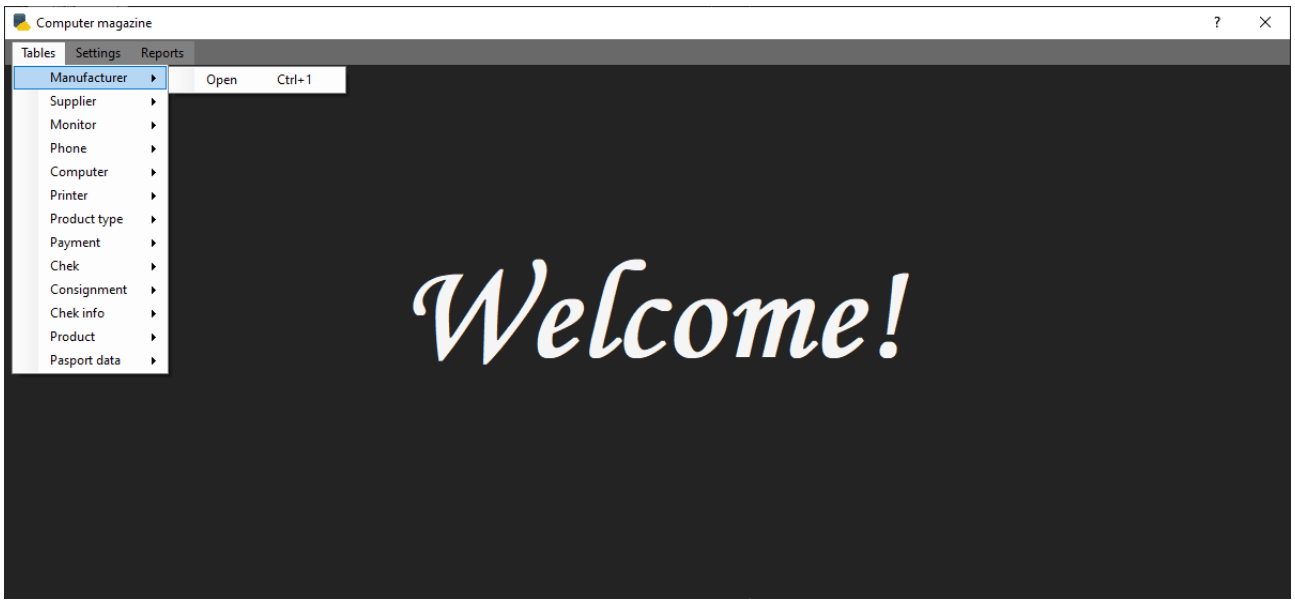
Продавец

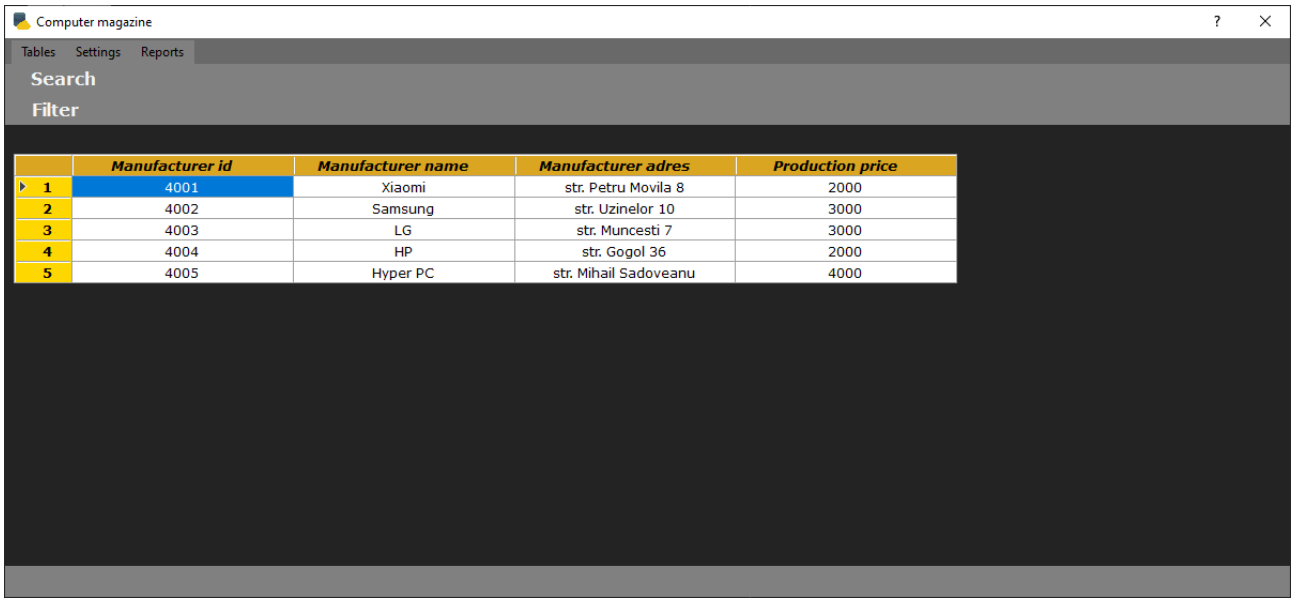

Computer magazine
Tables
Settings
Manufacturer
Supplier
Monitor
Phone
Computer
Printer
Product type
Payment
Chek
Consignment
Chek info
Product
Pasport data
Open Ctrl+1
Welcome!

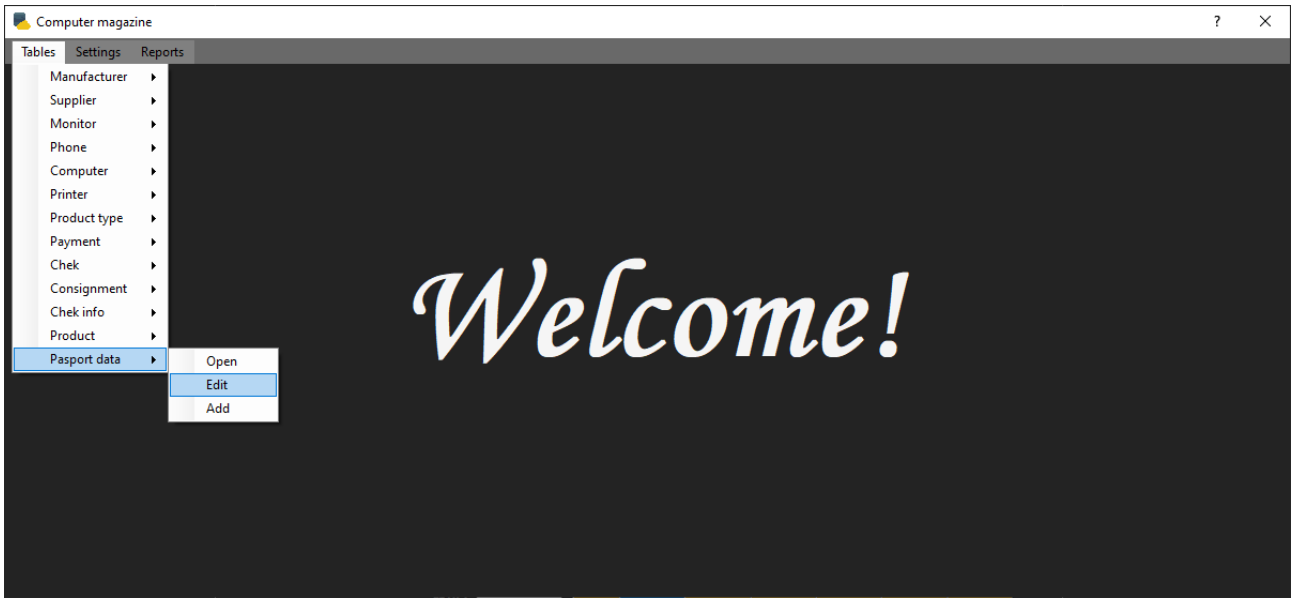

Computer magazine
Tables
Settings
Search
Filter
Manufacturer id
Manufacturer name
Manufacturer adres
Production price
1 4001 Xiaomi str. Petru Movila 8 2000
2 4002 Samsung str. Uzinelor 10 3000
3 4003 LG str. Muncesti 7 3000
4 4004 HP str. Gogol 36 2000
5 4005 Hyper PC str. Mihail Sadoveanu 4000

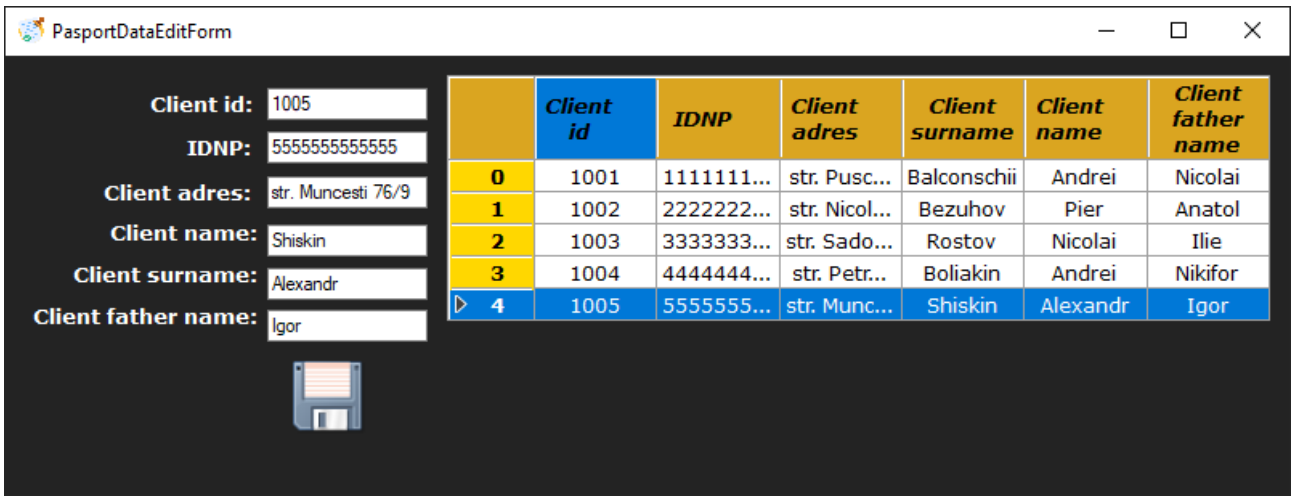

Computer magazine
Tables
Settings
Manufacturer
Supplier
Monitor
Phone
Computer
Printer
Product type
Payment
Chek
Consignment
Chek info
Product
Pasport data
Open
Add
Welcome!

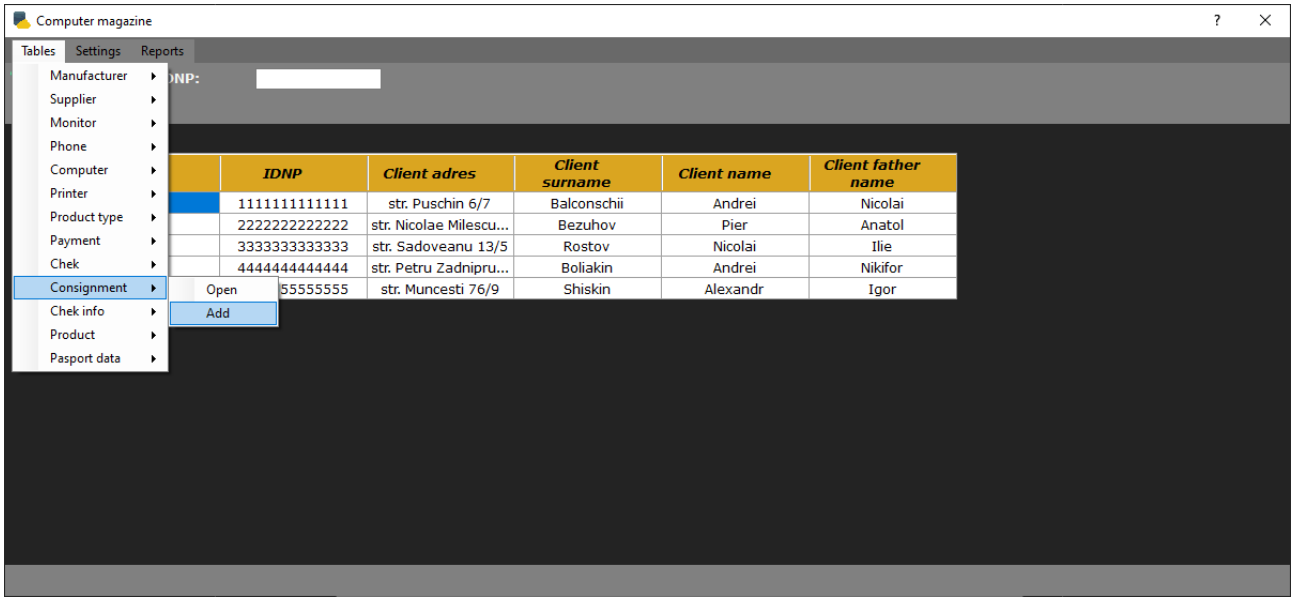
Менеджер

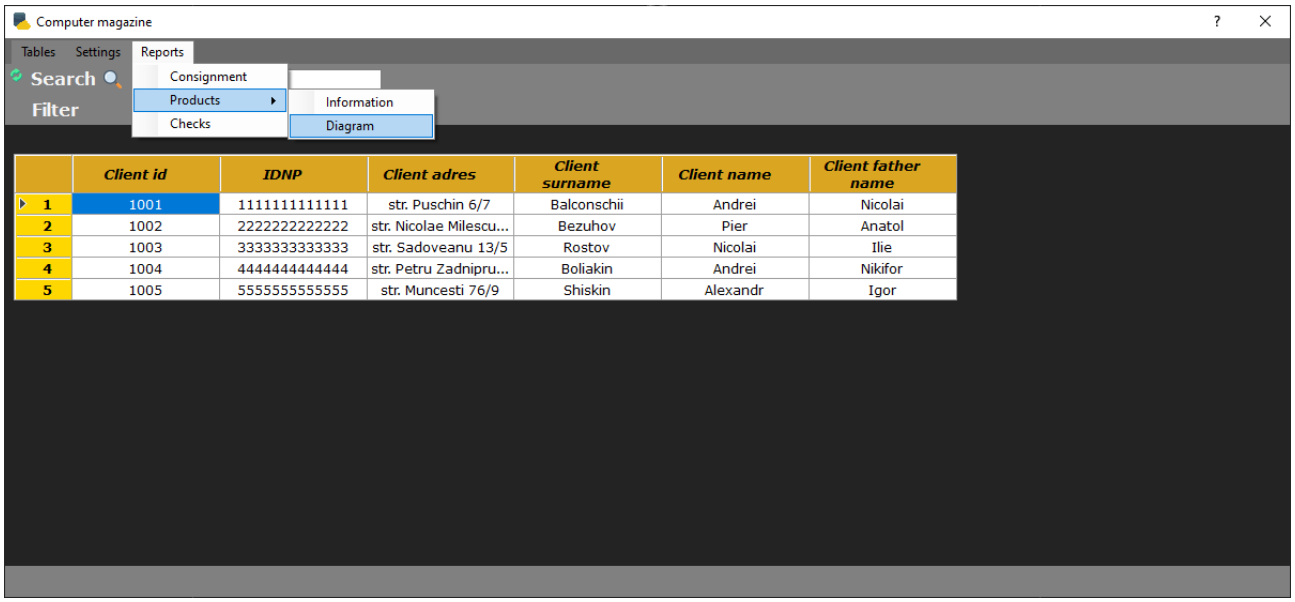

Computer magazine
Tables
Settings
Reports
Manufacturer
Supplier
Monitor
Phone
Computer
Printer
Product type
Payment
Chek
Consignment
Chek info
Product
Pasport data
Open Ctrl+1
Welcome!

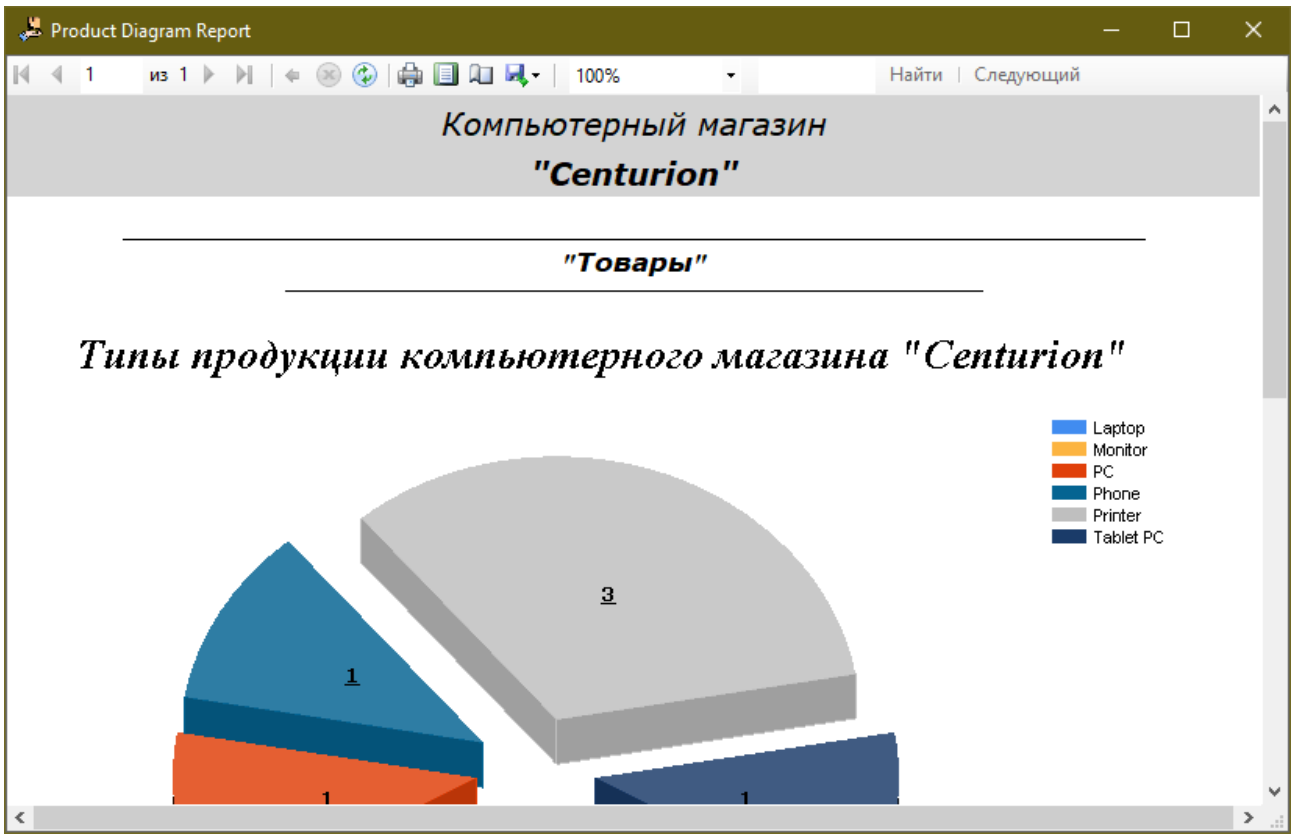

Computer magazine
Tables
Settings
Reports
Search
Filter
Manufacturer id
Manufacturer name
Manufacturer adres
Production price
1 4001 Xiaomi str. Petru Movila 8 2000
2 4002 Samsung str. Uzinelor 10 3000
3 4003 LG str. Muncesti 7 3000
4 4004 HP str. Gogol 36 2000
5 4005 Hyper PC str. Mihail Sadoveanu 4000


Computer magazine
Tables
Settings
Reports
Manufacturer
Supplier
Monitor
Phone
Computer
Printer
Product type
Payment
Chek
Consignment
Chek info
Product
Pasport data
Open
Edit
Add
Welcome!

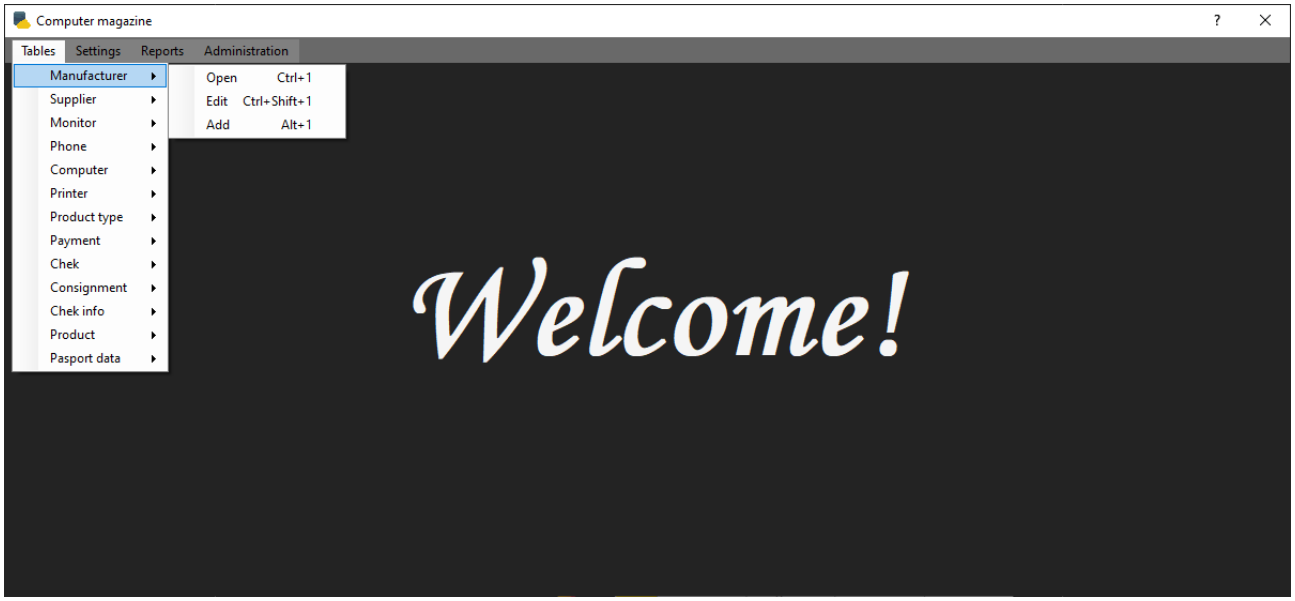

PasportDataEditForm
Client id: 1005
IDNP: 5555555555555
Client adres: str. Muncesti 76/9
Client name: Shiskin
Client surname: Alexandr
Client father name: Igor
Client id | IDNP | Client adres | Client surname | Client name | Client father name
0 | 1001 | 1111111... | str. Pusc... | Balconschii | Andrei | Nicolai
1 | 1002 | 2222222... | str. Nicol... | Bezuhov | Pier | Anatol
2 | 1003 | 3333333... | str. Sado... | Rostov | Nicolai | Ilie
3 | 1004 | 4444444... | str. Petr... | Boliakin | Andrei | Nikifor
4 | 1005 | 5555555... | str. Munc... | Shiskin | Alexandr | Igor


Computer magazine
Tables
Settings
Reports
Manufacturer
Supplier
Monitor
Phone
Computer
Printer
Product type
Payment
Chek
Consignment
Chek info
Product
Pasport data
Open
Add
IDNP | Client adres | Client surname | Client name | Client father name
1111111111111 | str. Puschin 6/7 | Balconschii | Andrei | Nicolai
2222222222222 | str. Nicolae Milescu... | Bezuhov | Pier | Anatol
3333333333333 | str. Sadoveanu 13/5 | Rostov | Nicolai | Ilie
4444444444444 | str. Petru Zadnipru... | Boliakin | Andrei | Nikifor
55555555 | str. Muncesti 76/9 | Shiskin | Alexandr | Igor


Computer magazine
Tables
Settings
Reports
Search
Filter
Consignment
Products
Checks
Information
Diagram
Client id | IDNP | Client adres | Client surname | Client name | Client father name
1 | 1001 | 1111111111111 | str. Puschin 6/7 | Balconschii | Andrei | Nicolai
2 | 1002 | 2222222222222 | str. Nicolae Milescu... | Bezuhov | Pier | Anatol
3 | 1003 | 3333333333333 | str. Sadoveanu 13/5 | Rostov | Nicolai | Ilie
4 | 1004 | 4444444444444 | str. Petru Zadnipru... | Boliakin | Andrei | Nikifor
5 | 1005 | 5555555555555 | str. Muncesti 76/9 | Shiskin | Alexandr | Igor

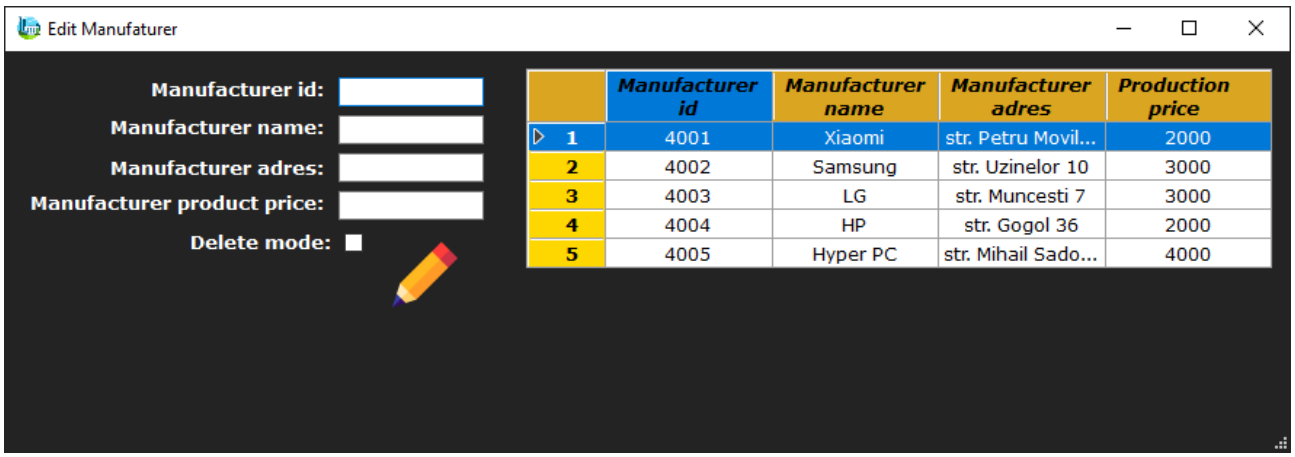


Администратор

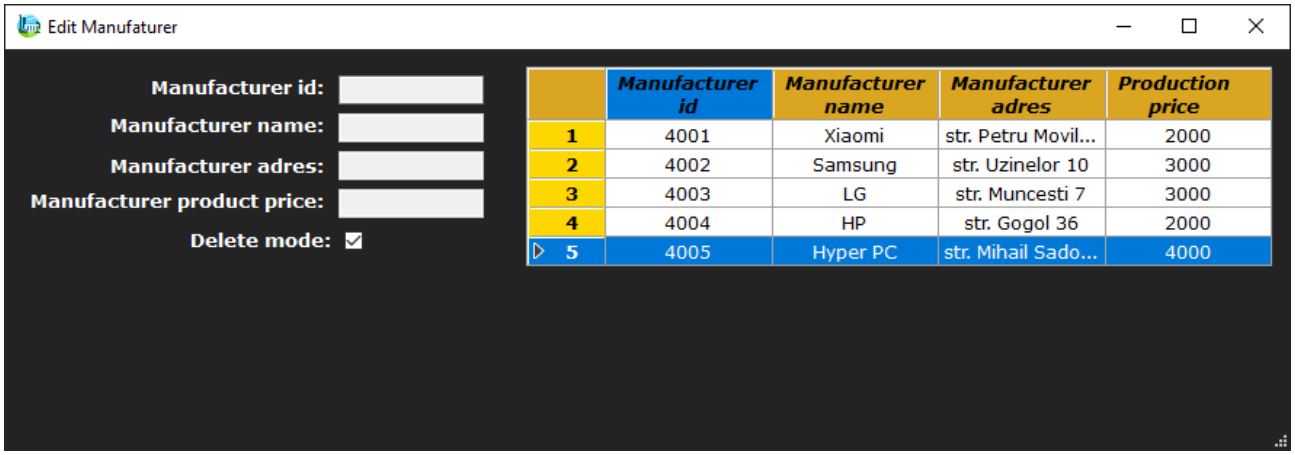


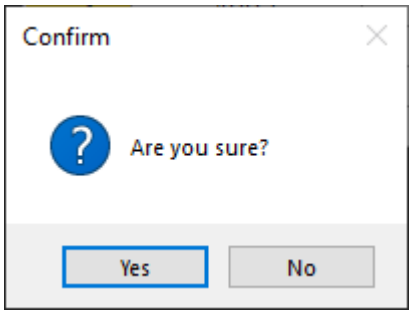
(полный доступ)

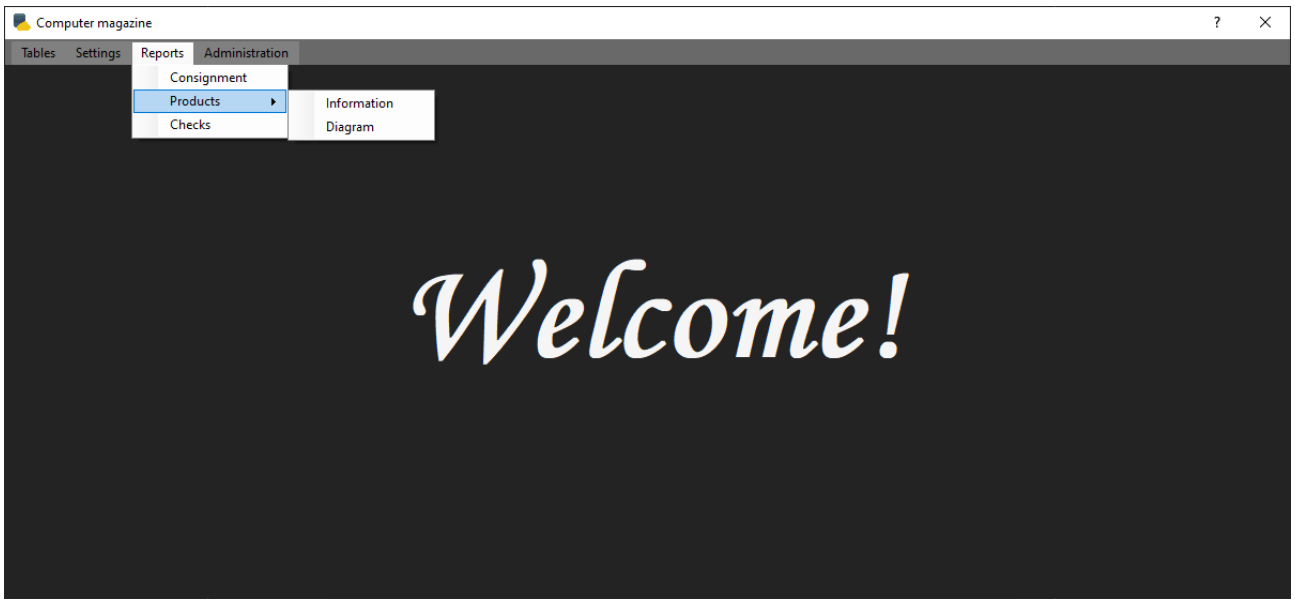
Редактирование

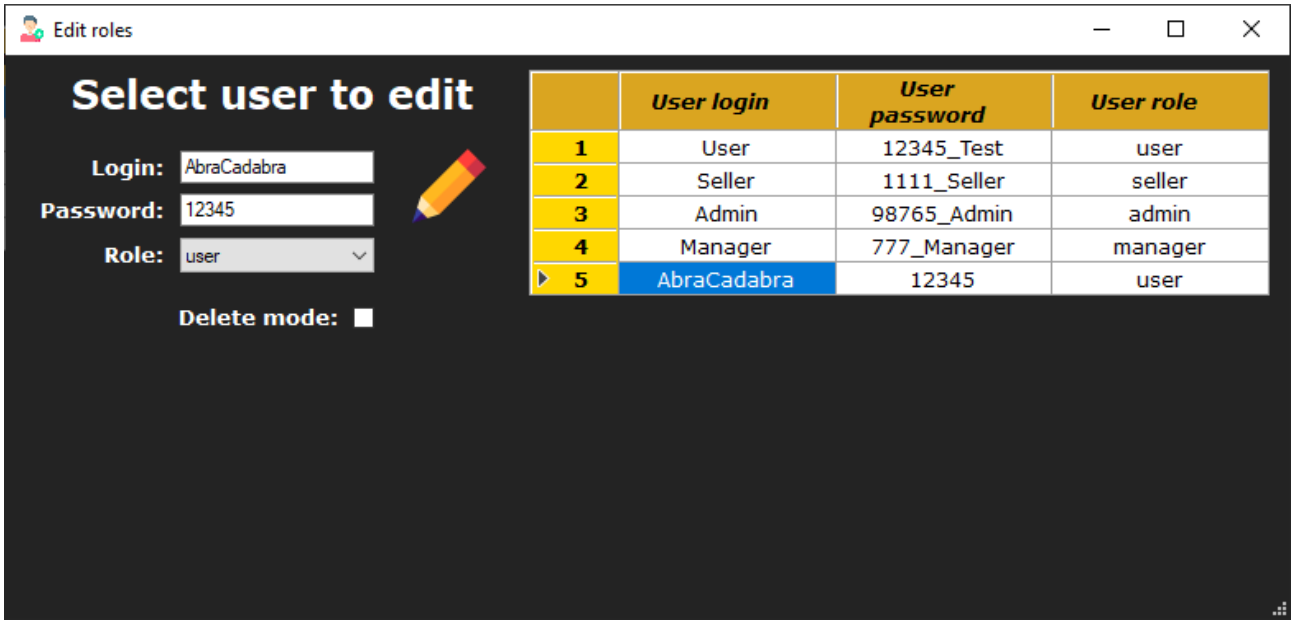

Edit Manufaturer
Manufacturer id:
Manufacturer name:
Manufacturer adres:
Manufacturer product price:
Delete mode:
Manufacturer id
Manufacturer name
Manufacturer adres
Production price
1 4001 Xiaomi str. Petru Movil... 2000
2 4002 Samsung str. Uzinelor 10 3000
3 4003 LG str. Muncesti 7 3000
4 4004 HP str. Gogol 36 2000
5 4005 Hyper PC str. Mihail Sado... 4000

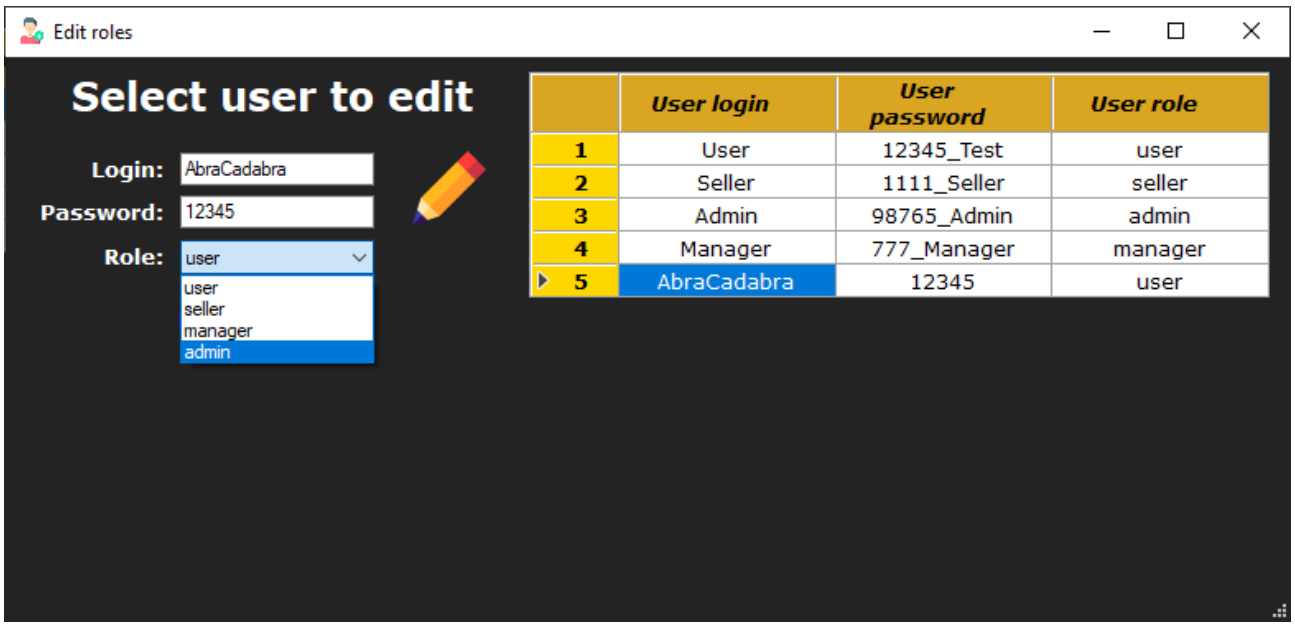
Удаление


Edit Manufaturer
Manufacturer id:
Manufacturer name:
Manufacturer adres:
Manufacturer product price:
Delete mode:
Manufacturer id
Manufacturer name
Manufacturer adres
Production price
1 4001 Xiaomi str. Petru Movil... 2000
2 4002 Samsung str. Uzinelor 10 3000
3 4003 LG str. Muncesti 7 3000
4 4004 HP str. Gogol 36 2000
5 4005 Hyper PC str. Mihail Sado... 4000

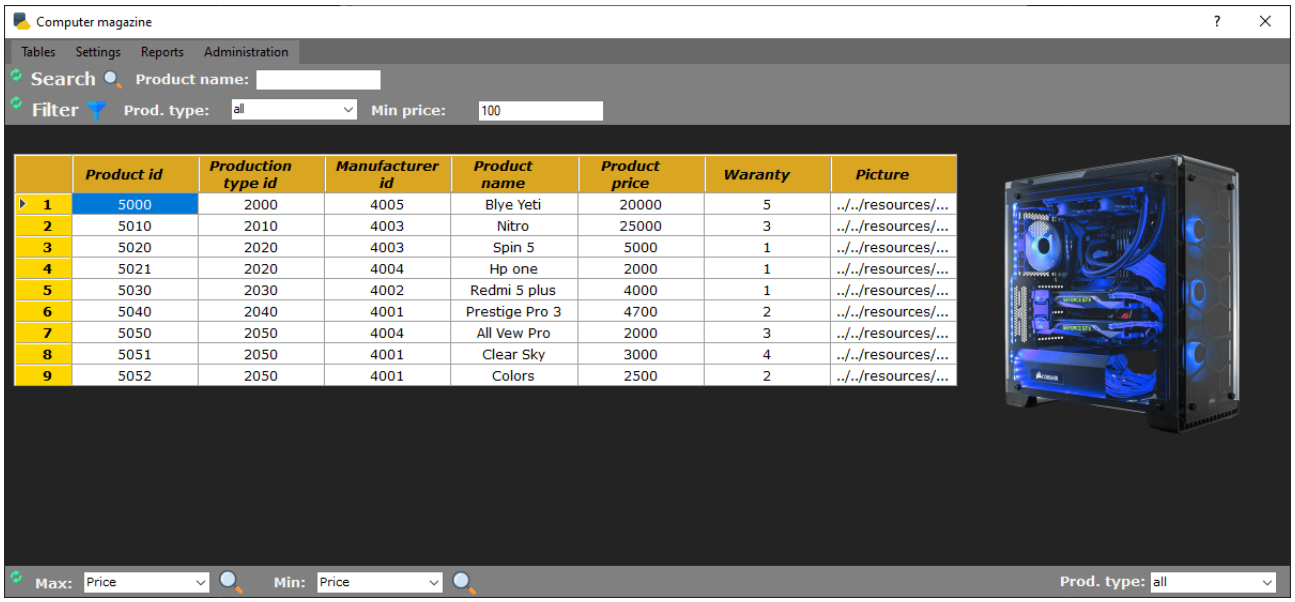

Confirm
Are you sure?
Yes
No

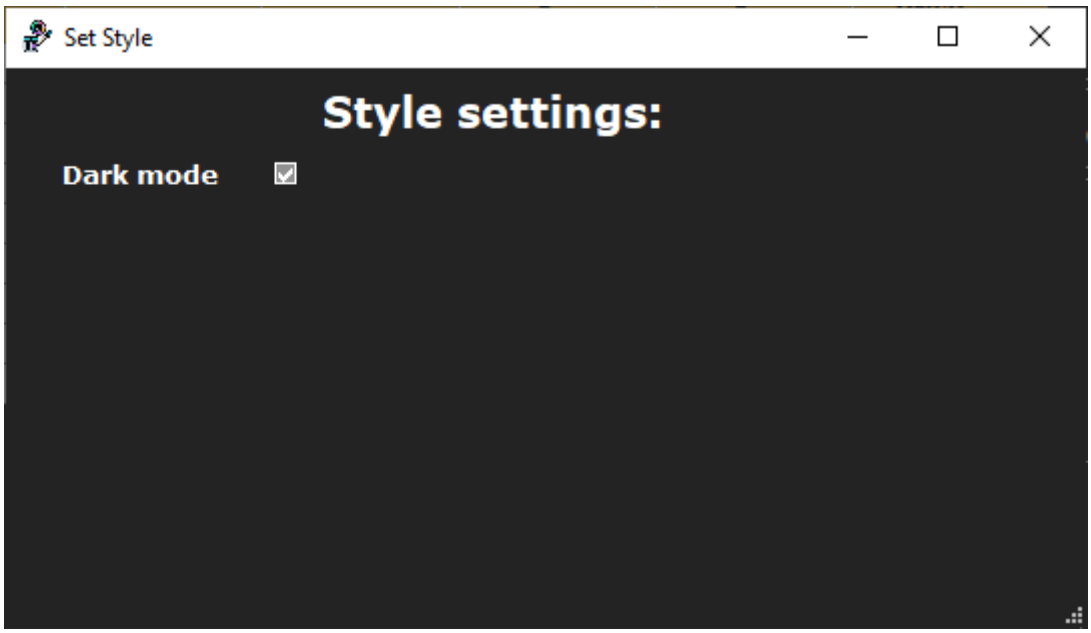

Computer magazine
Tables
Settings
Reports
Administration
Consignment
Products
Checks
Information
Diagram
Welcome!

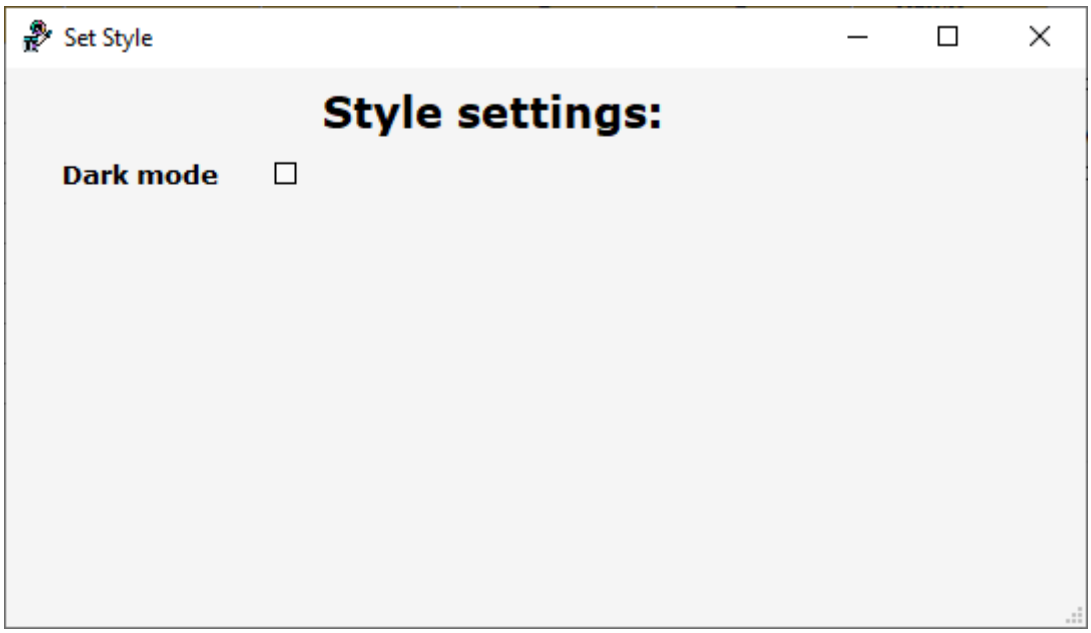

Edit roles
Select user to edit
Login:
AbraCadabra
Password:
12345
Role:
user
Delete mode:
User login
User password
User role
1
User
12345_Test
user
2
Seller
1111_Seller
seller
3
Admin
98765_Admin
admin
4
Manager
777_Manager
manager
5
AbraCadabra
12345
user

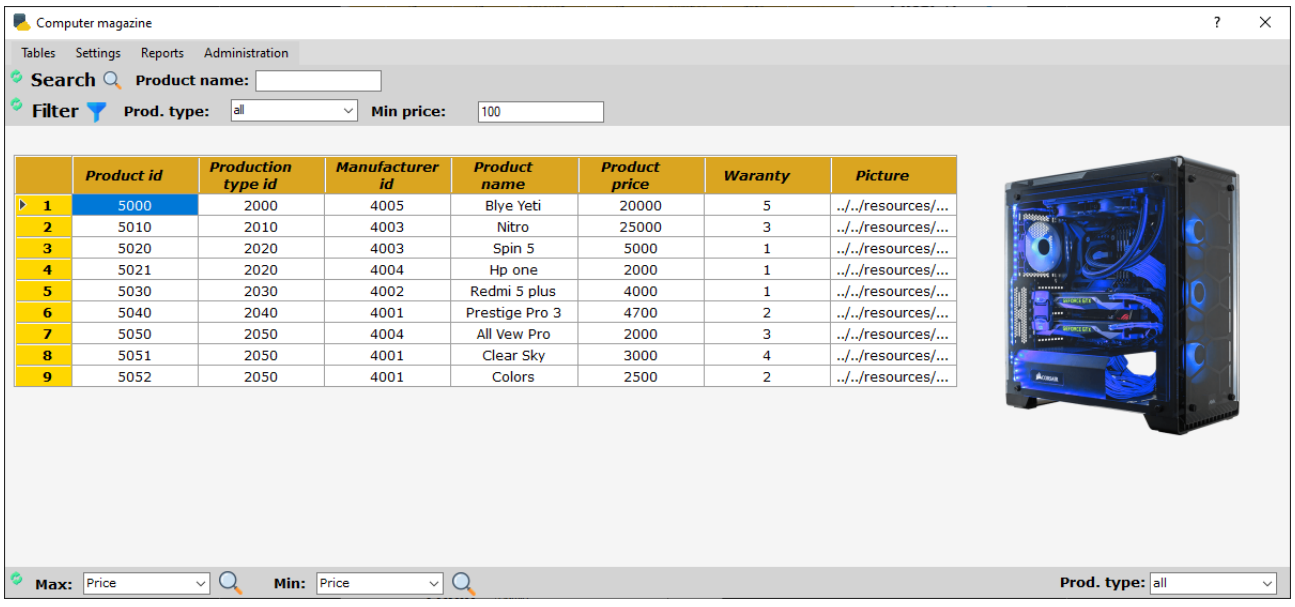


Изменение темы после перезапуска приложения:



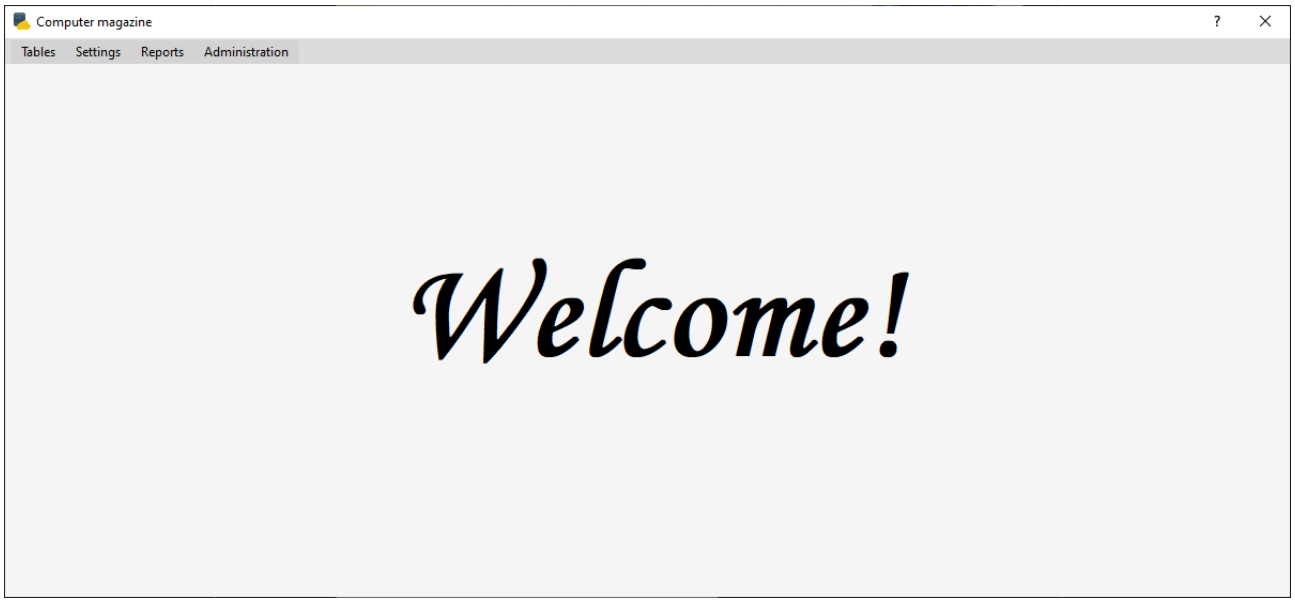






	Product id	Production type id	Manufacturer id	Product name	Product price	Waranty	Picture
1	5000	2000	4005	Blye Yeti	20000	5	../../resources/...
2	5010	2010	4003	Nitro	25000	3	../../resources/...
3	5020	2020	4003	Spin 5	5000	1	../../resources/...
4	5021	2020	4004	Hp one	2000	1	../../resources/...
5	5030	2030	4002	Redmi 5 plus	4000	1	../../resources/...
6	5040	2040	4001	Prestige Pro 3	4700	2	../../resources/...
7	5050	2050	4004	All Vew Pro	2000	3	../../resources/...
8	5051	2050	4001	Clear Sky	3000	4	../../resources/...
9	5052	2050	4001	Colors	2500	2	../../resources/...

Перезапускаем приложение…



Фрагмент кода

```
using Microsoft.ReportingServices.ReportProcessing.ReportObjectModel;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Net.Mail;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
```

```
{
    public partial class Form1 : Form
    {
        private string cfg = "../../config/config.cfg";
        Dictionary<string, string> config;
        private string table;
        private ErrorProvider error = new ErrorProvider();
        private string user;
        private string user_login;

        Computer_magazineDataSet ds;

        private string style;

        public Form1()
        {
            InitializeComponent();


            config = new Dictionary<string, string>();


            //reading saved config
            using (StreamReader file = new StreamReader(cfg))
            {
                string[] lines = file.ReadToEnd().Split('\n');

                foreach(string line in lines)
                {
                    //MessageBox.Show(line);
                    if(line != "")
                        config.Add((line.Split(':'))[0], (line.Split(':'))[1]);
                }
            }

            if (config.ContainsKey("style"))
            {
                //style = config["style"];
                config.TryGetValue("style", out style);
                style = style.Replace("\r", ""); //\r - perenos karetki

                //MessageBox.Show(style);

                if (style.Equals("white"))
                {
                    this.BackColor = Color.WhiteSmoke;
                    dataGridView1.BackgroundColor= Color.WhiteSmoke;
                    menuStrip1.BackColor= Color.Gainsboro;
                    panel1.BackColor= Color.LightGray;
                    panel2.BackColor= Color.LightGray;
                    panel3.BackColor= Color.LightGray;
                    tablesToolStripMenuItem.BackColor= Color.LightGray;
                    settingsToolStripMenuItem.BackColor= Color.LightGray;
                    reportsToolStripMenuItem.BackColor= Color.LightGray;
                    rolesToolStripMenuItem.BackColor = Color.LightGray;

                    label1.ForeColor= Color.Black;
                    label2.ForeColor= Color.Black;
                    label3.ForeColor= Color.Black;
                    label4.ForeColor= Color.Black;
                    label5.ForeColor= Color.Black;
                    label6.ForeColor= Color.Black;
                    label7.ForeColor= Color.Black;
                    checkBox1.ForeColor= Color.Black;
                    label8.ForeColor= Color.Black;
                    label9.ForeColor= Color.Black;
                    label10.ForeColor= Color.Black;
                    label11.ForeColor= Color.Black;
                    label12.ForeColor= Color.Black;
```

```
            label13.ForeColor= Color.Black;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            menuStrip1.BackColor = Color.DimGray;
            panel1.BackColor = Color.Gray;
            panel2.BackColor = Color.Gray;
            panel3.BackColor = Color.Gray;
            tablesToolStripMenuItem.BackColor = Color.Gray;
            settingsToolStripMenuItem.BackColor = Color.Gray;
            reportsToolStripMenuItem.BackColor = Color.Gray;
            rolesToolStripMenuItem.BackColor = Color.Gray;

            label1.ForeColor = Color.WhiteSmoke;
            label2.ForeColor = Color.WhiteSmoke;
            label3.ForeColor = Color.WhiteSmoke;
            label4.ForeColor = Color.WhiteSmoke;
            label5.ForeColor = Color.WhiteSmoke;
            label6.ForeColor = Color.WhiteSmoke;
            label7.ForeColor = Color.WhiteSmoke;
            checkBox1.ForeColor = Color.WhiteSmoke;
            label8.ForeColor = Color.WhiteSmoke;
            label9.ForeColor = Color.WhiteSmoke;
            label10.ForeColor = Color.WhiteSmoke;
            label11.ForeColor = Color.WhiteSmoke;
            label12.ForeColor = Color.WhiteSmoke;
            label13.ForeColor = Color.WhiteSmoke;
        }
    }


    comboBox1.Items.Add("Consign.№");
    comboBox1.Items.Add("Date");


    helpProvider1.HelpNamespace = "../../help/help.html";
}

private void Form1_Load(object sender, EventArgs e)
{
    RegistrationForm rf = new RegistrationForm(this);
    this.Hide();
    rf.ShowDialog();


    if (user == "user")
    {
        editToolStripMenuItem.Visible = false;
        editToolStripMenuItem1.Visible = false;
        editToolStripMenuItem2.Visible = false;
        editToolStripMenuItem3.Visible = false;
        editToolStripMenuItem4.Visible = false;
        editToolStripMenuItem5.Visible = false;
        editToolStripMenuItem6.Visible = false;
        editToolStripMenuItem7.Visible = false;
        editToolStripMenuItem8.Visible = false;
        editToolStripMenuItem9.Visible = false;
        editToolStripMenuItem10.Visible = false;
        editToolStripMenuItem11.Visible = false;
        editToolStripMenuItem12.Visible = false;

        chekToolStripMenuItem.Visible = false;
        consignmentToolStripMenuItem.Visible = false;
        chekInfoToolStripMenuItem.Visible = false;
        pasportDataToolStripMenuItem.Visible = false;
```

```
        addToolStripMenuItem.Visible = false;
        toolStripMenuItem1.Visible = false;
        toolStripMenuItem2.Visible = false;
        toolStripMenuItem3.Visible = false;
        toolStripMenuItem4.Visible = false;
        toolStripMenuItem5.Visible = false;
        toolStripMenuItem6.Visible = false;
        toolStripMenuItem7.Visible = false;
        toolStripMenuItem8.Visible = false;
        toolStripMenuItem9.Visible = false;
        toolStripMenuItem10.Visible = false;
        toolStripMenuItem11.Visible = false;
        addToolStripMenuItem1.Visible = false;

        reportsToolStripMenuItem.Visible = false;

        rolesToolStripMenuItem.Visible = false;
    }

    if (user == "admin")
    {
        editToolStripMenuItem.Visible = true;
        editToolStripMenuItem1.Visible = true;
        editToolStripMenuItem2.Visible = true;
        editToolStripMenuItem3.Visible = true;
        editToolStripMenuItem4.Visible = true;
        editToolStripMenuItem5.Visible = true;
        editToolStripMenuItem6.Visible = true;
        editToolStripMenuItem7.Visible = true;
        editToolStripMenuItem8.Visible = true;
        editToolStripMenuItem9.Visible = true;
        editToolStripMenuItem10.Visible = true;
        editToolStripMenuItem11.Visible = true;
        editToolStripMenuItem12.Visible = true;

        addToolStripMenuItem.Visible = true;
        toolStripMenuItem1.Visible = true;
        toolStripMenuItem2.Visible = true;
        toolStripMenuItem3.Visible = true;
        toolStripMenuItem4.Visible = true;
        toolStripMenuItem5.Visible = true;
        toolStripMenuItem6.Visible = true;
        toolStripMenuItem7.Visible = true;
        toolStripMenuItem8.Visible = true;
        toolStripMenuItem9.Visible = true;
        toolStripMenuItem10.Visible = true;
        toolStripMenuItem11.Visible = true;
        addToolStripMenuItem1.Visible = true;

        reportsToolStripMenuItem.Visible = true;

        rolesToolStripMenuItem.Visible = true;
    }

    if (user == "seller")
    {
        editToolStripMenuItem.Visible = false;
        editToolStripMenuItem1.Visible = false;
        editToolStripMenuItem2.Visible = false;
        editToolStripMenuItem3.Visible = false;
        editToolStripMenuItem4.Visible = false;
        editToolStripMenuItem5.Visible = false;
        editToolStripMenuItem6.Visible = false;
        editToolStripMenuItem7.Visible = false;
        editToolStripMenuItem8.Visible = false;
        editToolStripMenuItem9.Visible = false;
        editToolStripMenuItem10.Visible = false;
        editToolStripMenuItem11.Visible = false;
        editToolStripMenuItem12.Visible = false;
```

```
            addToolStripMenuItem.Visible = false;
            toolStripMenuItem1.Visible = false;
            toolStripMenuItem2.Visible = false;
            toolStripMenuItem3.Visible = false;
            toolStripMenuItem4.Visible = false;
            toolStripMenuItem5.Visible = false;
            toolStripMenuItem6.Visible = false;
            toolStripMenuItem7.Visible = false;
            toolStripMenuItem8.Visible = true;
            toolStripMenuItem9.Visible = false;
            toolStripMenuItem10.Visible = true;
            toolStripMenuItem11.Visible = false;
            addToolStripMenuItem1.Visible = true;

            reportsToolStripMenuItem.Visible = false;

            rolesToolStripMenuItem.Visible = false;
        }

        if (user == "manager")
        {
            editToolStripMenuItem.Visible = false;
            editToolStripMenuItem1.Visible = false;
            editToolStripMenuItem2.Visible = false;
            editToolStripMenuItem3.Visible = false;
            editToolStripMenuItem4.Visible = false;
            editToolStripMenuItem5.Visible = false;
            editToolStripMenuItem6.Visible = false;
            editToolStripMenuItem7.Visible = false;
            editToolStripMenuItem8.Visible = true;
            editToolStripMenuItem9.Visible = false;
            editToolStripMenuItem10.Visible = true;
            editToolStripMenuItem11.Visible = false;
            editToolStripMenuItem12.Visible = true;

            addToolStripMenuItem.Visible = false;
            toolStripMenuItem1.Visible = false;
            toolStripMenuItem2.Visible = false;
            toolStripMenuItem3.Visible = false;
            toolStripMenuItem4.Visible = false;
            toolStripMenuItem5.Visible = false;
            toolStripMenuItem6.Visible = false;
            toolStripMenuItem7.Visible = false;
            toolStripMenuItem8.Visible = true;
            toolStripMenuItem9.Visible = true;
            toolStripMenuItem10.Visible = true;
            toolStripMenuItem11.Visible = true;
            addToolStripMenuItem1.Visible = true;

            reportsToolStripMenuItem.Visible = true;

            rolesToolStripMenuItem.Visible = false;
        }
    }

    public string getStyle()
    {
        return style;
    }

    public void setStyle(string style)
    {
        config.Remove("style");
        config.Add("style", style);

        using (StreamWriter file = new StreamWriter(cfg))
        {
            foreach(string key in config.Keys)
```

```
            {
                config.TryGetValue(key, out string value);
                file.WriteLine(key+":"+value);
            }
        }

        this.style = style;

        if (style == "white")
        {
            this.BackColor = Color.WhiteSmoke;
            dataGridView1.BackgroundColor= Color.WhiteSmoke;
            menuStrip1.BackColor = Color.Gainsboro;
            panel1.BackColor = Color.LightGray;
            panel2.BackColor = Color.LightGray;
            panel3.BackColor = Color.LightGray;
            tablesToolStripMenuItem.BackColor = Color.LightGray;
            settingsToolStripMenuItem.BackColor = Color.LightGray;
            reportsToolStripMenuItem.BackColor = Color.LightGray;
            rolesToolStripMenuItem.BackColor = Color.LightGray;

            label1.ForeColor = Color.Black;
            label2.ForeColor = Color.Black;
            label3.ForeColor = Color.Black;
            label4.ForeColor = Color.Black;
            label5.ForeColor = Color.Black;
            label6.ForeColor = Color.Black;
            label7.ForeColor = Color.Black;
            checkBox1.ForeColor = Color.Black;

            label8.ForeColor = Color.Black;
            label9.ForeColor = Color.Black;
            label10.ForeColor = Color.Black;
            label11.ForeColor = Color.Black;
            label12.ForeColor = Color.Black;
            label13.ForeColor = Color.Black;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            menuStrip1.BackColor = Color.DimGray;
            panel1.BackColor = Color.Gray;
            panel2.BackColor = Color.Gray;
            panel3.BackColor = Color.Gray;
            tablesToolStripMenuItem.BackColor = Color.Gray;
            settingsToolStripMenuItem.BackColor = Color.Gray;
            reportsToolStripMenuItem.BackColor = Color.Gray;
            rolesToolStripMenuItem.BackColor = Color.Gray;

            label1.ForeColor = Color.WhiteSmoke;
            label2.ForeColor = Color.WhiteSmoke;
            label3.ForeColor = Color.WhiteSmoke;
            label4.ForeColor = Color.WhiteSmoke;
            label5.ForeColor = Color.WhiteSmoke;
            label6.ForeColor = Color.WhiteSmoke;
            label7.ForeColor = Color.WhiteSmoke;
            checkBox1.ForeColor = Color.WhiteSmoke;

            label8.ForeColor = Color.WhiteSmoke;
            label9.ForeColor = Color.WhiteSmoke;
            label10.ForeColor = Color.WhiteSmoke;
            label11.ForeColor = Color.WhiteSmoke;
            label12.ForeColor = Color.WhiteSmoke;
            label13.ForeColor = Color.WhiteSmoke;
        }
    }
```

```
public string getUser()
{
    return this.user;
}

public void setUser(string user)
{
    this.user = user;
}

public string getUserLogin()
{
    return this.user_login;
}

public void setUserLogin(String user_login)
{
    this.user_login = user_login;
}



private void styleToolStripMenuItem_Click(object sender, EventArgs e)
{
    StyleForm style = new StyleForm(this);
    style.ShowDialog();
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    updateManufacturer();
}

private void openToolStripMenuItem1_Click(object sender, EventArgs e)
{
    updateSupplier();
}

...


private void addToolStripMenuItem_Click(object sender, EventArgs e)
{
    ManEditForm add = new ManEditForm(this, "add");
    add.ShowDialog();
    updateManufacturer();
}
private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    ManEditForm edit = new ManEditForm(this, "edit");
    edit.ShowDialog();
    updateManufacturer();
}



private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    SupEditForm add = new SupEditForm(this, "add");
    add.ShowDialog();
    updateSupplier();
}

private void editToolStripMenuItem1_Click(object sender, EventArgs e)
{
    SupEditForm edit = new SupEditForm(this, "edit");
    edit.ShowDialog();
    updateSupplier();
```

```
}



...

private void editToolStripMenuItem8_Click(object sender, EventArgs e)
{
    if (user == "manager")
    {
        ChekEditForm edit = new ChekEditForm(this, "edit-only");
        edit.ShowDialog();
        updateChek();
    }
    else
    {
        ChekEditForm edit = new ChekEditForm(this, "edit");
        edit.ShowDialog();
        updateChek();
    }

}



...



private void updateManufacturer()
{
    panel1.Visible = true;
    panel2.Visible = true;
    panel3.Visible = true;
    label13.Visible = false;
    pictureBox1.Visible = false;
    //search
    label1.Visible= true;
    label2.Visible= false;
    textBox1.Visible= false;
    button1.Visible= false;
    button2.Visible= false;
    comboBox1.Visible = false;
    dateTimePicker1.Visible = false;

    //filter
    label3.Visible=true;
    label4.Visible = false;
    label5.Visible = false;
    label6.Visible = false;
    label7.Visible = false;
    comboBox2.Items.Clear();
    comboBox2.Visible = false;
    comboBox3.Visible = false;
    comboBox4.Visible = false;
    comboBox5.Visible = false;
    checkBox1.Visible = false;
    button3.Visible = false;
    button4.Visible = false;
    textBox5.Visible = false;
    textBox3.Visible = false;
    textBox4.Visible = false;

    //calculs

    label8.Visible = false;
    label9.Visible = false;
    label10.Visible = false;
```

```
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;

            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from manufacturer";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 4;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Manufacturer id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Manufacturer name";
                    dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Production price";
                    if (user == "user") dataGridView1.Columns[0].Visible = false;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        public void updateSupplier()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
```

```
            label13.Visible = false;
            //searchs
            label1.Visible = true;
            label2.Visible = true;
            label2.Text = "Telephone: ";
            table = "supplier";
            textBox1.Visible = true;
            button1.Visible = true;
            button2.Visible = true;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;


            //filter
            label3.Visible = true;
            label4.Visible = true;
            label4.Text = "Prod. type:";
            comboBox2.Visible = true;
            comboBox2.Items.Clear();
            comboBox2.Items.Add("all");
            comboBox2.Items.Add("PC");
            comboBox2.Items.Add("Laptop");
            comboBox2.Items.Add("Monitor");
            comboBox2.Items.Add("Phone");
            comboBox2.Items.Add("Printer");
            comboBox2.Items.Add("Tablet");
            comboBox2.SelectedIndex = 0;
            button3.Visible = true;
            button4.Visible = true;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;


            //calculs


            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;



            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
```

```
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from supplier";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Supplier name";
                    dataGridView1.Columns[2].HeaderCell.Value = "Supplier adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Production type";
                    dataGridView1.Columns[4].HeaderCell.Value = "Telephones";
                    dataGridView1.Columns[5].HeaderCell.Value = "Manufactirer id";
                    if (user == "user")
                    {
                        dataGridView1.Columns[0].Visible = false;
                        dataGridView1.Columns[5].Visible = false;
                    }
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }


        ...


        private void manufacturerToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateManufacturer();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void suplToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateSupplier();
            tablesToolStripMenuItem.HideDropDown();
        }

        ...
```

```
        private void pasportDataToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updatePasportData();
        }




        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if(table == "supplier")
            {
                if(!Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\+(\\d+)?$") && (int)e.KeyChar !=
8)
                {
                    e.Handled = true;
                }
            } else if(table == "consignment")
            {
                if (!Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
                {
                    e.Handled = true;
                }
            }
        }




        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;
            if(table == "supplier" &&
!Regex.IsMatch(textBox1.Text,"^\\+373\\d\\d\\d\\d\\d\\d\\d$"))
            {
                flag = false;
                error.SetError(textBox1, "Number should be in forat +373xxxxxxxx");
            }

            try {
                if (table == "supplier" && flag)
                {
                    string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    SqlConnection connection = new SqlConnection(connectionString);
                    using (connection)
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from supplier where
Telephone = @Telephone", connection);

                        command.Parameters.Add("@Telephone", textBox1.Text);

                        SqlDataReader reader = command.ExecuteReader();

                        if (reader.HasRows)
                        {
                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
```

```
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                        }
                    }
                }



                if (table == "consignment" && flag)
                {
                    string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    SqlConnection connection = new SqlConnection(connectionString);
                    if (comboBox1.Text == "Date")
                    {
                        //MessageBox.Show(dateTimePicker1.Text + " == " +
dateTimePicker1.Value.ToString());
                        using (connection)
                        {
                            connection.Open();

                            SqlCommand command = new SqlCommand("select * from consignment where
Consignment_date = @Consignment_date", connection);

                            command.Parameters.Add("@Consignment_date", dateTimePicker1.Value);

                            SqlDataReader reader = command.ExecuteReader();

                            //MessageBox.Show(reader.HasRows.ToString());

                            if (reader.HasRows)
                            {
                                dataGridView1.Rows.Clear();
                                int i = 0;

                                while (reader.Read())
                                {
                                    dataGridView1.RowCount++;
                                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                    dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                    i++;
                                }
                            }
                        }
                    }
                    else
                    {
                        using (connection)
                        {
                            connection.Open();

                            SqlCommand command = new SqlCommand("select * from consignment where
consignment_number = @consignment_number", connection);

                            command.Parameters.Add("@consignment_number", textBox1.Text);

                            SqlDataReader reader = command.ExecuteReader();
```

```
                        if (reader.HasRows)
                        {
                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                        }
                    }
                }
            }

            ...
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }



        textBox1.Text = "";
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if(table == "supplier")
        {
            updateSupplier();
        }
        if(table == "consignment")
        {
            updateConsignment();
        }
        if(table == "chekInfo")
        {
            updateChekInfoChek();
        }
        if(table == "product")
        {
            updateProduct();
        }
        if(table == "pasport")
        {
            updatePasportData();
        }
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if(comboBox1.Text == "Date")
        {
            dateTimePicker1.Visible = true;
            textBox1.Visible = false;
        }
        else
```

```
            {
                dateTimePicker1.Visible = false;
                textBox1.Visible = true;
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if(table == "supplier")
            {
                updateSupplier();
            }
            if (table == "monitor")
            {
                updateMonitor();
            }
            if (table == "phone")
            {
                updatePhone();
            }
            if (table == "computer")
            {
                updateComputer();
            }
            if (table == "printer")
            {
                updatePrinter();
            }
            if (table == "consignment")
            {
                updateConsignment();
            }
            if (table == "chekInfo")
            {
                updateChekInfoChek();
            }
            if (table == "product")
            {
                updateProduct();
            }


        }

        private void button4_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try {
                if(table == "supplier")
                {
                    using (connection)
                    {
                        connection.Open();

                        using (SqlCommand command = new SqlCommand("select * from supplier where
production_type like @production_type", connection))
                        {
                            command.Parameters.Add("@production_type", comboBox2.Text == "all" ?
"%": "%"+comboBox2.Text + "%");

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                                int i = 0;

                                while (reader.Read())
```

```
                                {
                                    dataGridView1.RowCount++;
                                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                    i++;
                                }
                                dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                                dataGridView1.Columns[1].HeaderCell.Value = "Supplier name";
                                dataGridView1.Columns[2].HeaderCell.Value = "Supplier adres";
                                dataGridView1.Columns[3].HeaderCell.Value = "Production type";
                                dataGridView1.Columns[4].HeaderCell.Value = "Telephones";
                                dataGridView1.Columns[5].HeaderCell.Value = "Manufactirer id";
                            }
                        }
                    }

                    ...
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void textBox5_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!Regex.IsMatch(textBox5.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
            {
                e.Handled = true;
            }
        }

        private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
            {
                e.Handled = true;
            }
        }

        private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!Regex.IsMatch(textBox4.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
            {
                e.Handled = true;
            }
        }


        private void button9_Click(object sender, EventArgs e)
        {
            if (table == "monitor")
            {
                updateMonitor();
            }
            if (table == "phone")
            {
```

```
                updatePhone();
            }
            if (table == "computer")
            {
                updateComputer();
            }
            if (table == "consignment")
            {
                updateConsignment();
            }
            if (table == "chekInfo")
            {
                updateChekInfoChek();
            }
            if (table == "product")
            {
                updateProduct();
            }
        }

        private void button5_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "monitor")
                    {
                        using (SqlCommand command = new SqlCommand("select * from Monitor where
Diagonal = (select max(Diagonal) from Monitor)", connection))
                        {
                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                            dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                            dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                            dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                            dataGridView1.Columns[5].HeaderCell.Value = "Application";
                        }
                    }

                    .
```

```
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void button6_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "monitor")
                    {
                        using (SqlCommand command = new SqlCommand("select * from Monitor where
Diagonal = (select min(Diagonal) from Monitor)", connection))
                        {
                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                            dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                            dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                            dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                            dataGridView1.Columns[5].HeaderCell.Value = "Application";
                        }
                    }

                    ...


                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void button8_Click(object sender, EventArgs e)
        {
```

```
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "consignment")
                    {
                        using (SqlCommand command = new SqlCommand("select AVG(Price) from
consignment", connection))
                        {
                            MessageBox.Show($"Consignments' Average Price =
{command.ExecuteScalar()}MDL", "Average price", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
                        }
                    }

                    if (table == "chekInfo")
                    {
                        using (SqlCommand command = new SqlCommand("select AVG(Gen_price) from
Chek_infoChek", connection))
                        {
                            MessageBox.Show($"Average buying price =
{command.ExecuteScalar()}MDL", "Average buying price", MessageBoxButtons.OK,
MessageBoxIcon.Asterisk);
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void button7_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "consignment")
                    {
                        using (SqlCommand command = new SqlCommand("select Sum(Price) from
consignment", connection))
                        {
                            MessageBox.Show($"Total consignments' outcome =
{command.ExecuteScalar()}MDL", "Total outcome", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                        }
                    }

                    if (table == "chekInfo")
                    {
                        using (SqlCommand command = new SqlCommand("select Sum(Gen_price) from
Chek_infoChek", connection))
                        {
                            MessageBox.Show($"Total income = {command.ExecuteScalar()}MDL",
"Total income", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        }
                    }
                }
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void consignmentToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            ConsignmentReportForm consignmentRF = new ConsignmentReportForm(ds);
            consignmentRF.Show();
        }

        private void checksToolStripMenuItem_Click(object sender, EventArgs e)
        {
            CheckReportForm crf = new CheckReportForm();
            crf.Show();
        }

        private void informationToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            ProductReportForm prf = new ProductReportForm();
            prf.Show();
        }

        private void diagramToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ProductDiagramReportForm pdrf = new ProductDiagramReportForm();
            pdrf.Show();
        }

        private void addToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            PasportDataEditForm pasportData = new PasportDataEditForm(this, "add");
            pasportData.ShowDialog();
            updatePasportData();
        }

        private void editToolStripMenuItem12_Click(object sender, EventArgs e)
        {
            if (user == "manager")
            {
                PasportDataEditForm pasportData = new PasportDataEditForm(this, "edit-only");
                pasportData.ShowDialog();
                updatePasportData();
            }
            else
            {
                PasportDataEditForm pasportData = new PasportDataEditForm(this, "edit");
                pasportData.ShowDialog();
                updatePasportData();
            }
        }

        private void addNewAdminToolStripMenuItem_Click(object sender, EventArgs e)
        {
            RolesEditForm roles = new RolesEditForm(this);
            roles.ShowDialog();
        }

        private void backupDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DateTime now = DateTime.Now;
            String backupName = now.Year.ToString() + "." + now.Month.ToString() + "." +
now.Day.ToString() + "_Backup.bak";
            saveFileDialog1.FileName= backupName;
            saveFileDialog1.DefaultExt = ".bak";
```

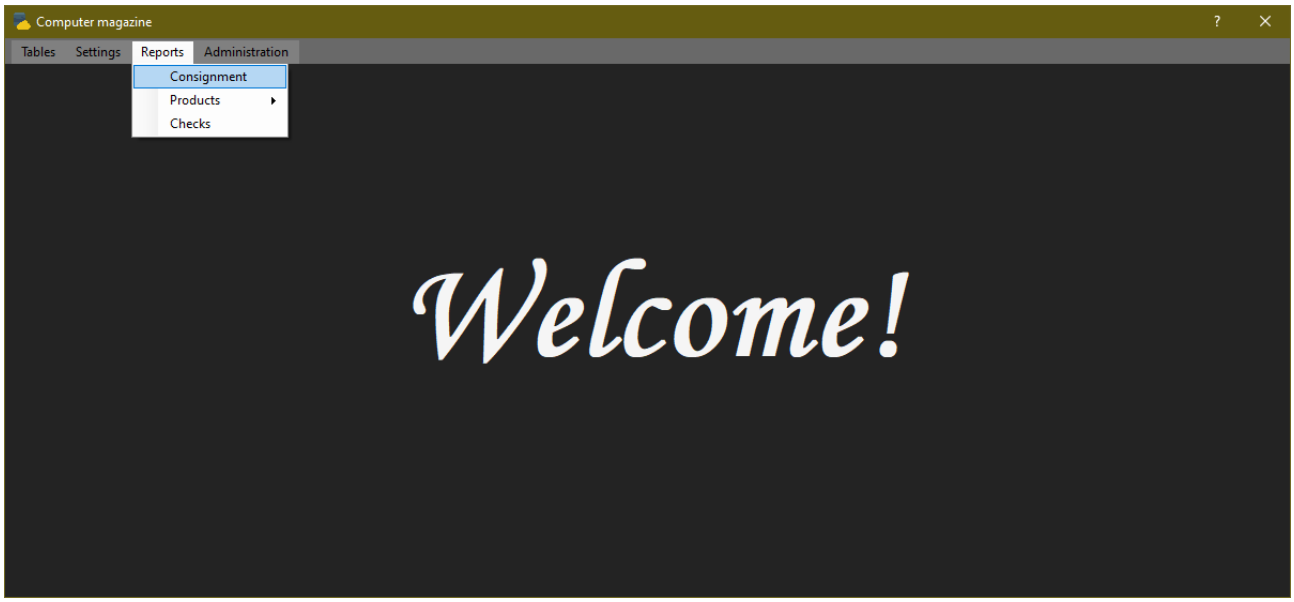
```
            if(saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(saveFileDialog1.FileName, ".+\\.bak")){

                    //MessageBox.Show(saveFileDialog1.FileName);


                    try
                    {
                        using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                        {
                            connection.Open();

                            String query = "backup database Computer_magazine to disk = @Path";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.Add("@Path", saveFileDialog1.FileName);

                            command.ExecuteNonQuery();
                        }

                        MessageBox.Show("Successfully saved!", "Success!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect backup extension!", "Invalid extension",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void restoreDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(openFileDialog1.FileName, ".+\\.bak"))
                {
                    try
                    {
                        using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                        {
                            connection.Open();

                            String query = "use master";
                            SqlCommand command = new SqlCommand(query, connection);
                            command.ExecuteNonQuery();

                            command.CommandText = "restore database Computer_magazine from disk =
@Path";
                            command.Parameters.Add("@Path", openFileDialog1.FileName);
                            command.ExecuteNonQuery();

                            command.CommandText = "use Computer_magazine";
                            command.ExecuteNonQuery();

                            MessageBox.Show("Successfully restored!", "Success!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    } catch(Exception ex)
```

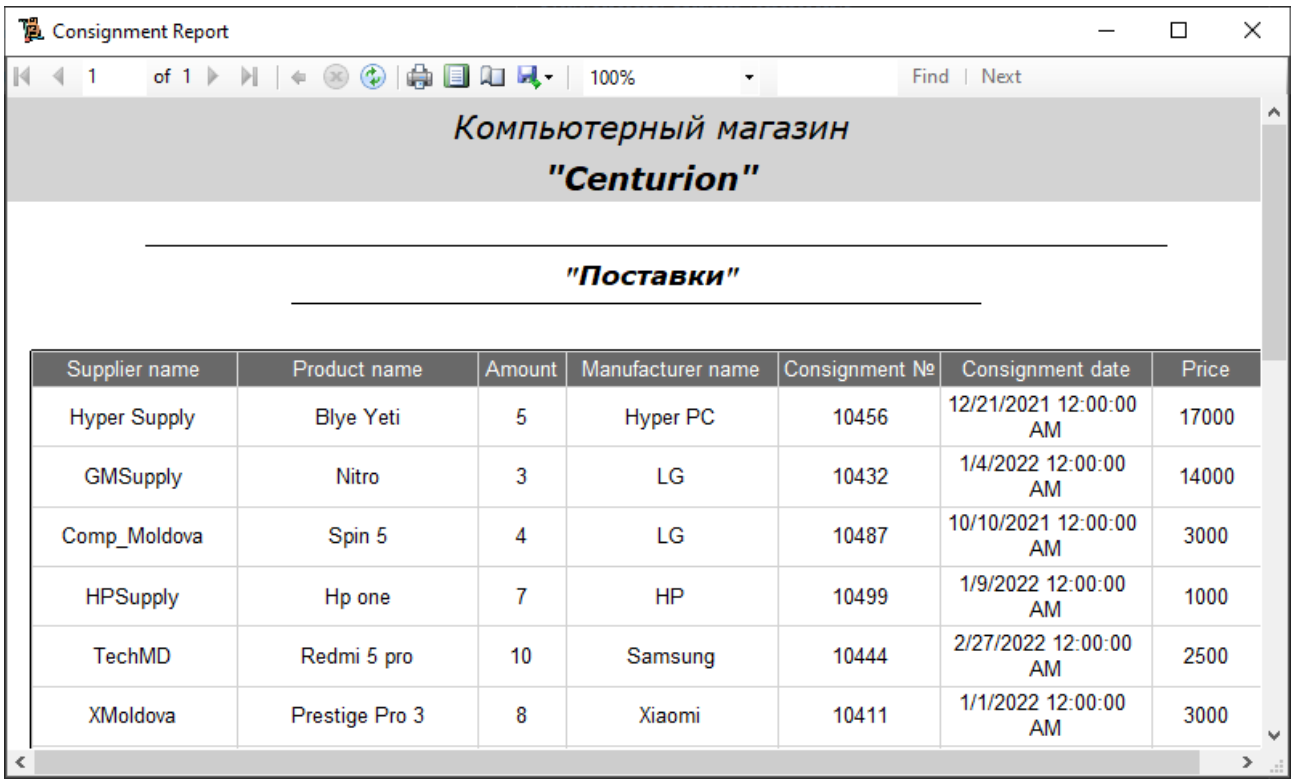
```
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect backup extension!", "Invalid extension",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            if (table == "product")
            {
                //MessageBox.Show(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
                pictureBox1.BackgroundImage =
Image.FromFile(dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString());
            }
        }
    }
}
```

Создание отчетов

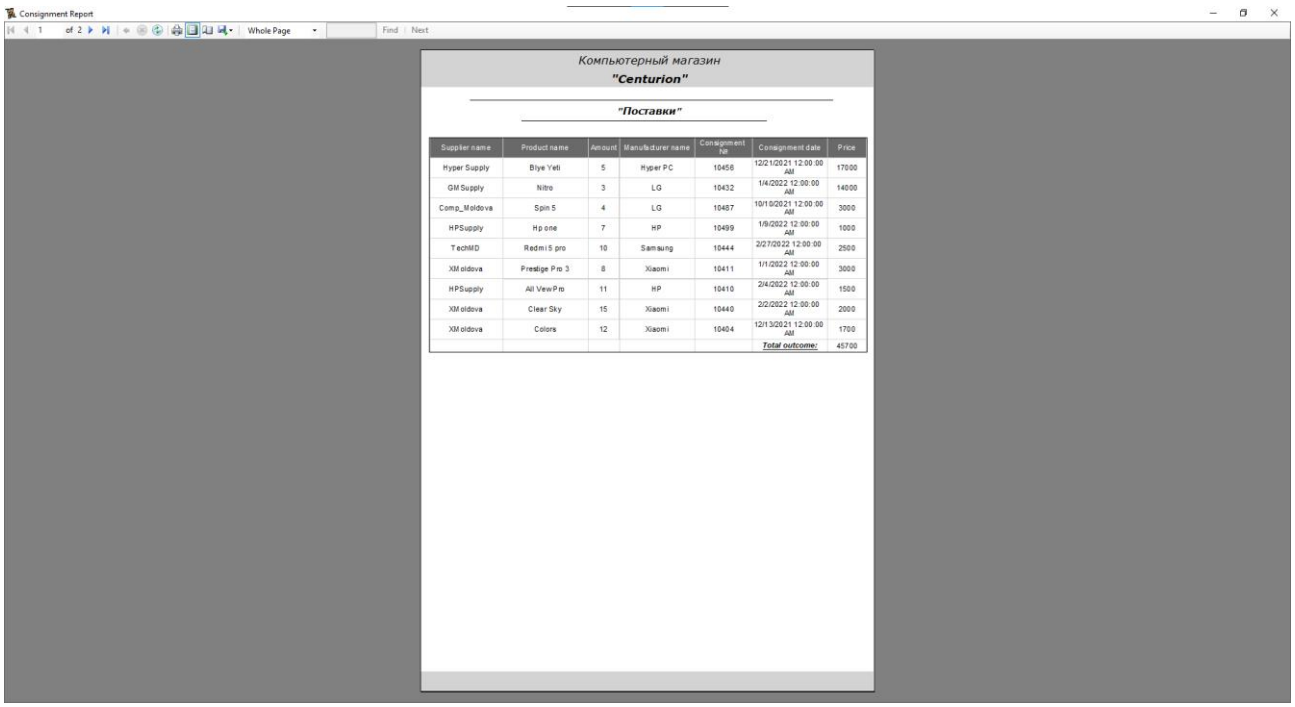
Для моего проекта были необходимы следующие отчеты:

Отчет журнала поставки:

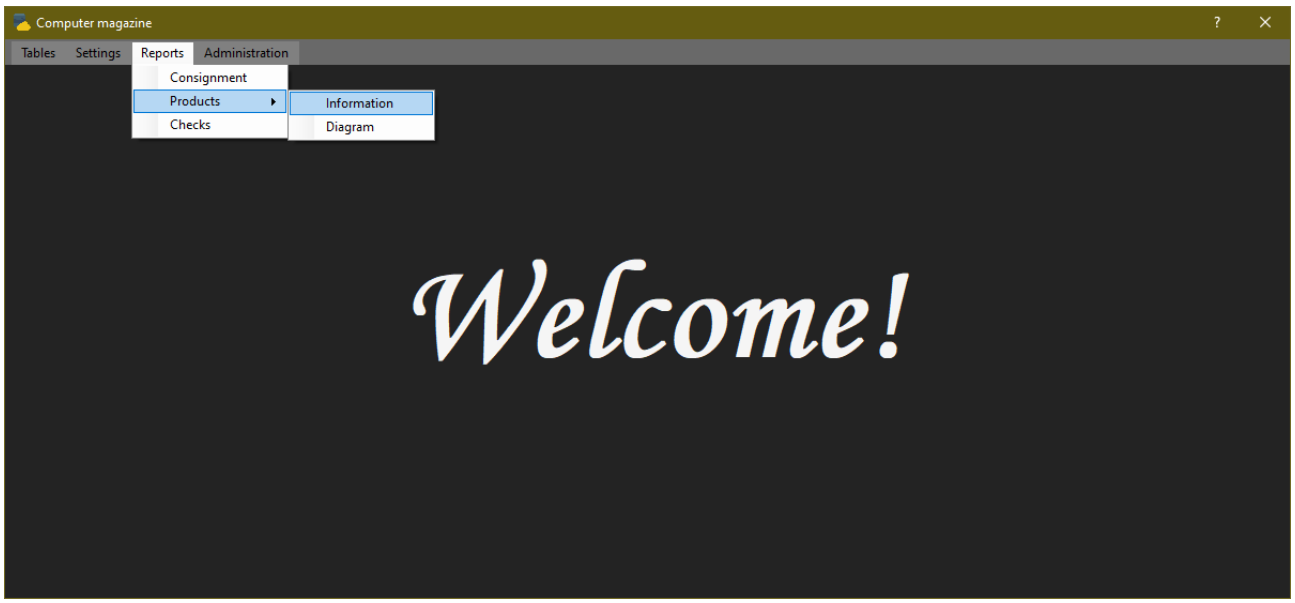




Supplier name	Product name	Amount	Manufacturer name	Consignment №	Consignment date	Price
Hyper Supply	Blye Yeti	5	Hyper PC	10456	12/21/2021 12:00:00 AM	17000
GMSupply	Nitro	3	LG	10432	1/4/2022 12:00:00 AM	14000
Comp_Moldova	Spin 5	4	LG	10487	10/10/2021 12:00:00 AM	3000
HPSupply	Hp one	7	HP	10499	1/9/2022 12:00:00 AM	1000
TechMD	Redmi 5 pro	10	Samsung	10444	2/27/2022 12:00:00 AM	2500
XMoldova	Prestige Pro 3	8	Xiaomi	10411	1/1/2022 12:00:00 AM	3000



Отчет по товарам (таблица):



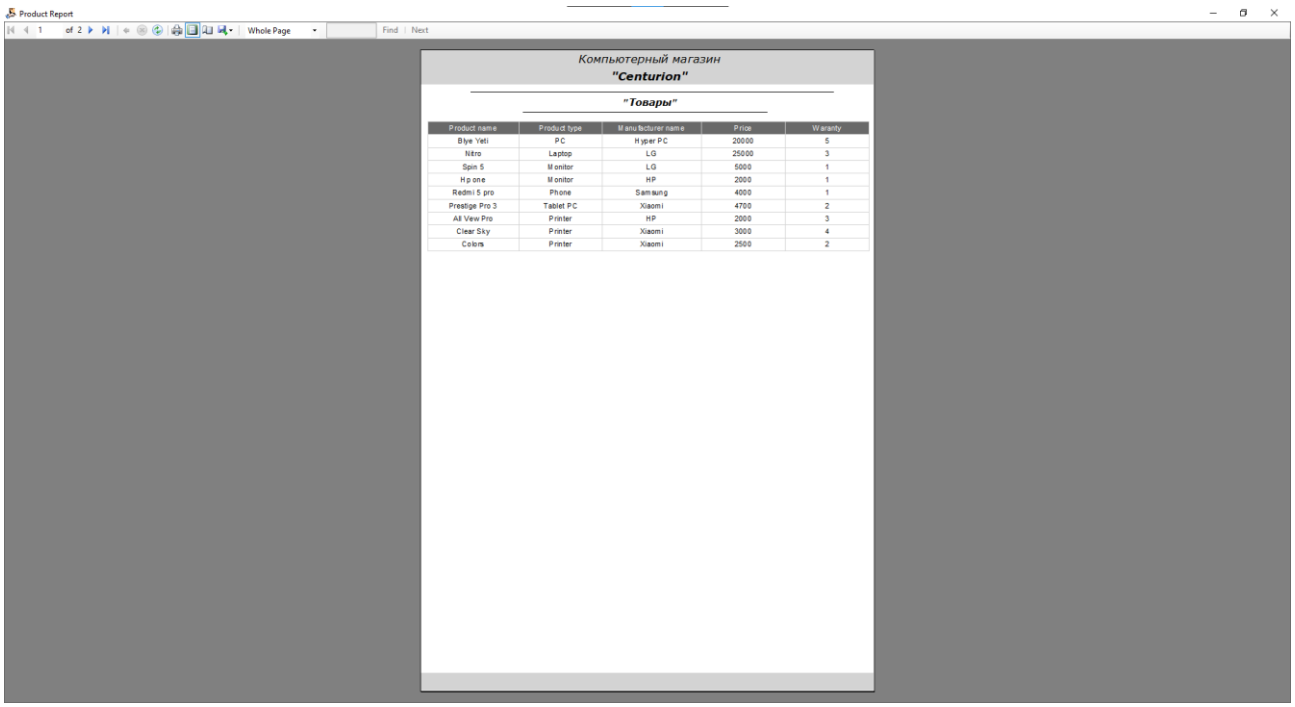
Product Report

1 of 1 100% Find | Next

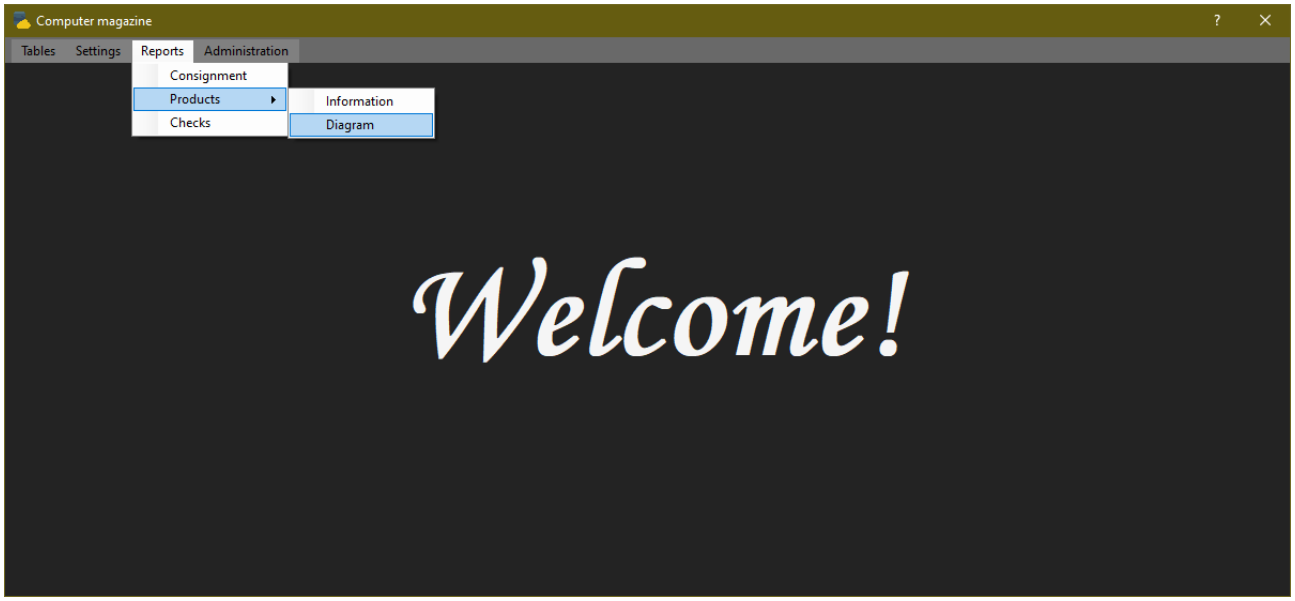
Компьютерный магазин
"Centurion"

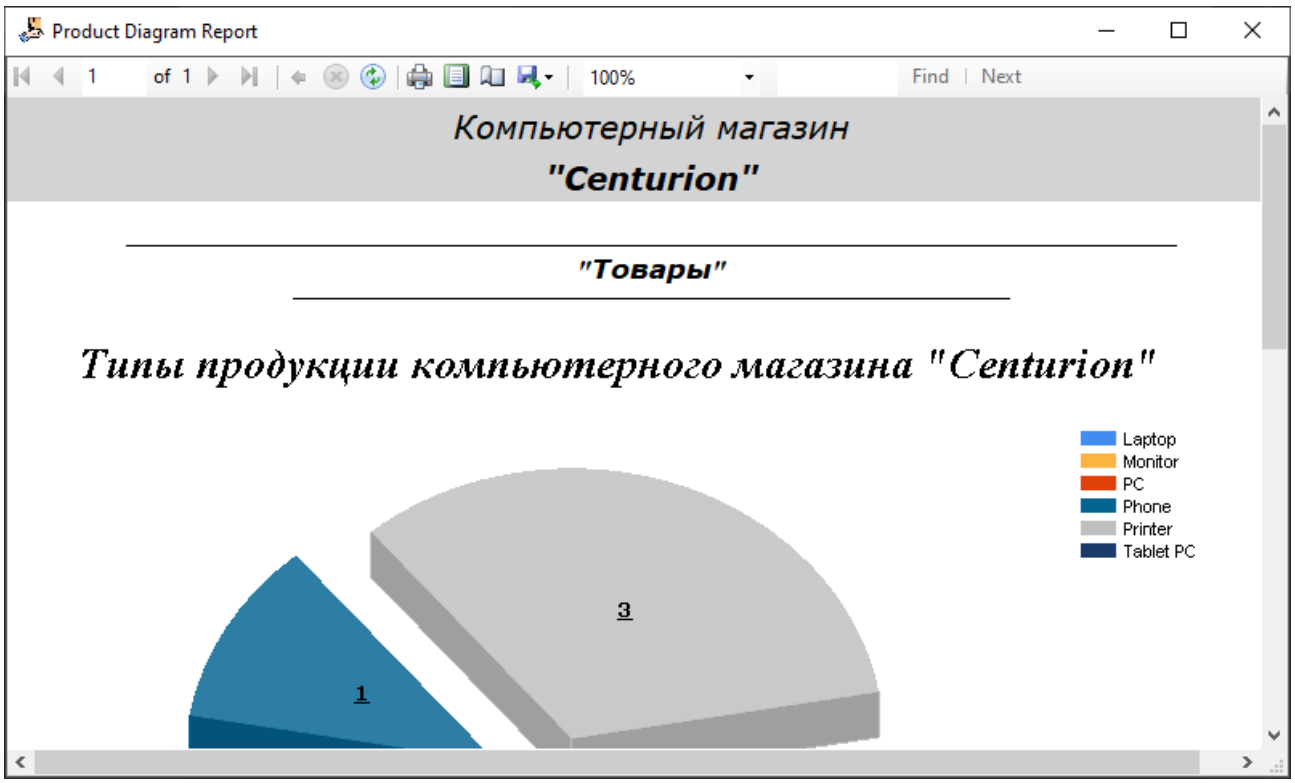
"Товары"

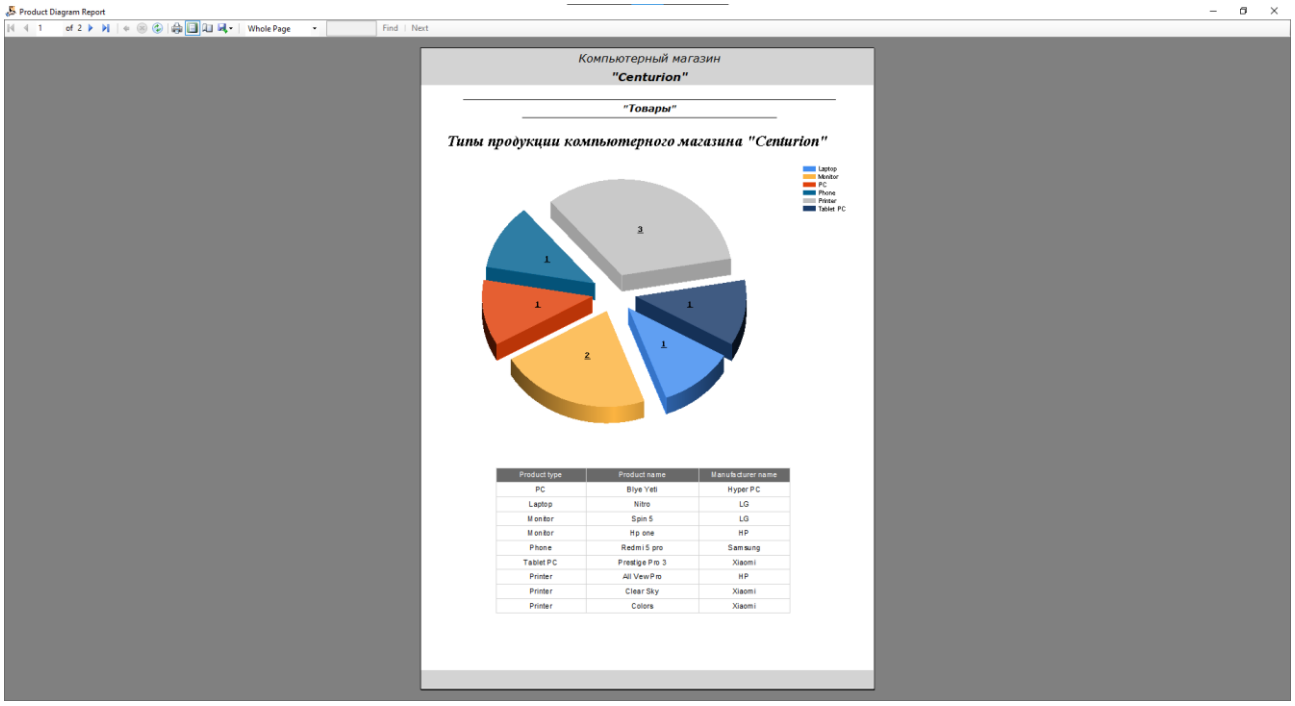
Product name	Product type	Manufacturer name	Price	Waranty
Blye Yeti	PC	Hyper PC	20000	5
Nitro	Laptop	LG	25000	3
Spin 5	Monitor	LG	5000	1
Hp one	Monitor	HP	2000	1
Redmi 5 pro	Phone	Samsung	4000	1
Prestige Pro 3	Tablet PC	Xiaomi	4700	2
All Vew Pro	Printer	HP	2000	3
Clear Sky	Printer	Xiaomi	3000	4
Colors	Printer	Xiaomi	2500	2



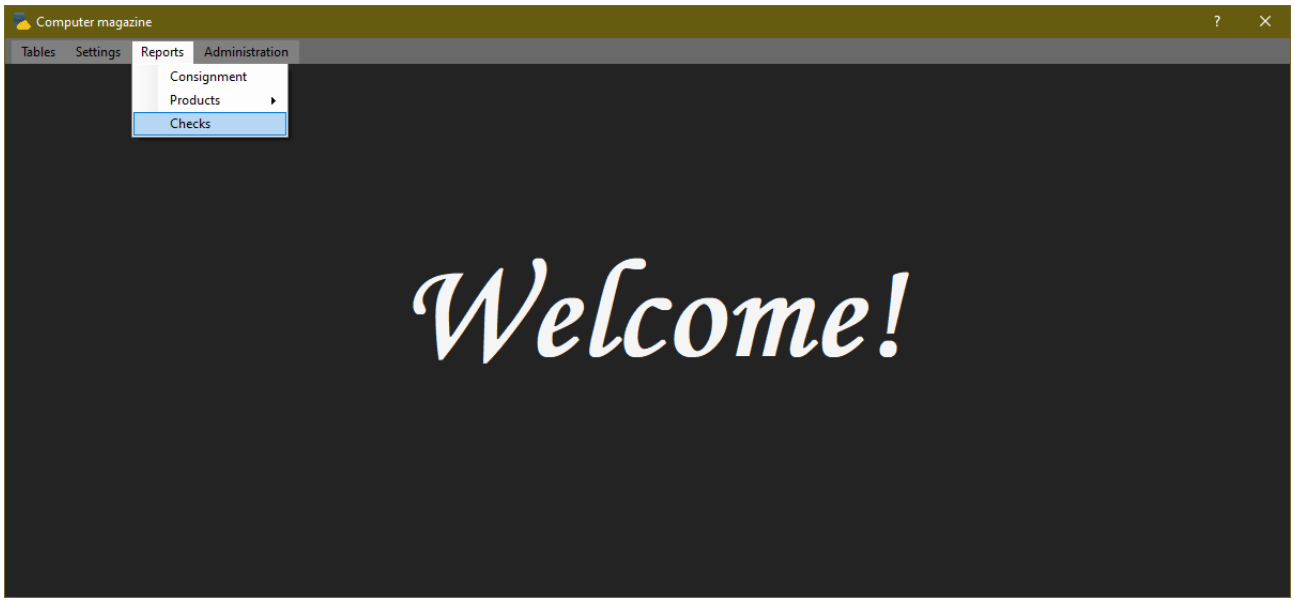
Отчет по товарам (диаграмма):

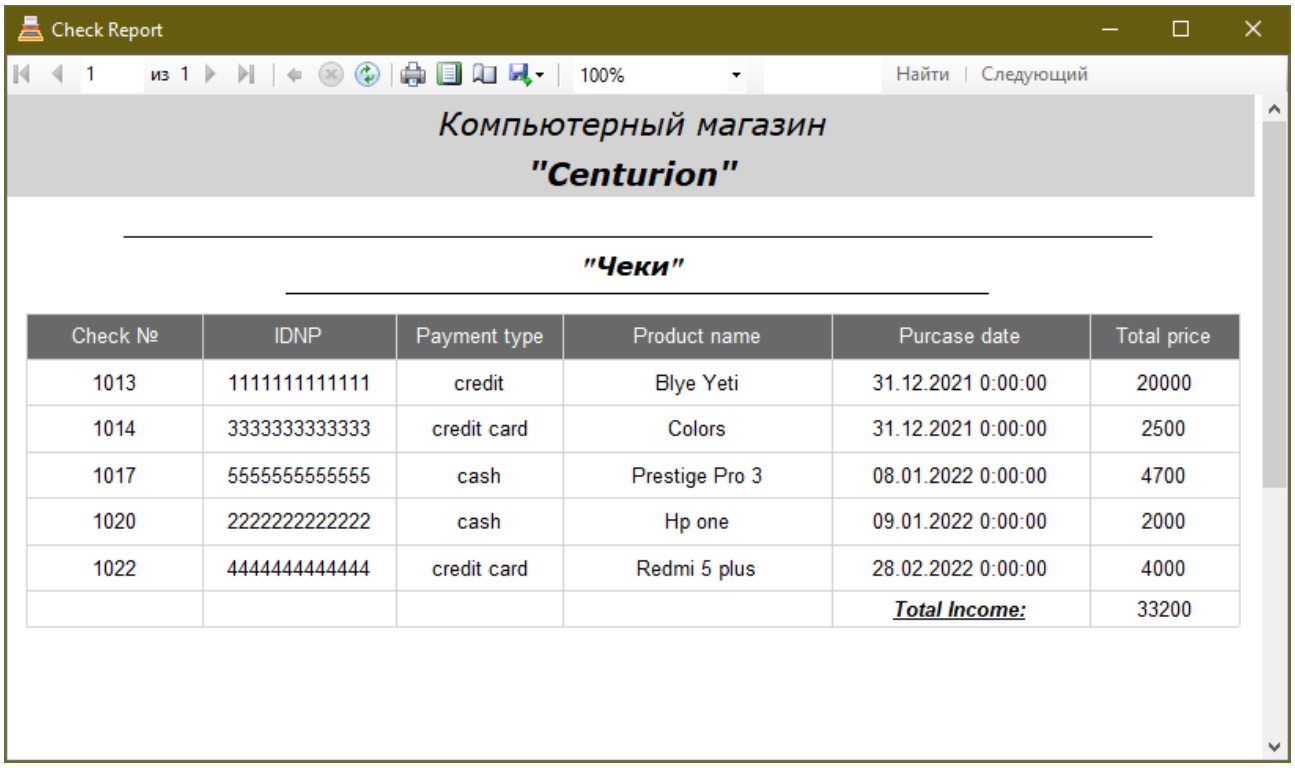



Product Diagram Report
1
of 1
100%
Find | Next
Компьютерный магазин
"Centurion"
"Товары"
Типы продукции компьютерного магазина "Centurion"
Laptop
Monitor
PC
Phone
Printer
Tablet PC
3
1

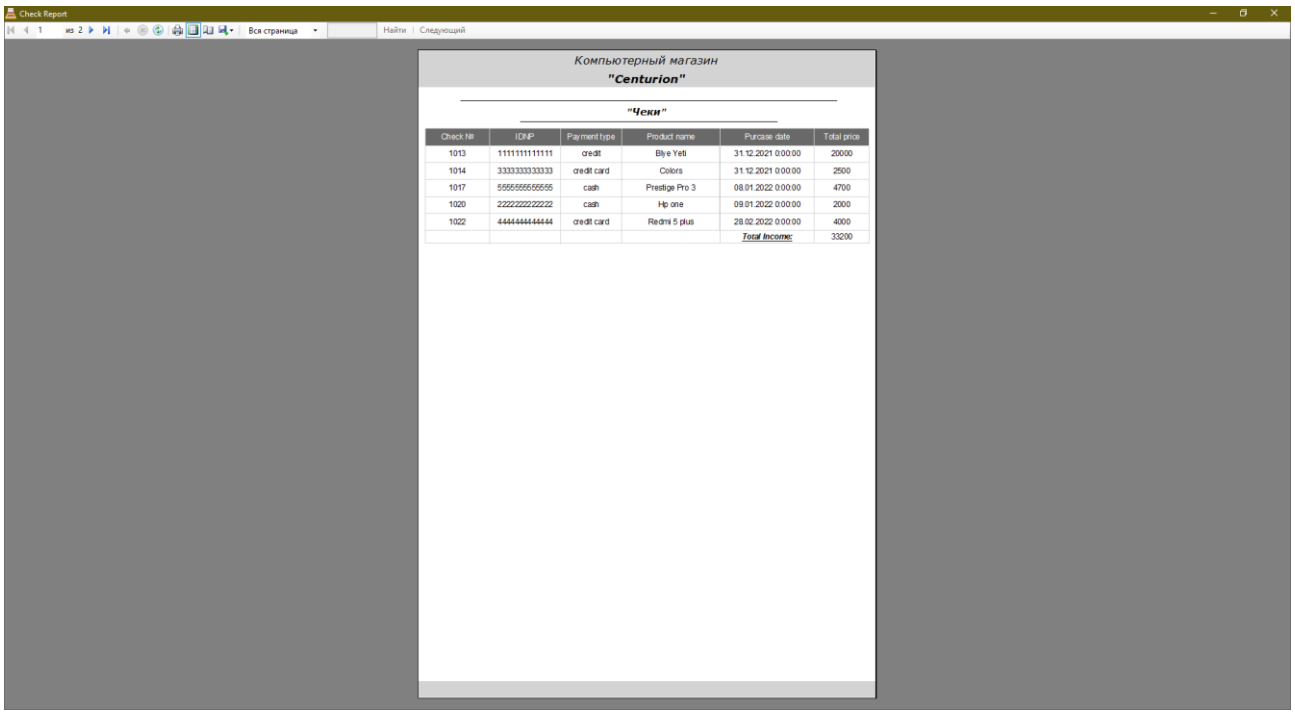

Компьютерный магазин
"Centurion"
"Товары"
Типы продукции компьютерного магазина "Centurion"
Product type
Product name
Manufacturer name
PC
Blye Yeti
Hyper PC
Laptop
Nitro
LG
Monitor
Spin 5
LG
Monitor
Hp one
HP
Phone
Redmi 5 pro
Samsung
Tablet PC
Prestige Pro 3
Xiaomi
Printer
All ViewPro
HP
Printer
Clear Sky
Xiaomi
Printer
Colors
Xiaomi

Отчет по всем чекам:





Check №	IDNP	Payment type	Product name	Purcase date	Total price
1013	1111111111111	credit	Blye Yeti	31.12.2021 0:00:00	20000
1014	3333333333333	credit card	Colors	31.12.2021 0:00:00	2500
1017	5555555555555	cash	Prestige Pro 3	08.01.2022 0:00:00	4700
1020	2222222222222	cash	Hp one	09.01.2022 0:00:00	2000
1022	4444444444444	credit card	Redmi 5 plus	28.02.2022 0:00:00	4000
				Total Income:	33200

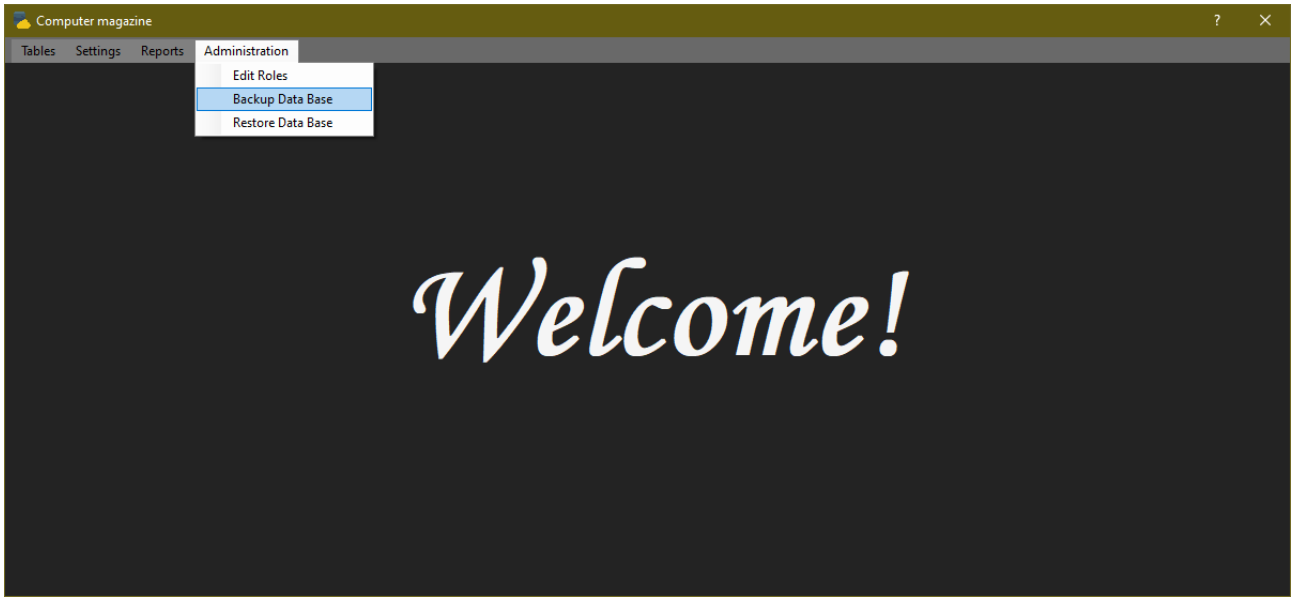


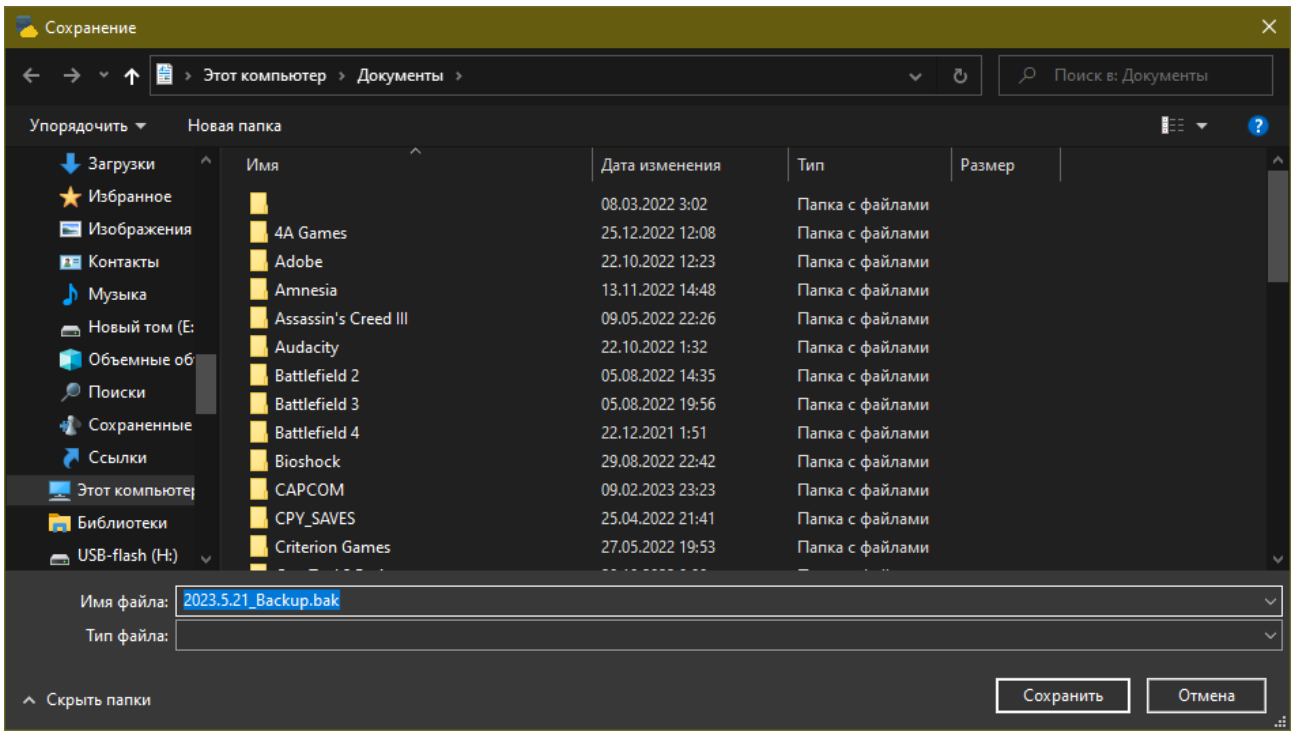
Поддержка приложения

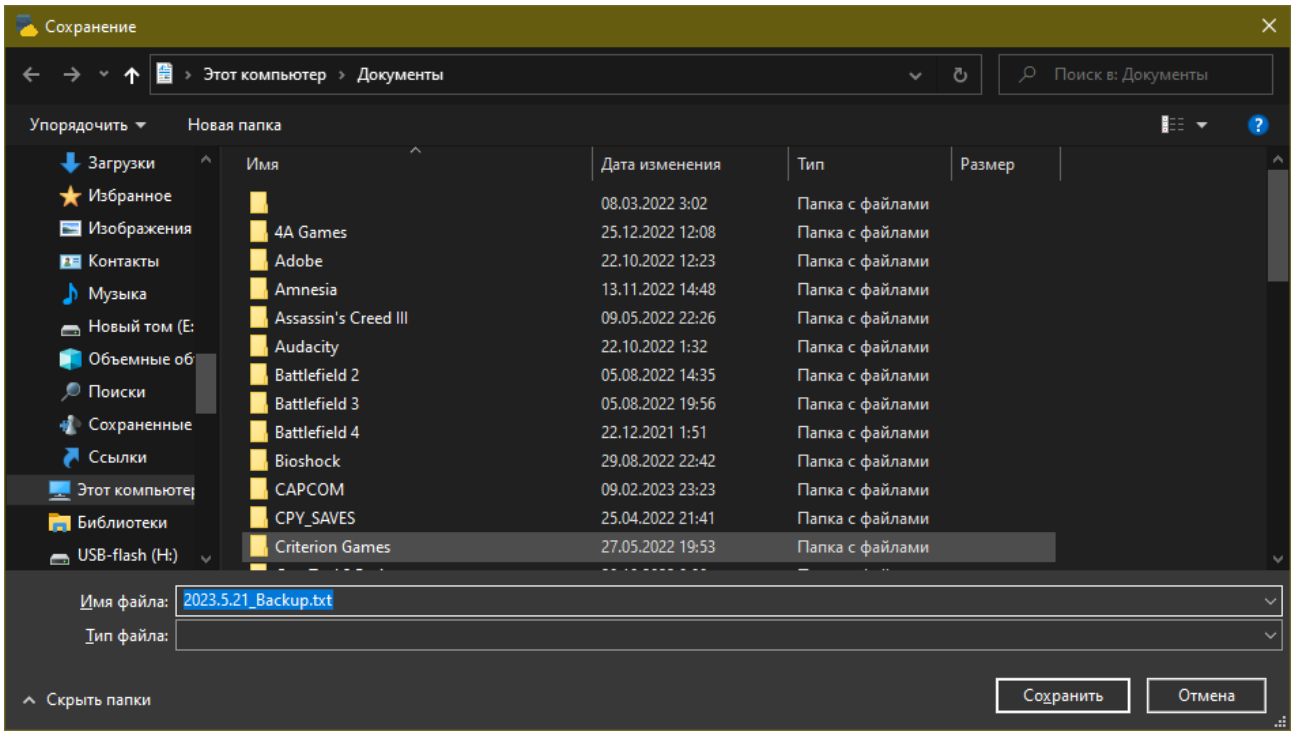
Создание резервных копий

Пользователь сам выбирает куда сохранять резервные копии и как их назвать. По умолчанию их названия – Ифслгз-‘текущая дата’.bak.

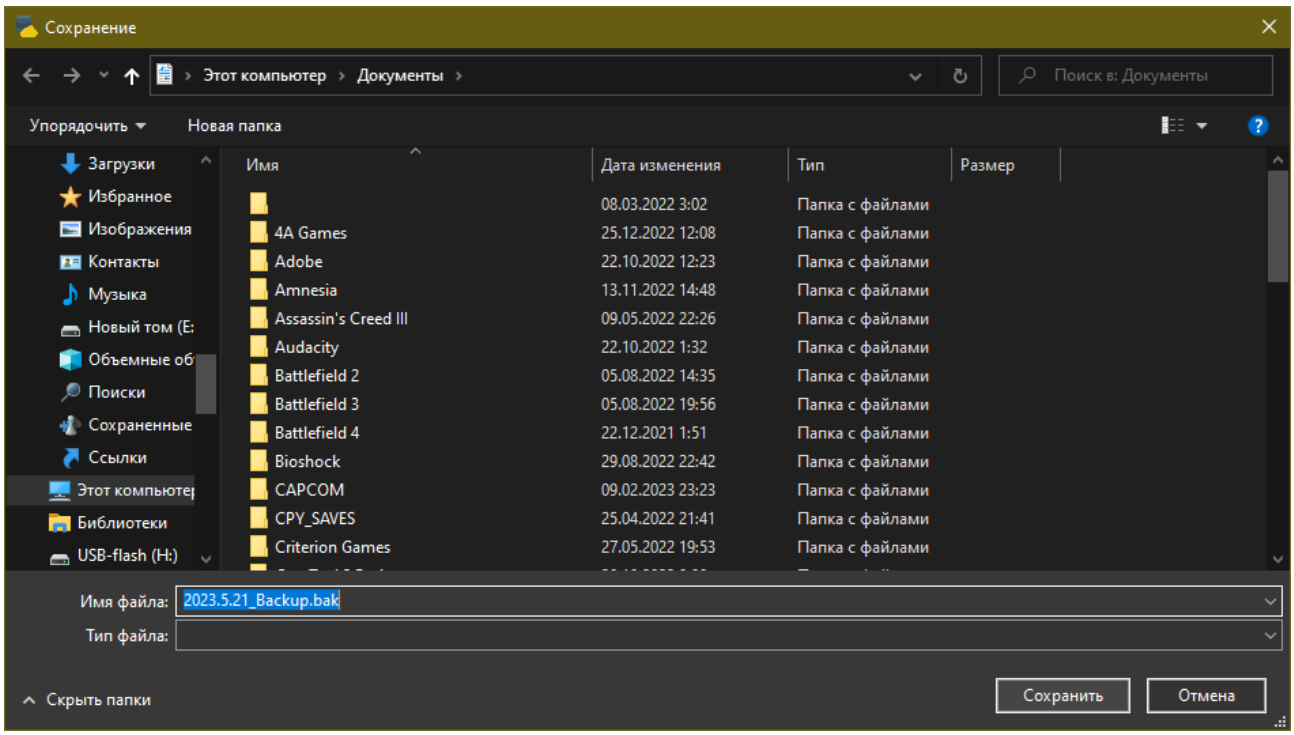
! Сохранение не файла .bak запрещено!

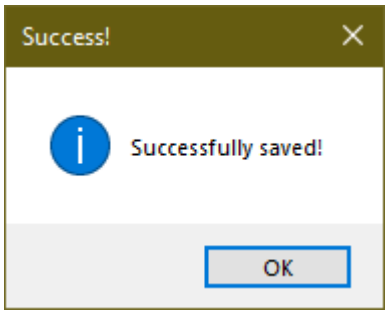



Сохранение
Этот компьютер › Документы ›
Поиск в: Документы
Упорядочить
Новая папка
Загрузки
Избранное
Изображения
Контакты
Музыка
Новый том (E:
Объемные об
Поиски
Сохраненные
Ссылки
Этот компьютер
Библиотеки
USB-flash (H:)
Имя
Дата изменения
Тип
Размер
08.03.2022 3:02 Папка с файлами
4A Games 25.12.2022 12:08 Папка с файлами
Adobe 22.10.2022 12:23 Папка с файлами
Amnesia 13.11.2022 14:48 Папка с файлами
Assassin's Creed III 09.05.2022 22:26 Папка с файлами
Audacity 22.10.2022 1:32 Папка с файлами
Battlefield 2 05.08.2022 14:35 Папка с файлами
Battlefield 3 05.08.2022 19:56 Папка с файлами
Battlefield 4 22.12.2021 1:51 Папка с файлами
Bioshock 29.08.2022 22:42 Папка с файлами
CAPCOM 09.02.2023 23:23 Папка с файлами
CPY_SAVES 25.04.2022 21:41 Папка с файлами
Criterion Games 27.05.2022 19:53 Папка с файлами
Имя файла: 2023.5.21_Backup.bak
Тип файла:
Скрыть папки
Сохранить
Отмена


Сохранение
Этот компьютер › Документы
Поиск в: Документы
Упорядочить
Новая папка
Загрузки
Избранное
Изображения
Контакты
Музыка
Новый том (E:
Объемные об
Поиски
Сохраненные
Ссылки
Этот компьютер
Библиотеки
USB-flash (H:)
Имя
Дата изменения
Тип
Размер
08.03.2022 3:02 Папка с файлами
4A Games 25.12.2022 12:08 Папка с файлами
Adobe 22.10.2022 12:23 Папка с файлами
Amnesia 13.11.2022 14:48 Папка с файлами
Assassin's Creed III 09.05.2022 22:26 Папка с файлами
Audacity 22.10.2022 1:32 Папка с файлами
Battlefield 2 05.08.2022 14:35 Папка с файлами
Battlefield 3 05.08.2022 19:56 Папка с файлами
Battlefield 4 22.12.2021 1:51 Папка с файлами
Bioshock 29.08.2022 22:42 Папка с файлами
CAPCOM 09.02.2023 23:23 Папка с файлами
CPY_SAVES 25.04.2022 21:41 Папка с файлами
Criterion Games 27.05.2022 19:53 Папка с файлами
Имя файла: 2023.5.21_Backup.txt
Тип файла:
Скрыть папки
Сохранить
Отмена


Invalid extension
Incorrect backup extension!
OK





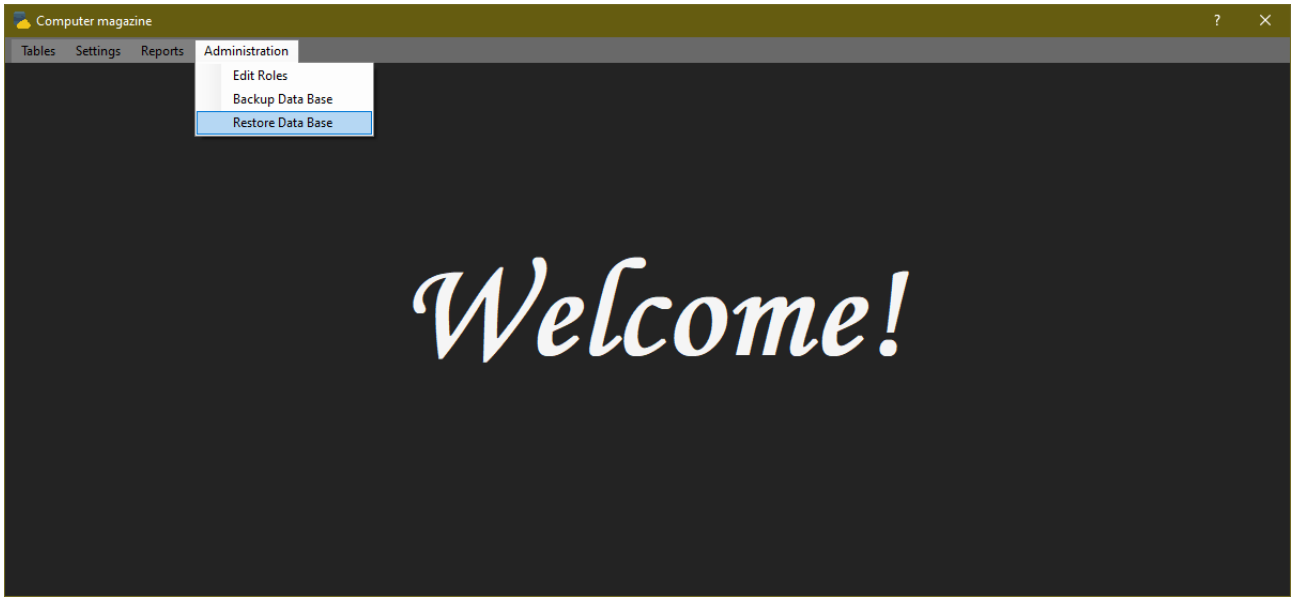
Код:

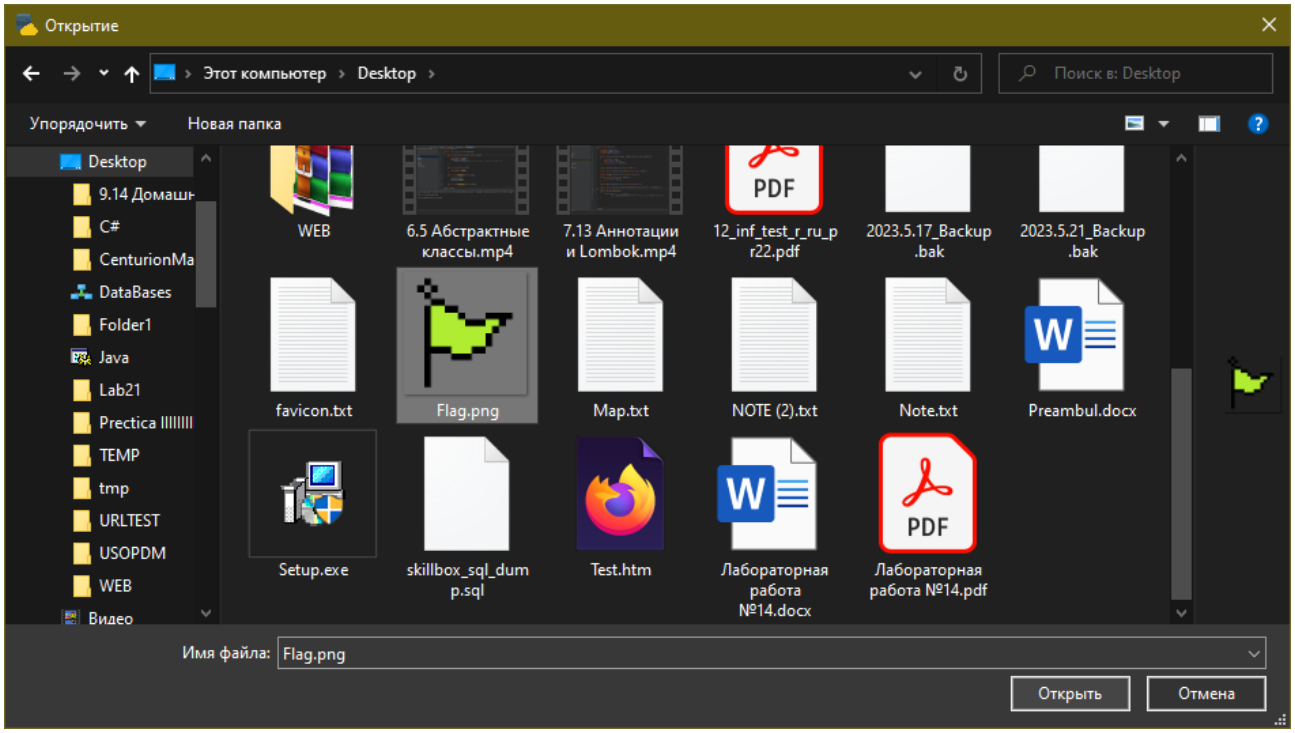
```
private void backupDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DateTime now = DateTime.Now;
            String backupName = now.Year.ToString() + "." + now.Month.ToString() + "." + now.Day.ToString() +
"_Backup.bak";
            saveFileDialog1.FileName= backupName;
            saveFileDialog1.DefaultExt = ".bak";

            if(saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(saveFileDialog1.FileName, ".+\\.bak")){

                    //MessageBox.Show(saveFileDialog1.FileName);


                    try
                    {
                        using (SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True"))
                        {
                            connection.Open();

                            String query = "backup database Computer_magazine to disk = @Path";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.Add("@Path", saveFileDialog1.FileName);

                            command.ExecuteNonQuery();
                        }

                        MessageBox.Show("Successfully saved!", "Success!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    }
```

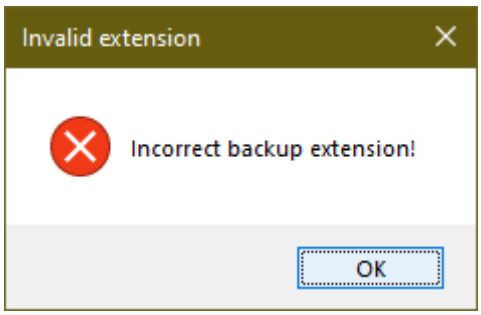
```
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("Incorrect backup extension!", "Invalid extension", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
```

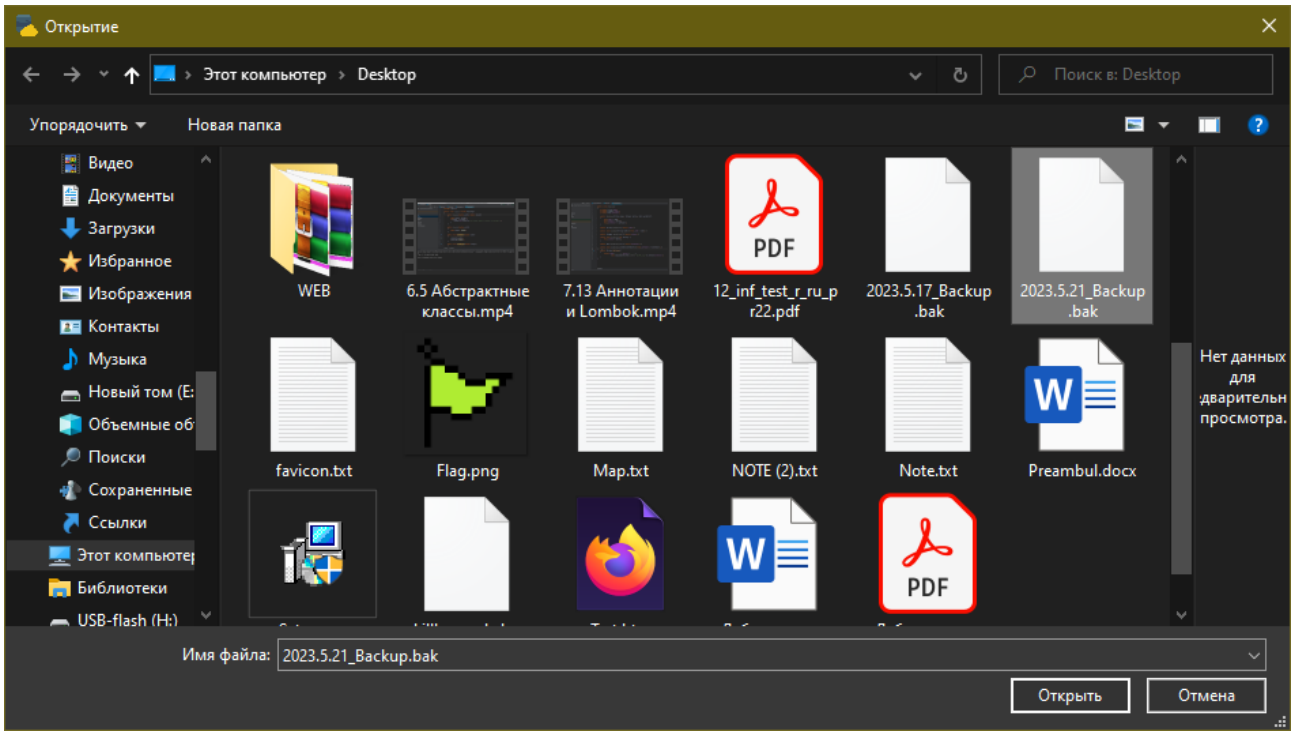
Восстановление данных

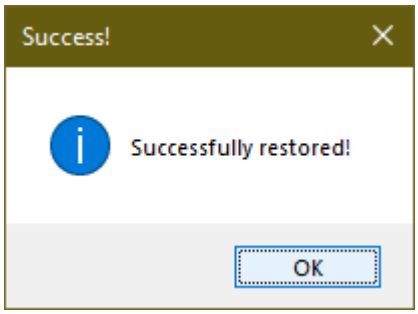
Восстановление БД доступно только из файла формата .bak!





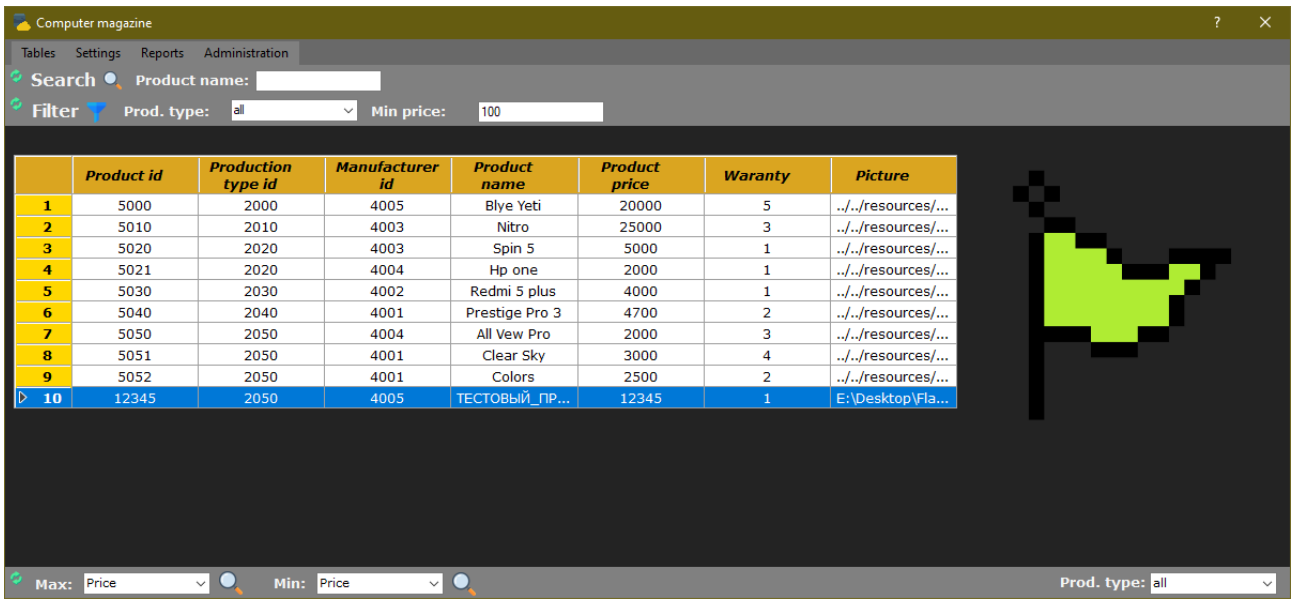




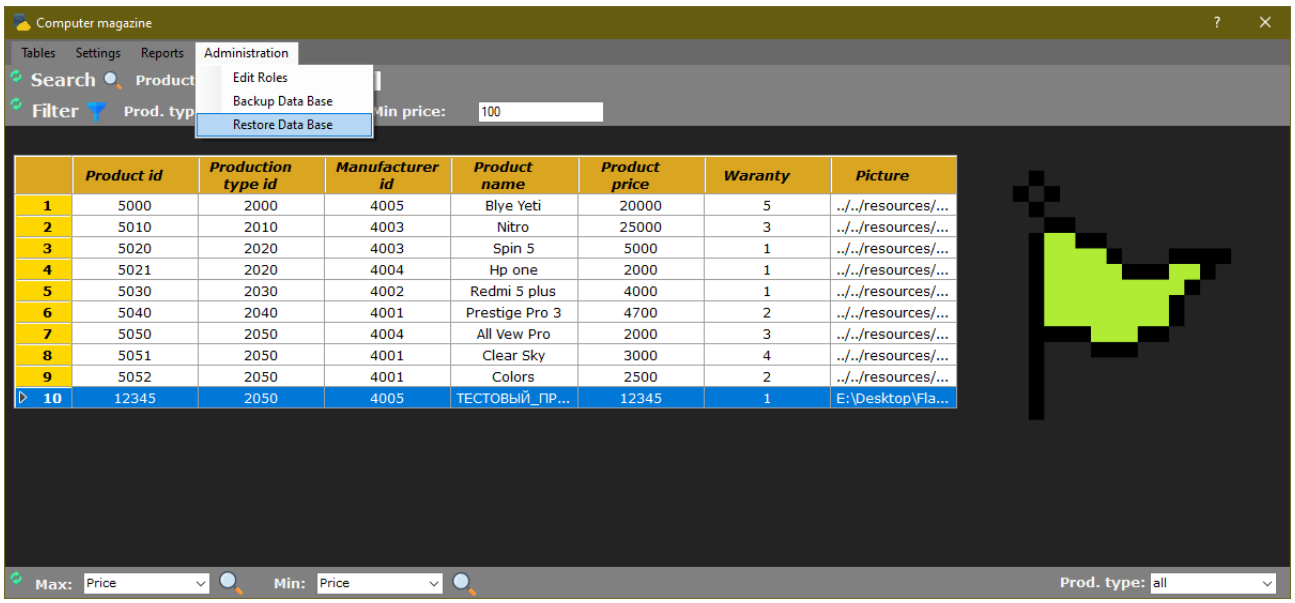


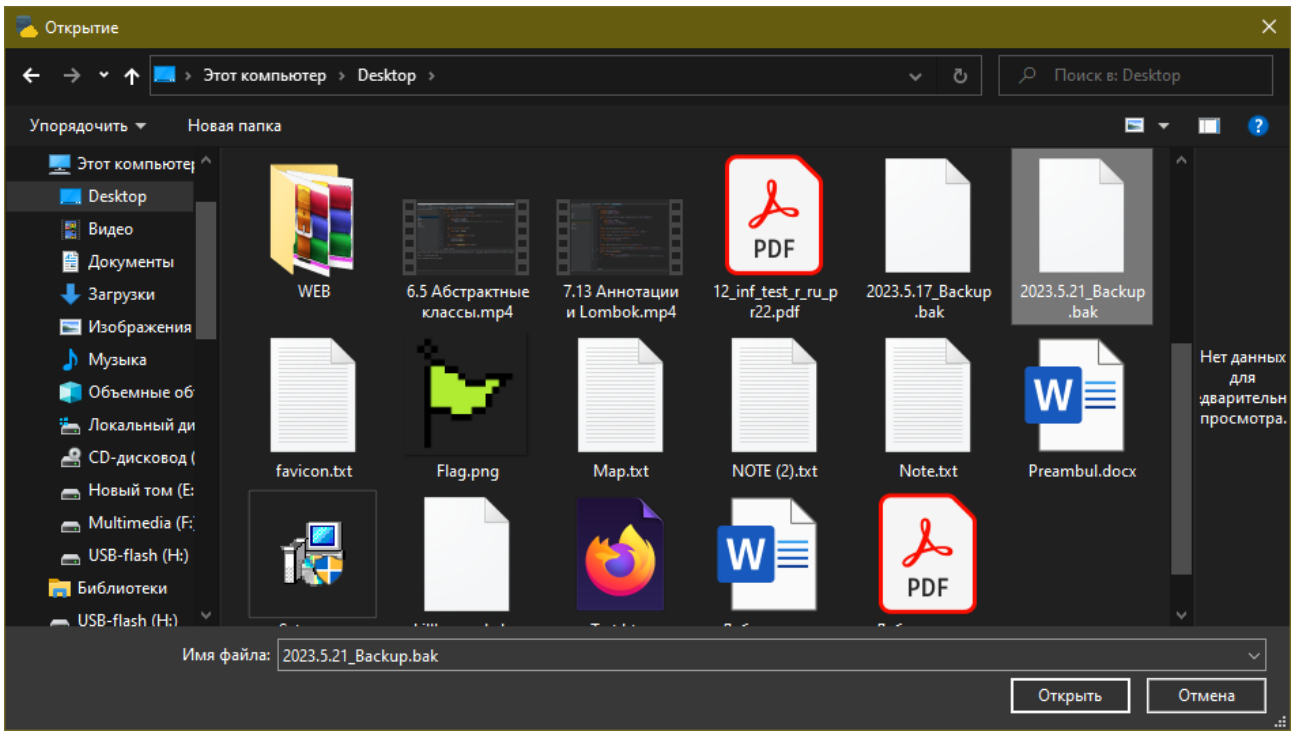
Доказательство восстановления!

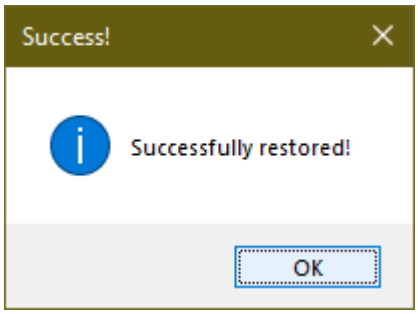
До восстановления:

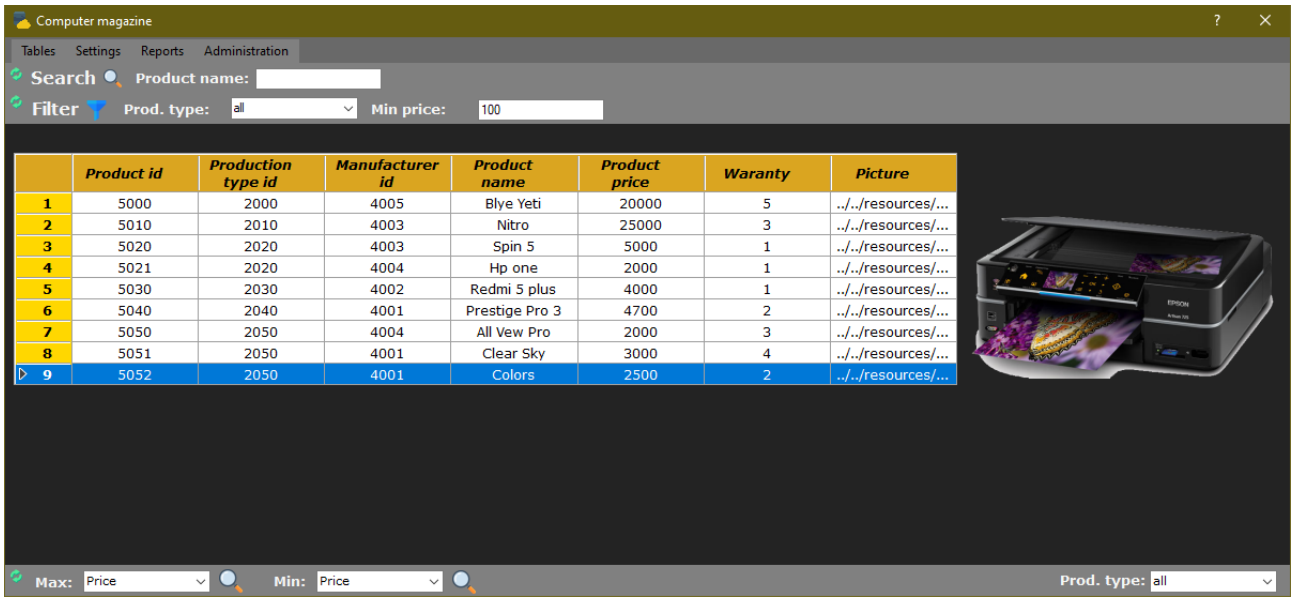


После восстановления:









	Product id	Production type id	Manufacturer id	Product name	Product price	Waranty	Picture
1	5000	2000	4005	Blye Yeti	20000	5	../../resources/...
2	5010	2010	4003	Nitro	25000	3	../../resources/...
3	5020	2020	4003	Spin 5	5000	1	../../resources/...
4	5021	2020	4004	Hp one	2000	1	../../resources/...
5	5030	2030	4002	Redmi 5 plus	4000	1	../../resources/...
6	5040	2040	4001	Prestige Pro 3	4700	2	../../resources/...
7	5050	2050	4004	All Vew Pro	2000	3	../../resources/...
8	5051	2050	4001	Clear Sky	3000	4	../../resources/...
9	5052	2050	4001	Colors	2500	2	../../resources/...

Код:

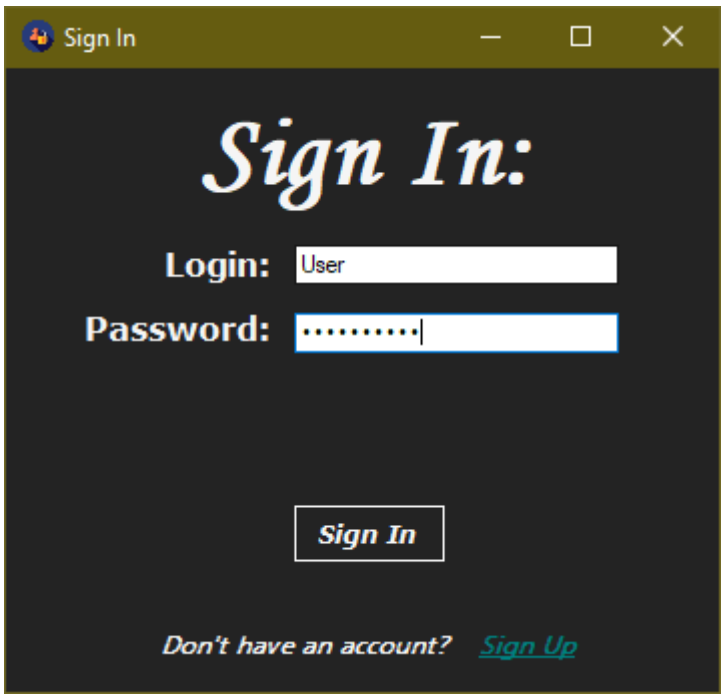
```
private void restoreDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(openFileDialog1.FileName, ".+\\.bak"))
```

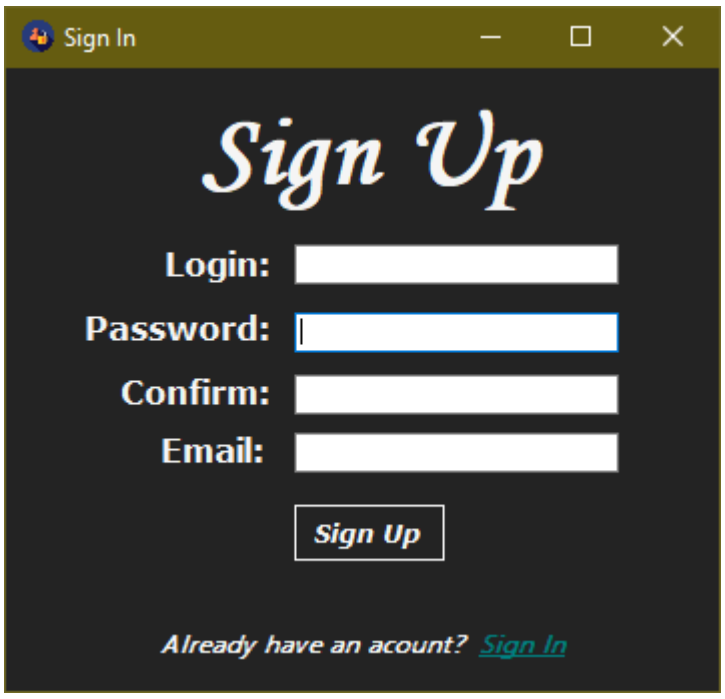
```
            {
                try
                {
                    using (SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True"))
                    {
                        connection.Open();

                        String query = "use master";
                        SqlCommand command = new SqlCommand(query, connection);
                        command.ExecuteNonQuery();

                        command.CommandText = "restore database Computer_magazine from disk = @Path";
                        command.Parameters.Add("@Path", openFileDialog1.FileName);
                        command.ExecuteNonQuery();

                        command.CommandText = "use Computer_magazine";
                        command.ExecuteNonQuery();

                        MessageBox.Show("Successfully restored!", "Success!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    }
                } catch(Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("Incorrect backup extension!", "Invalid extension", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
```

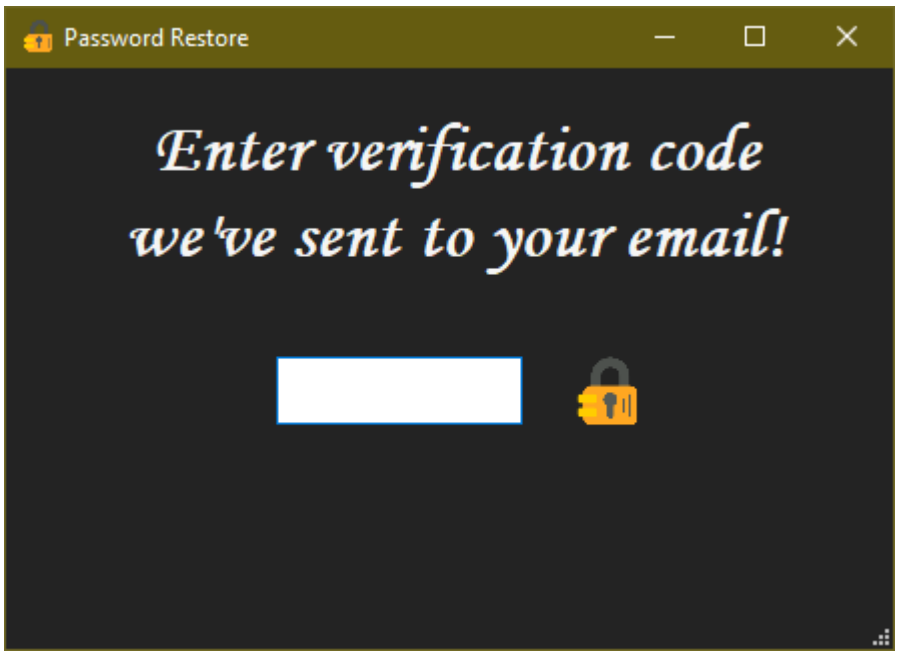
Безопасность данных

Для обеспечения безопасности БД мною были предусмотрены следующие пункты

Авторизация







Транзакции

```
--Transactions--

--automated transactions--

/*
update Pasport_data
set Pay_code = 102
where Id = 1005
go


select *
from Pasport_data
go


create table supplier(Sup_code int primary key,
                                        Sup_name varchar(20),
                                        Company varchar(20),
                                        Sup_Adres Adres)
go


alter table supplier
add Telephone TNumber
go


insert into supplier(Sup_code, Sup_name, Man_cod, Sup_Adres, Telephone)
values  (3001, 'Comp_Moldova', 4003, 'ул. Штефан чел маре 21/5', '+37367354672'),
                (3002, 'Hyper Supply', 4005, 'ул. Гоголя 34/4', '+37369984519'),
                (3003, 'GMSupply', 4006, 'ул. Колумна 5','+37378534765'),
                (3004, 'TechMD', 4002, 'ул. Узинелор 7','+37379945543'),
                (3005, 'XMoldova', 4001, 'ул. Петру Мовилэ 7','+37360606607'),
                (3006, 'Hyper Supply', 4004, 'ул. Михаил Садовяну 3','+37369984519')
go
*/




--nonautomated transactions--
```

```
--implicit transactions--

set implicit_transactions on
go


--Deleting data about ClientS and after that rollback--

--transaction begin--
select *
from Pasport_data;

delete from Pasport_data;

select *
from Pasport_data;

rollback transaction
go
--transaction end--

select *
from Pasport_data
go


set implicit_transactions off
go



--Changing product's (Blue Yeti's) manufacturer (from Hyper PC to LG) (from manufacturer_pruduction_price) =>
changing product's manufacturer in consignment journal (consignment)!--

set implicit_transactions on
go


declare @ProductName varchar(20), @ProductCode int;

set @ProductName = 'Blue Yeti';
set @ProductCode = (select cod_prod
                                        from Product
                                        where Prod_name = @ProductName);


declare @NewManufacturerName varchar(20), @NewManufacturerCode int;

set @NewManufacturerName = 'LG';
set @NewManufacturerCode = (select Man_code
                                                        from manufacturer
                                                        where Man_name = @NewManufacturerName);

--transaction begin--

update manufacturer_pruduction_price
set Man_code = @NewManufacturerCode
where cod_prod = @ProductCode;

update consignment
set Man_code = @NewManufacturerCode
where cod_prod = @ProductCode
go

commit transaction;
--transaction end--

set implicit_transactions off
go



select *
from manufacturer_pruduction_price
go
```

```
select *
from Product
go

select *
from manufacturer
go

select *
from consignment
go




--Explicit + distributed transactions--

--withdraw money from a banking card for user Болякин Андрей Никифорович (Boliakin Andrei Nikiforovici)--
--client searching will be released by IDNP searching--

select *
from Pasport_data
go

select *
from Payment
go



select *
from Banking.dbo.Credit_card
go

begin distributed transaction
declare @ClientCode int, @IDNP char(13), @TotalToPay smallint;

set @ClientCode = (select Id
                                    from Pasport_data
                                    where C_Name = 'Андрей' and C_Surname = 'Болякин' and C_FatherName =
'Никифорович')

set @IDNP = (select IDNP
                        from Pasport_data
                        where Id = @ClientCode); --client's IDNP in our magazine--

set @TotalToPay = (select Gen_price
                                    from Chek_infoChek
                                    where cod_client = @ClientCode)

set @ClientCode = (select cod_client                    --searching the client by his IDNP
                                    from Banking.dbo.Client
                                    where IDNP = @IDNP);

print 'Total to pay: ' + cast(@TotalToPay as varchar(20));

if(@TotalToPay <= (select money_amount from Banking.dbo.Credit_card where cod_client = @ClientCode))
begin
        update Banking.dbo.Credit_card
        set money_amount = money_amount - @TotalToPay
        where cod_client = @ClientCode;

        print 'Success!';
        print 'Transaction is successfull!'
        commit transaction;
end
else
begin
        print 'Ooops!';
        print 'You do not have enough money!';
        rollback transaction;
end;
go
```

```
select *
from Banking.dbo.Credit_card
go

select *
from Banking.dbo.Client
go



--Creating a credit contract for client Безухов Пьер Анатольевичь--


update Chek
set Pay_code = 101
where Check_code = 1014
go


begin distributed transaction
declare @ContractNumber int, @BankCode int, @OpeningDate date, @Term int, @CreditSize money, @ClientCode int,
@ClientFIO varchar(70);

set @ClientFIO = 'Безухов Пьер Анатольевичь';

set @ContractNumber = (select max(contract_number)
                                        from Banking.dbo.Credit_contract) + 1;

set @BankCode = 101; --MAIP, for example--

set @OpeningDate = getDate();

set @Term = 24; --2 years, for example--


--in the shop--
set @ClientCode = (select Id
                                    from Pasport_data
                                    where C_Surname = 'Безухов' and C_Name = 'Пьер' and C_FatherName =
'Анатольевичь')

set @CreditSize = (select Gen_price
                                    from Chek_infoChek
                                    where cod_client = @ClientCode);


--in bank--
set @ClientCode = (select cod_client
                                    from Banking.dbo.Client
                                    where fio = @ClientFIO);

if(@ClientCode is null)
begin
print 'Inserting new client!';
set @ClientCode = (select max(cod_client) from Banking.dbo.Client)+1;

insert into Banking.dbo.Client
values (@ClientCode, @ClientFIO, 'ул. Николае Милеску Спэтарул 21/1', 40, '1234567890123', '+37355555555', 0,
6000, 'BezPierAn');
end


insert into Banking.dbo.Credit_contract(contract_number, cod_bank, cod_client, opening_date, term,
credit_size)
values(@ContractNumber, @BankCode, @ClientCode, @OpeningDate, @Term, @CreditSize);

commit transaction
go



select *
from Banking.dbo.Credit_contract
go


select *
```

```
from Pasport_data
go

select *
from Chek
go

select *
from Chek_infoChek
go

select *
from Payment
go
```

Распределение ролей

В моем проекте предусмотрены 4 роли:

Обычный пользователь (наименьшие права) только просмотр таблиц и настройка внешнего вида (темы) приложения...

Продавец, который помимо просмотра таблиц и изменения внешнего вида приложения также может добавлять новые чеки, а также нового покупателя (паспортные данные).

Менеджер, которому добавились возможности редактировать информацию в чеках и информацию о пользователях, а также добавлять новые записи в жернал поставки и новые продкуты. Также у него появилась возможность создания отчетов.

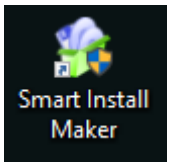
Администратор (полный доступ). Он может редактировать информацию полностью во всех таблицах (добавлять, редактировать и удалять). У него есть возможность печати отчетов а также у него появилась возможность назначения ролей, редактирования логина и пороля пользователя. Может делать резервные копии БД и восстанавливать её...

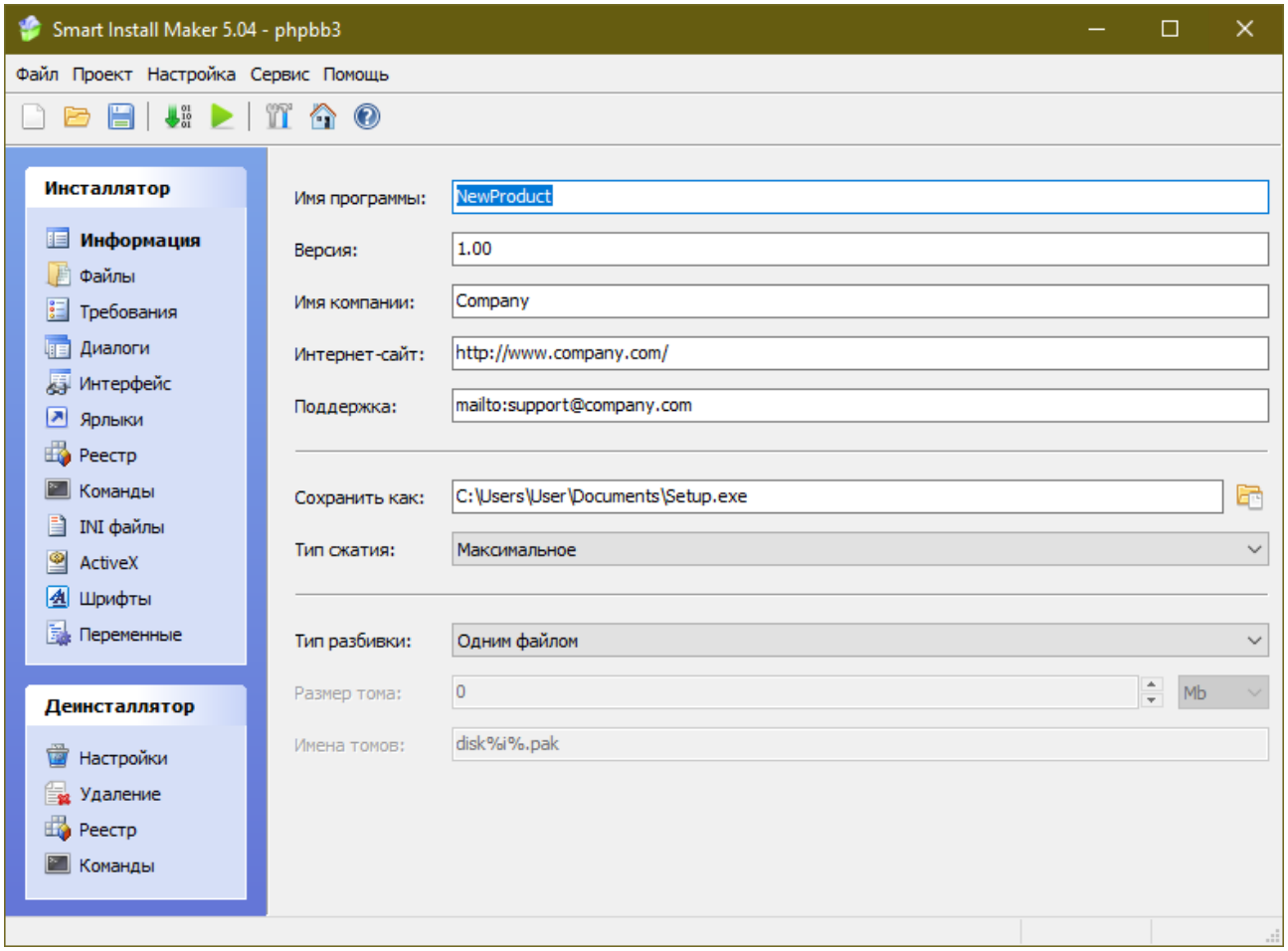
А также ***Параметризированные запросы***, ***проверка корректности введения данных*** и ***невозможность ввода пустых полей***, ***проверка существования (занятости) пользователя (логина) и почты***, ***корректность пароля пользователя (минимум 5 символов длинна и корректность подтверждения пароля при регистрации)***, ***а также возможность восстановления пароля с отправкой пароля на почтовый email пользователя*** …

Создание установочного файла

Для создания установочного файла мною была использована программа **Smart Install Maker.**

Процесс создания установщика:

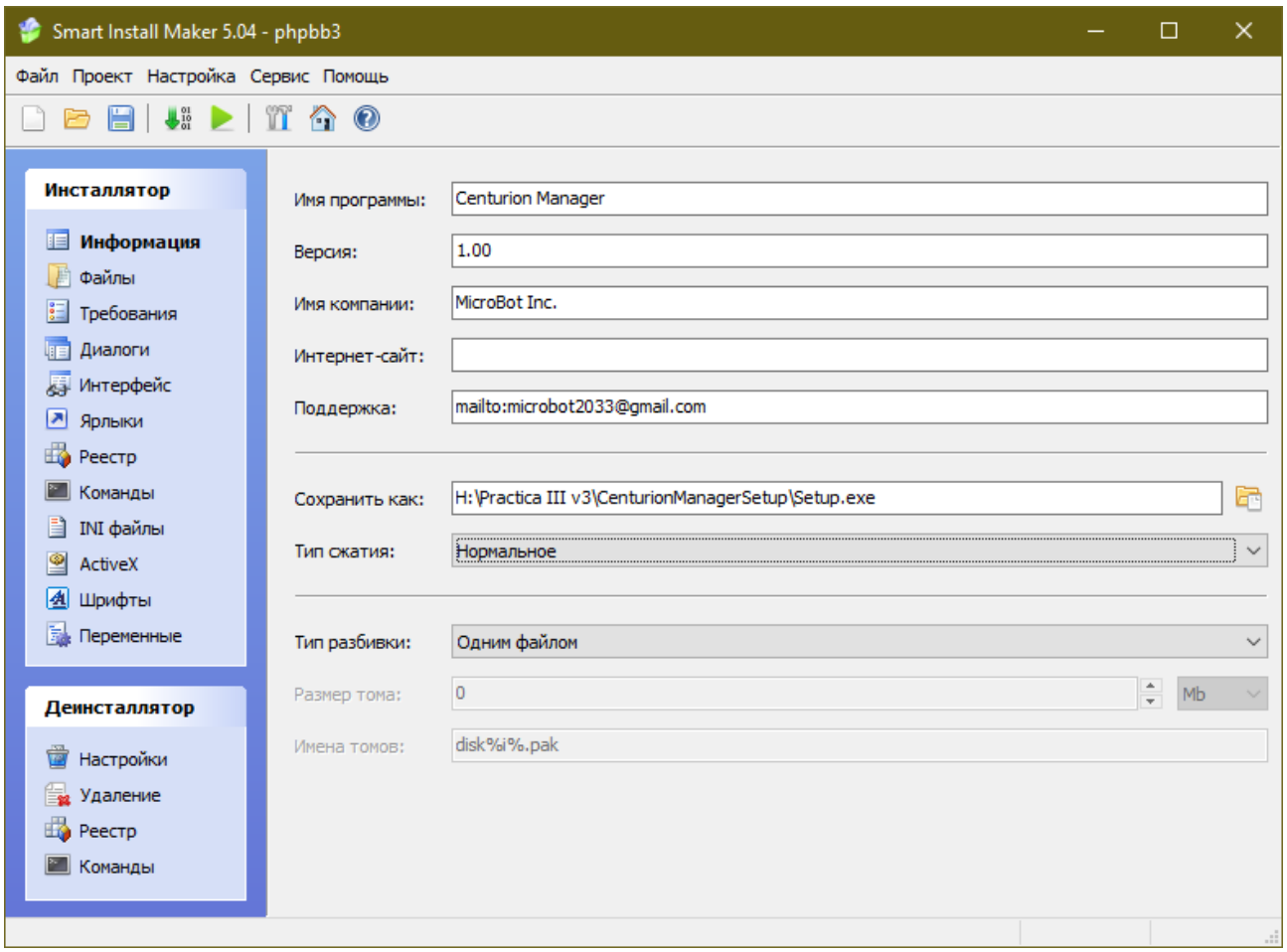


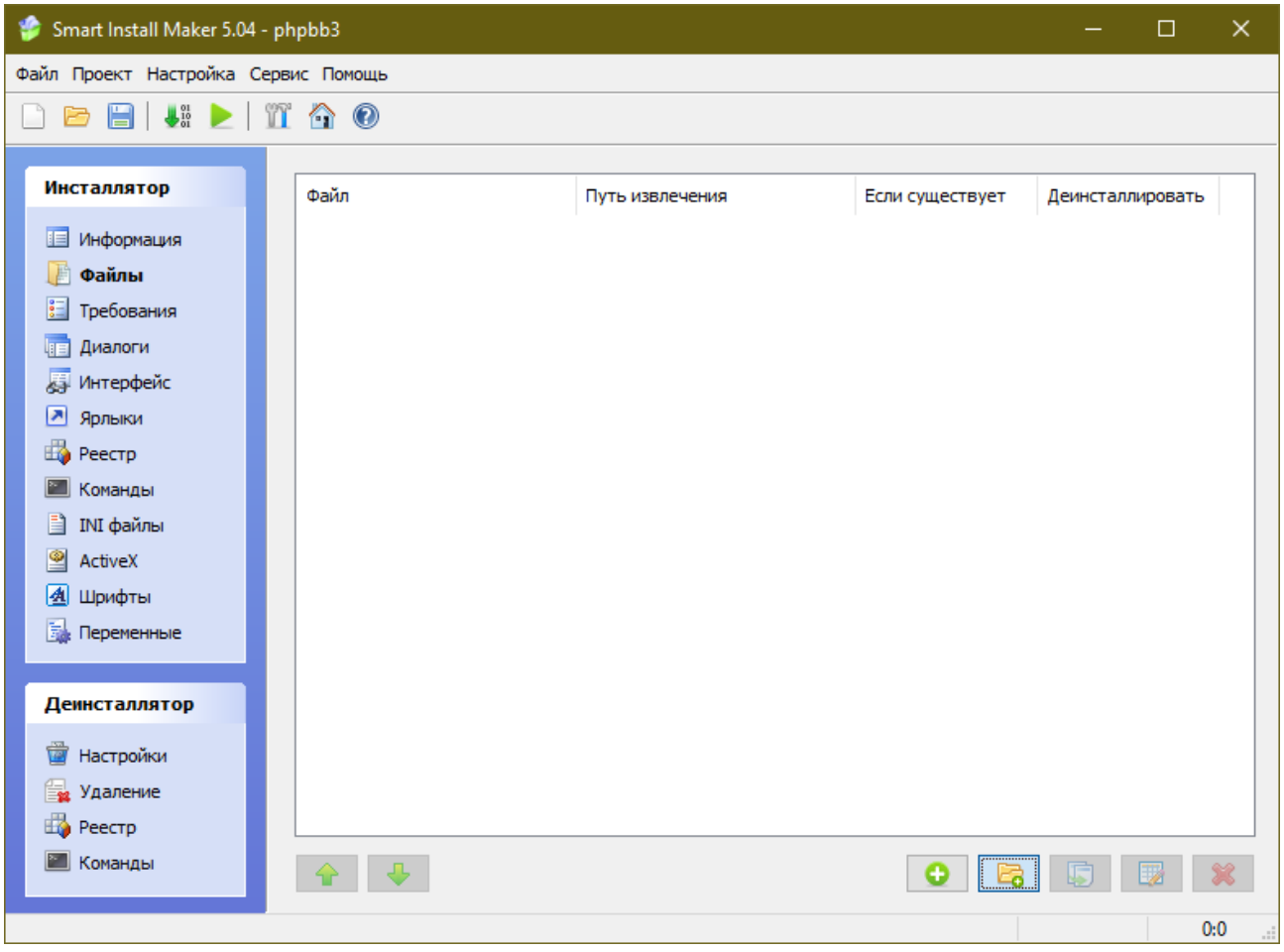


! Тип сжатия. Совет!

Чем сжатие больше, тем инсталлятор весит меньше, но тем дольше устанавливается программа.

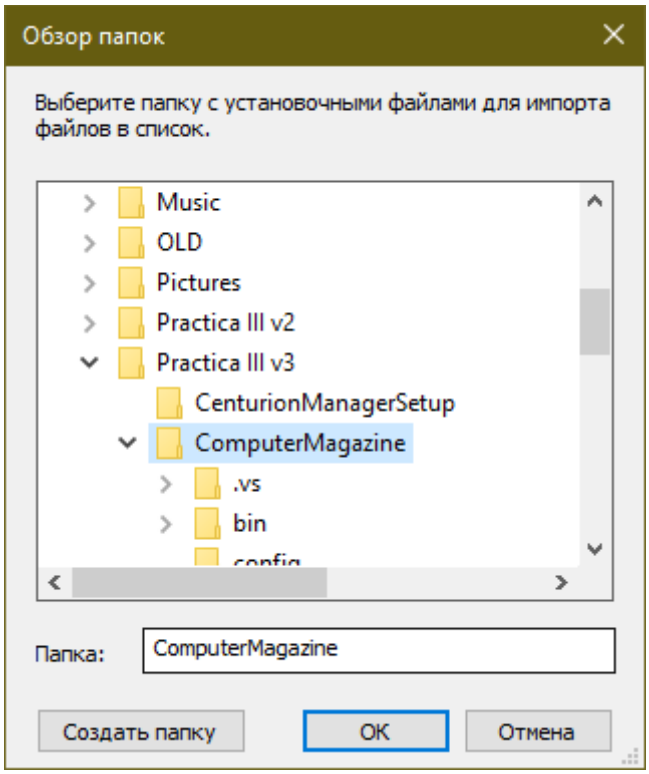
Чем сжатие меньше, тем больше весит инсталлятор, но тем быстрее устанавливается программа…

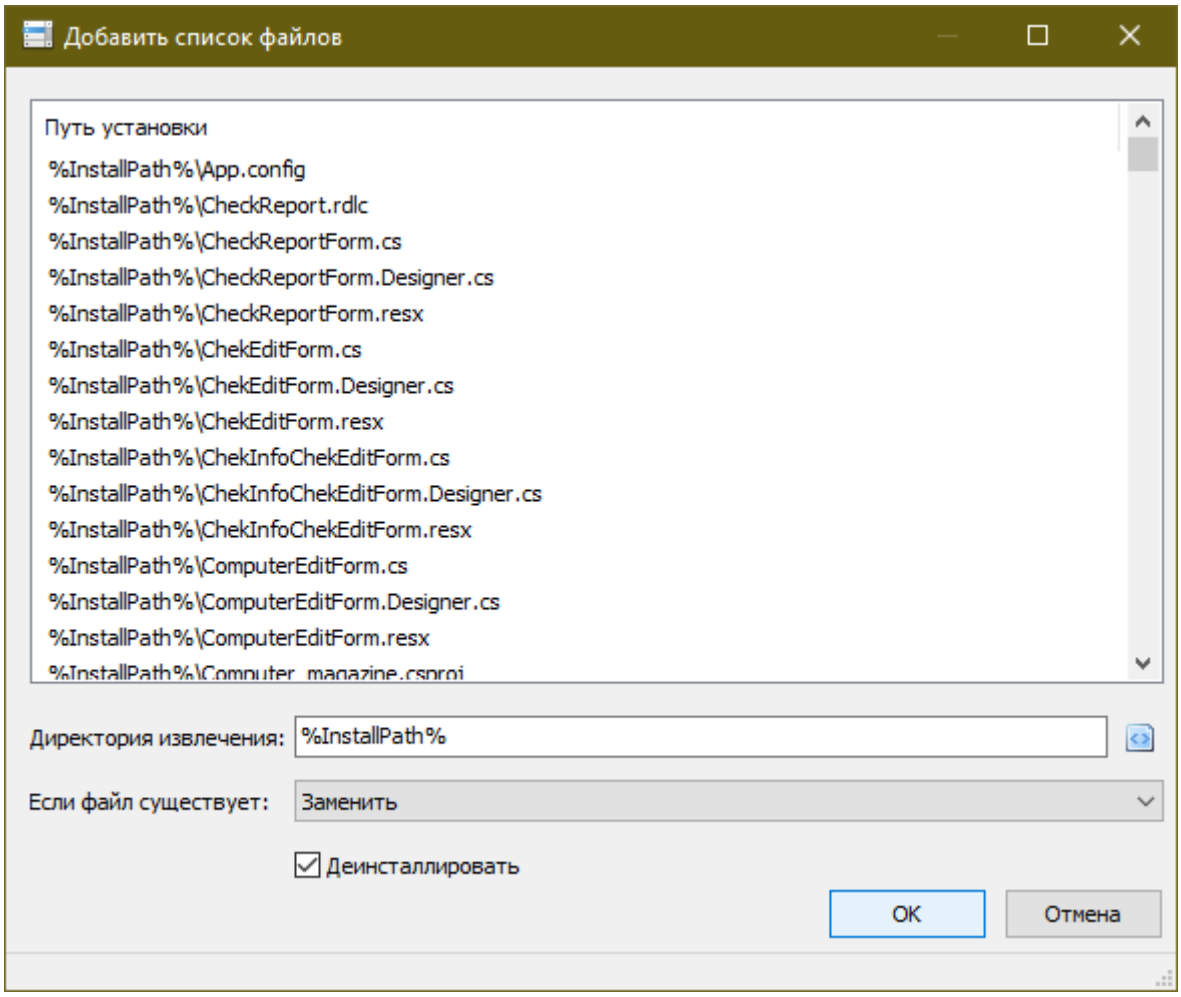

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Имя программы: Centurion Manager
Версия: 1.00
Имя компании: MicroBot Inc.
Интернет-сайт:
Поддержка: mailto:microbot2033@gmail.com
Сохранить как: H:\Practica III v3\CenturionManagerSetup\Setup.exe
Тип сжатия: Нормальное
Тип разбивки: Одним файлом
Размер тома: 0 Mb
Имена томов: disk%i%.pak

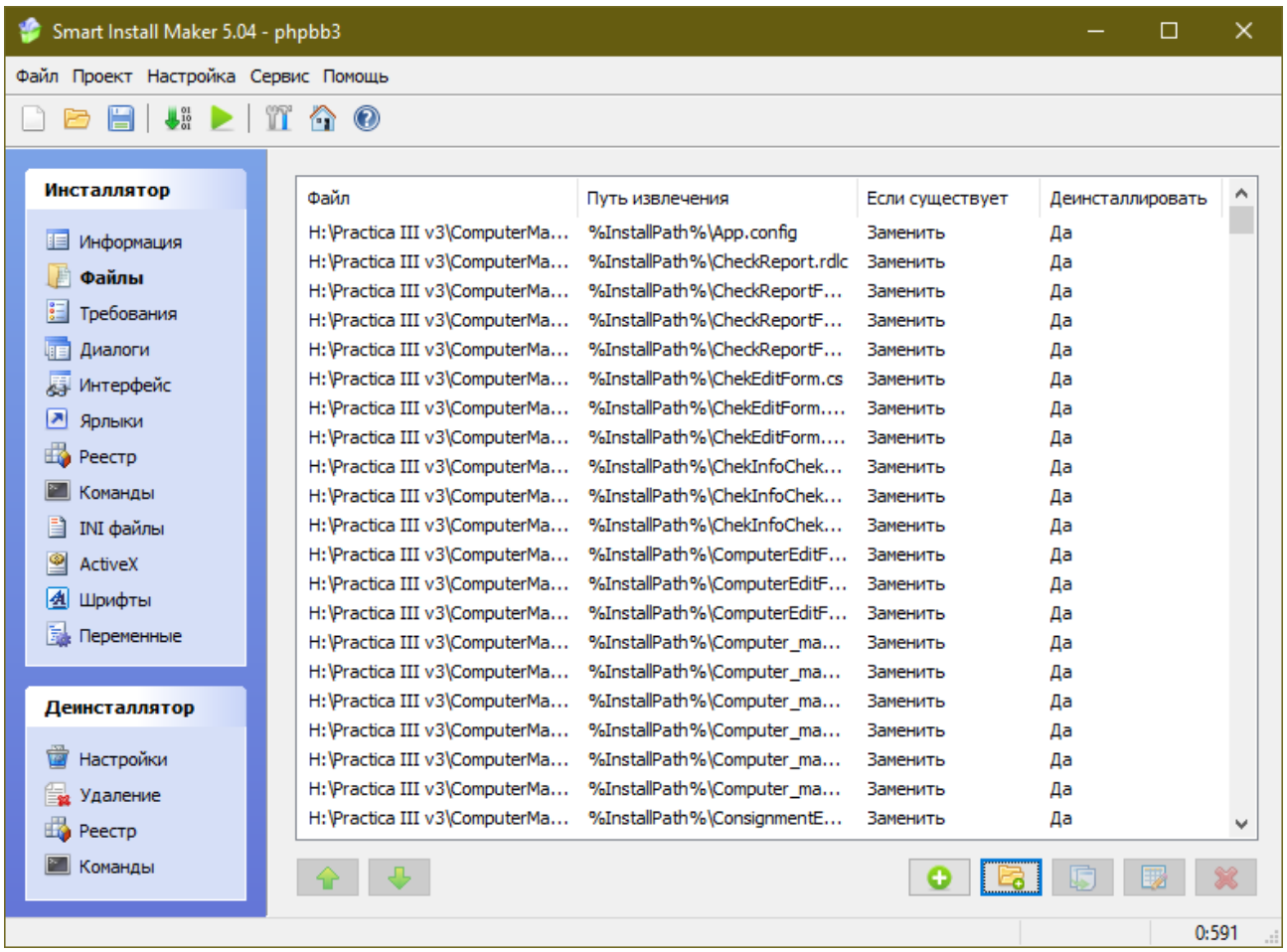

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Файл
Путь извлечения
Если существует
Деинсталлировать
0:0

Нажимаем на значок папочки внизу, чтобы добавить все файлы сразу.

Выбираем папку с приложением

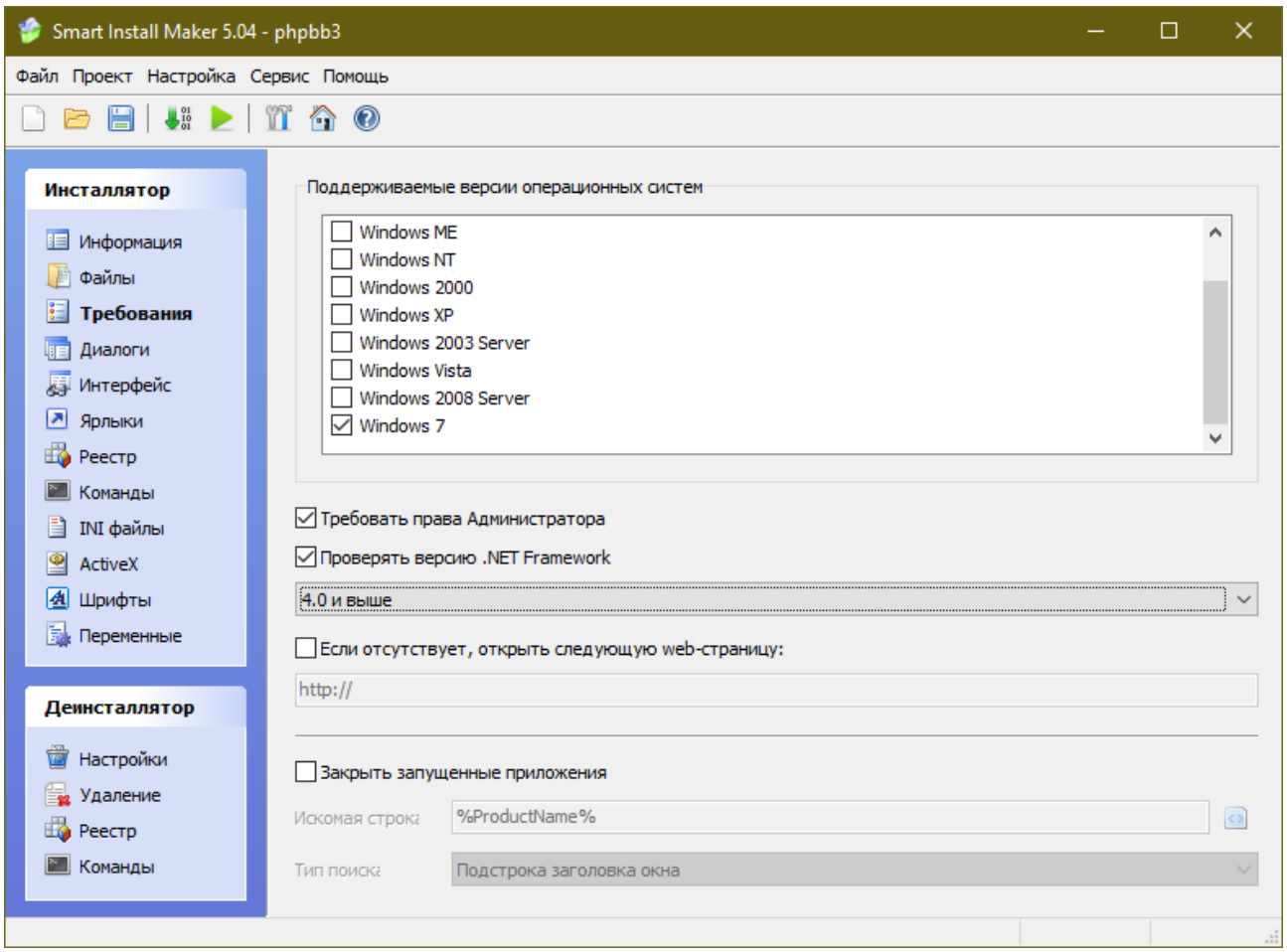






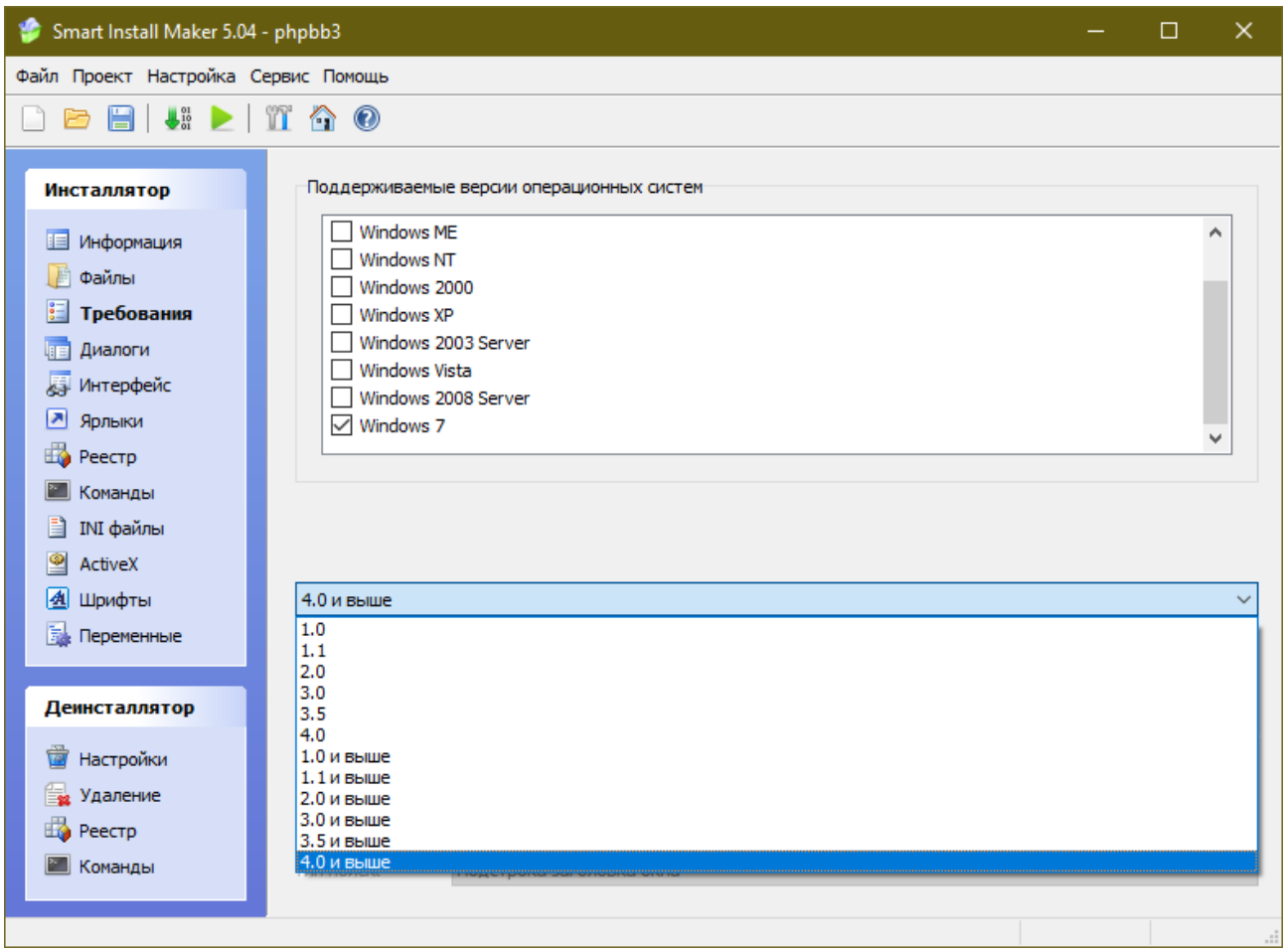
! Примечание!

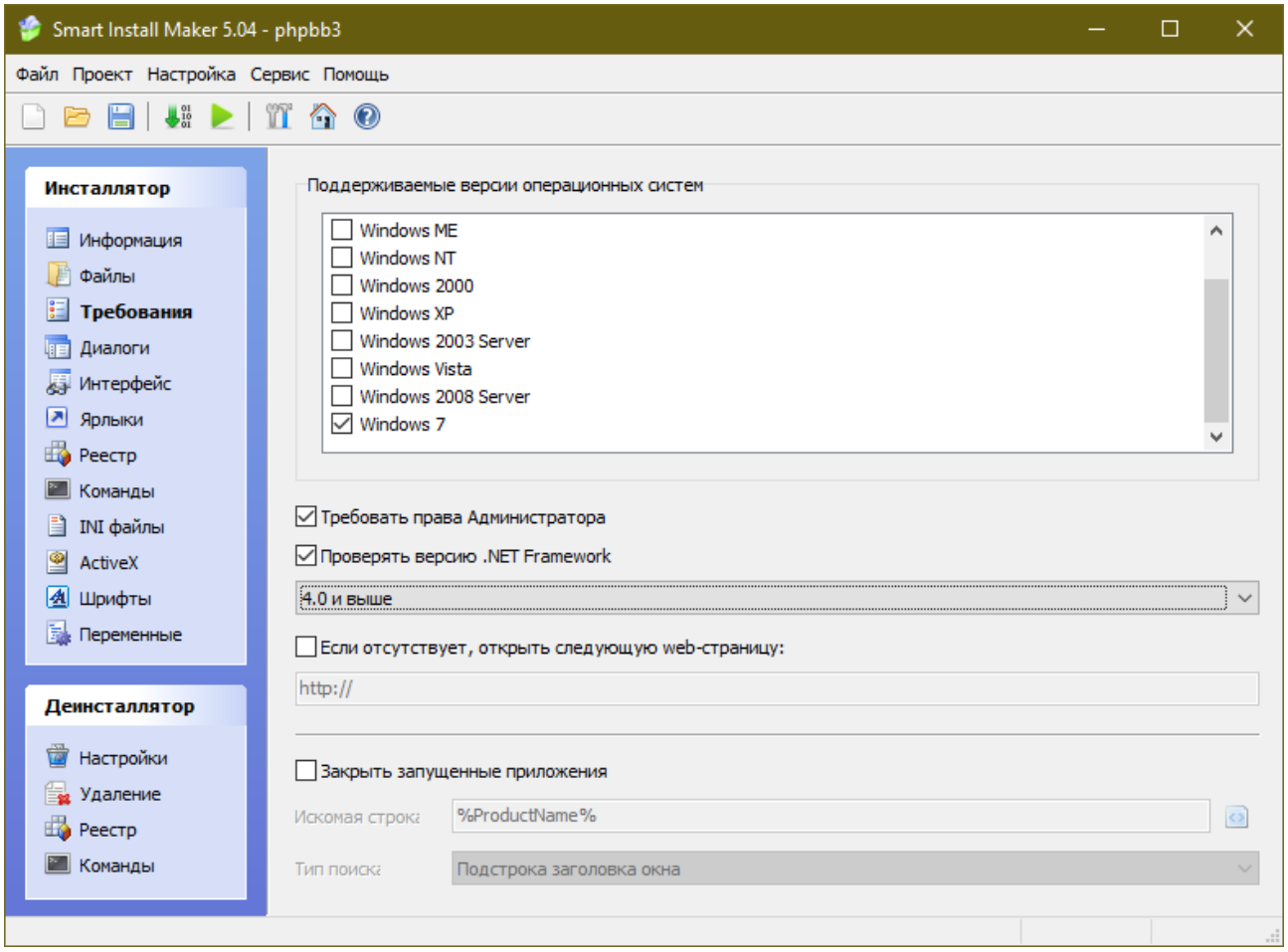
%InstallPath% - это путь до папки приложения, в моём случае это – *H:/Practica III v3/ComputerMagazine*

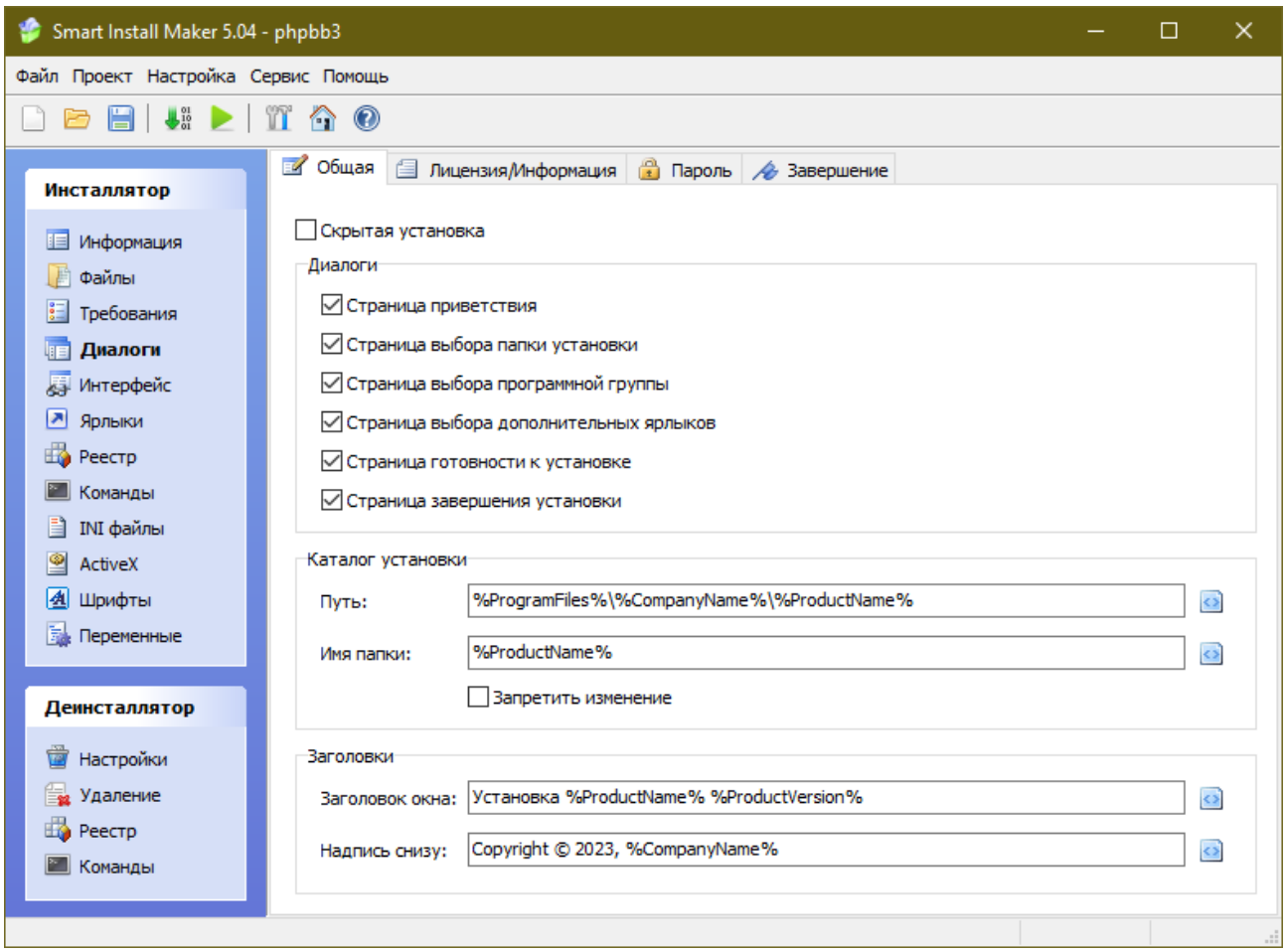


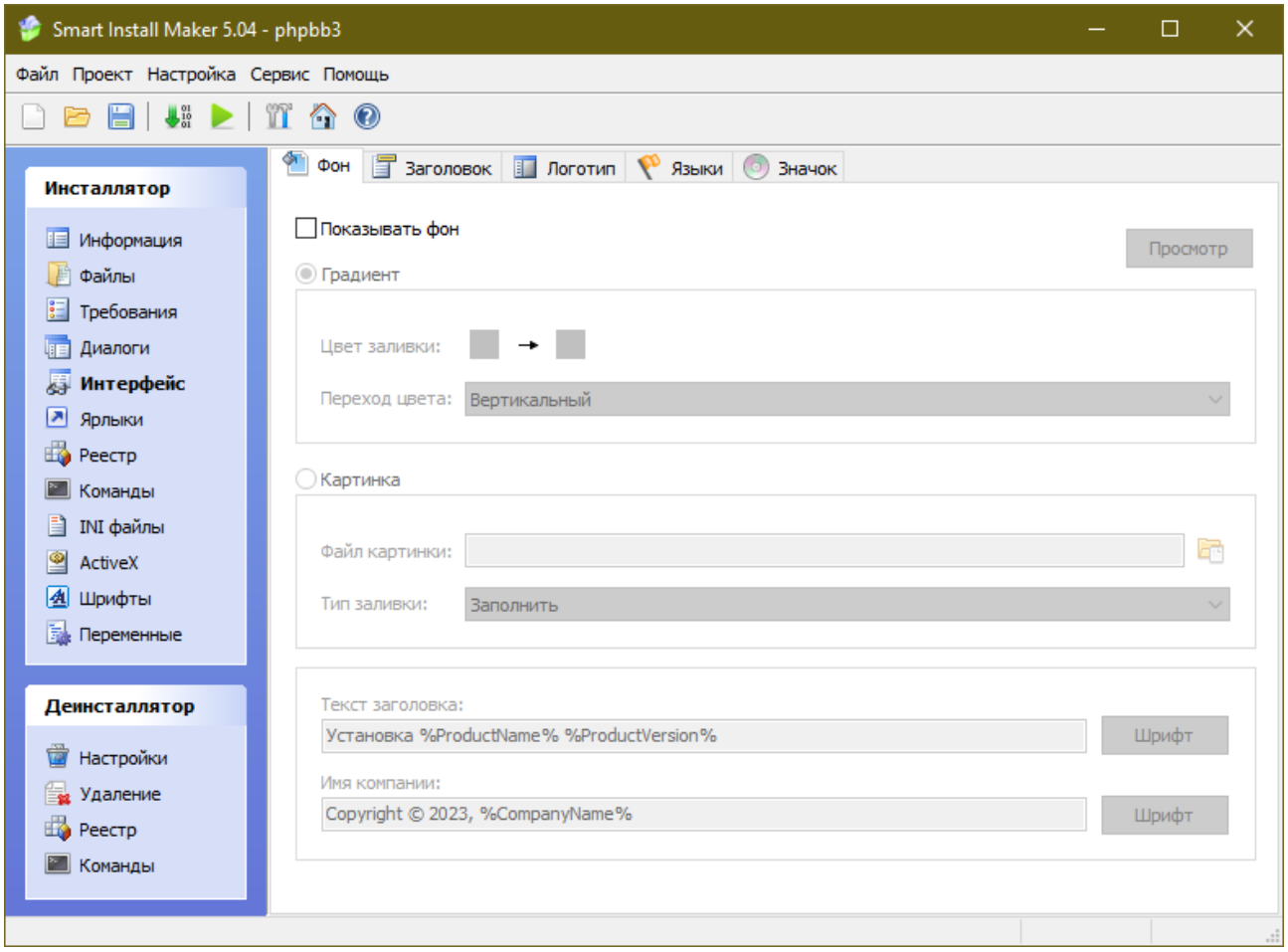
В требованиях ставим Windows 7 (это Win7 b и выше)

А также «Проверять версию .NET»

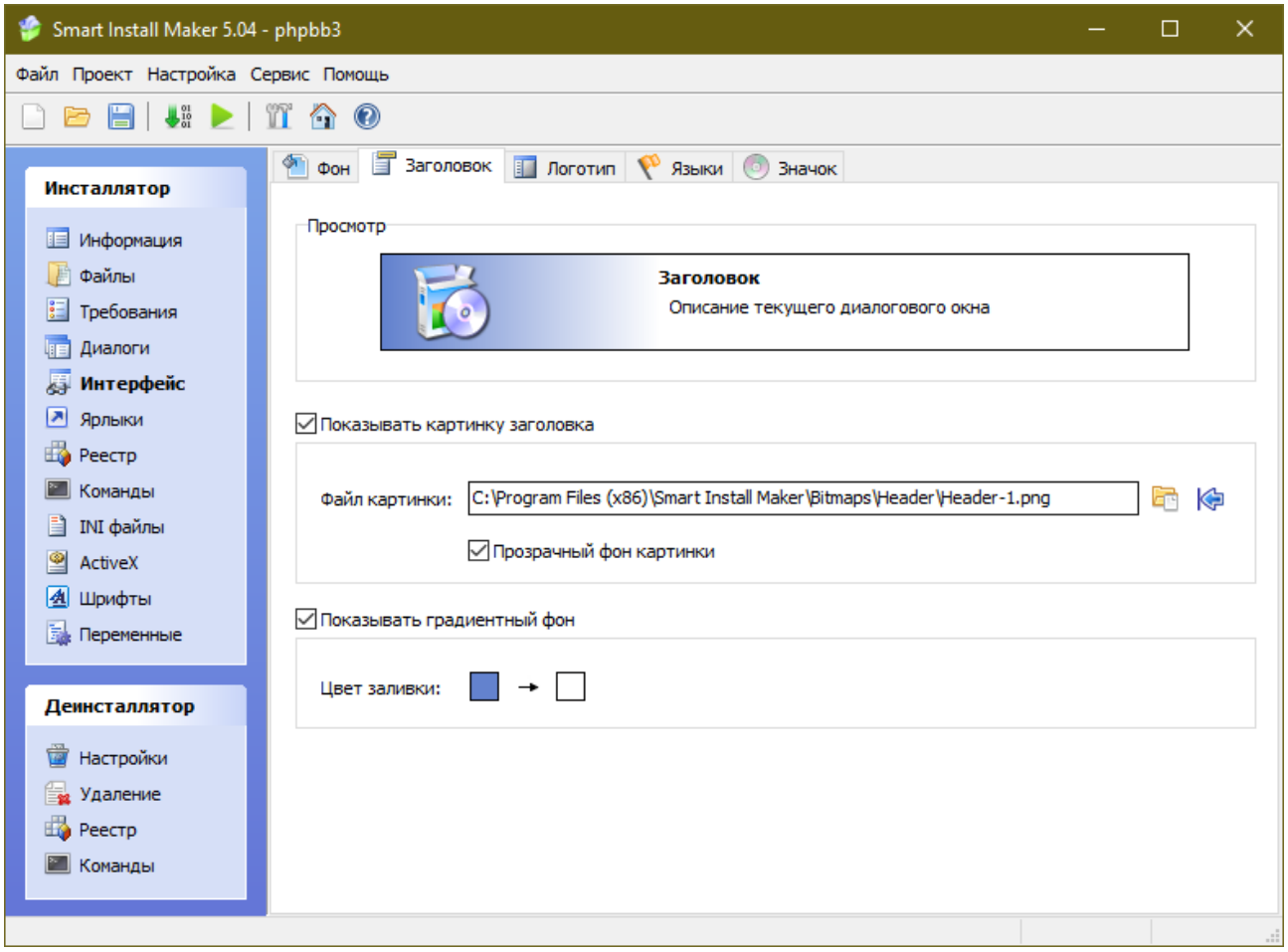

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Поддерживаемые версии операционных систем
Windows ME
Windows NT
Windows 2000
Windows XP
Windows 2003 Server
Windows Vista
Windows 2008 Server
Windows 7
4.0 и выше
1.0
1.1
2.0
3.0
3.5
4.0
1.0 и выше
1.1 и выше
2.0 и выше
3.0 и выше
3.5 и выше
4.0 и выше

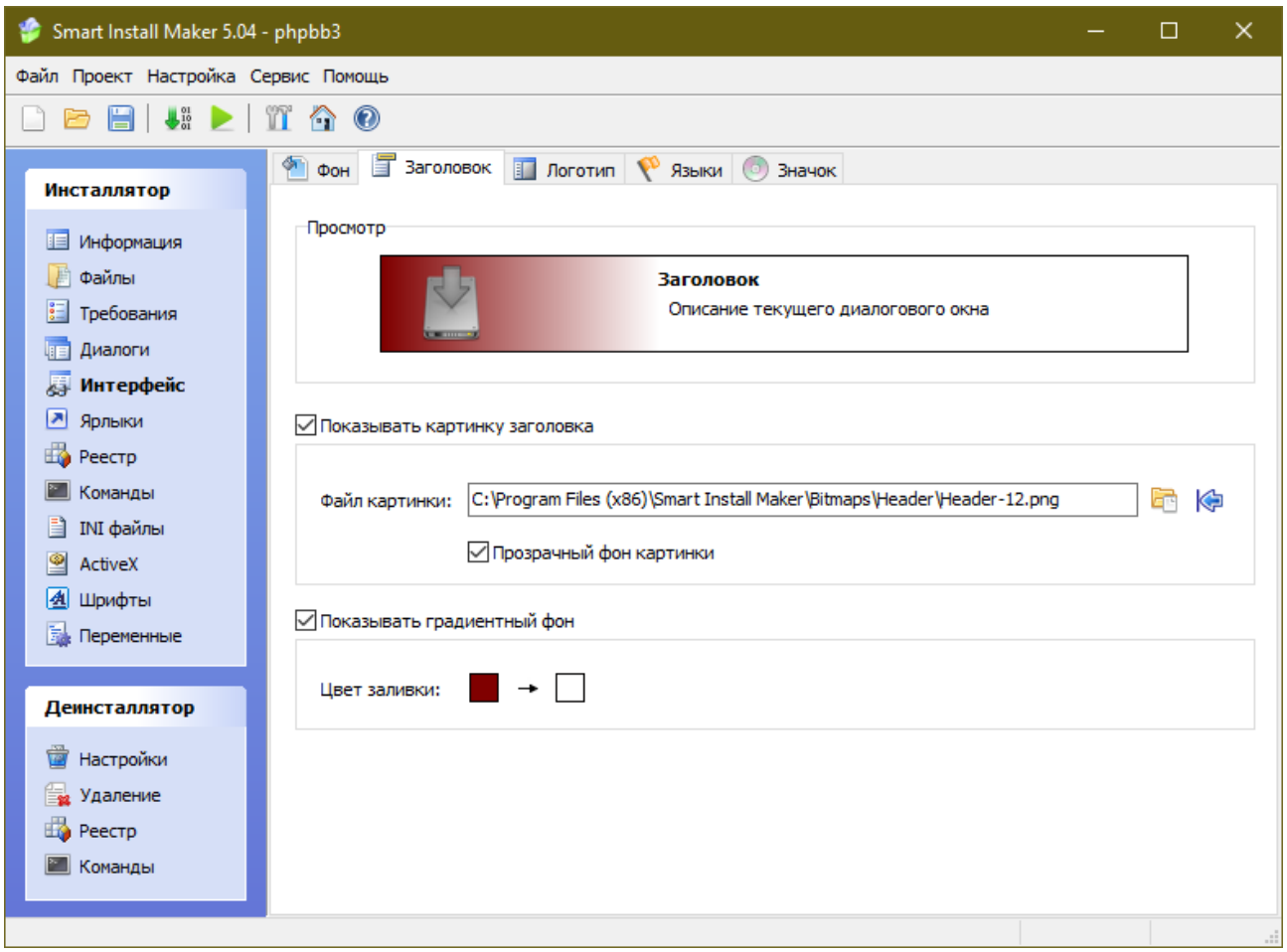

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Поддерживаемые версии операционных систем
Windows ME
Windows NT
Windows 2000
Windows XP
Windows 2003 Server
Windows Vista
Windows 2008 Server
Windows 7
Требовать права Администратора
Проверять версию .NET Framework
4.0 и выше
Если отсутствует, открыть следующую web-страницу:
http://
Закрыть запущенные приложения
Искомая строка
%ProductName%
Тип поиска
Подстрока заголовка окна

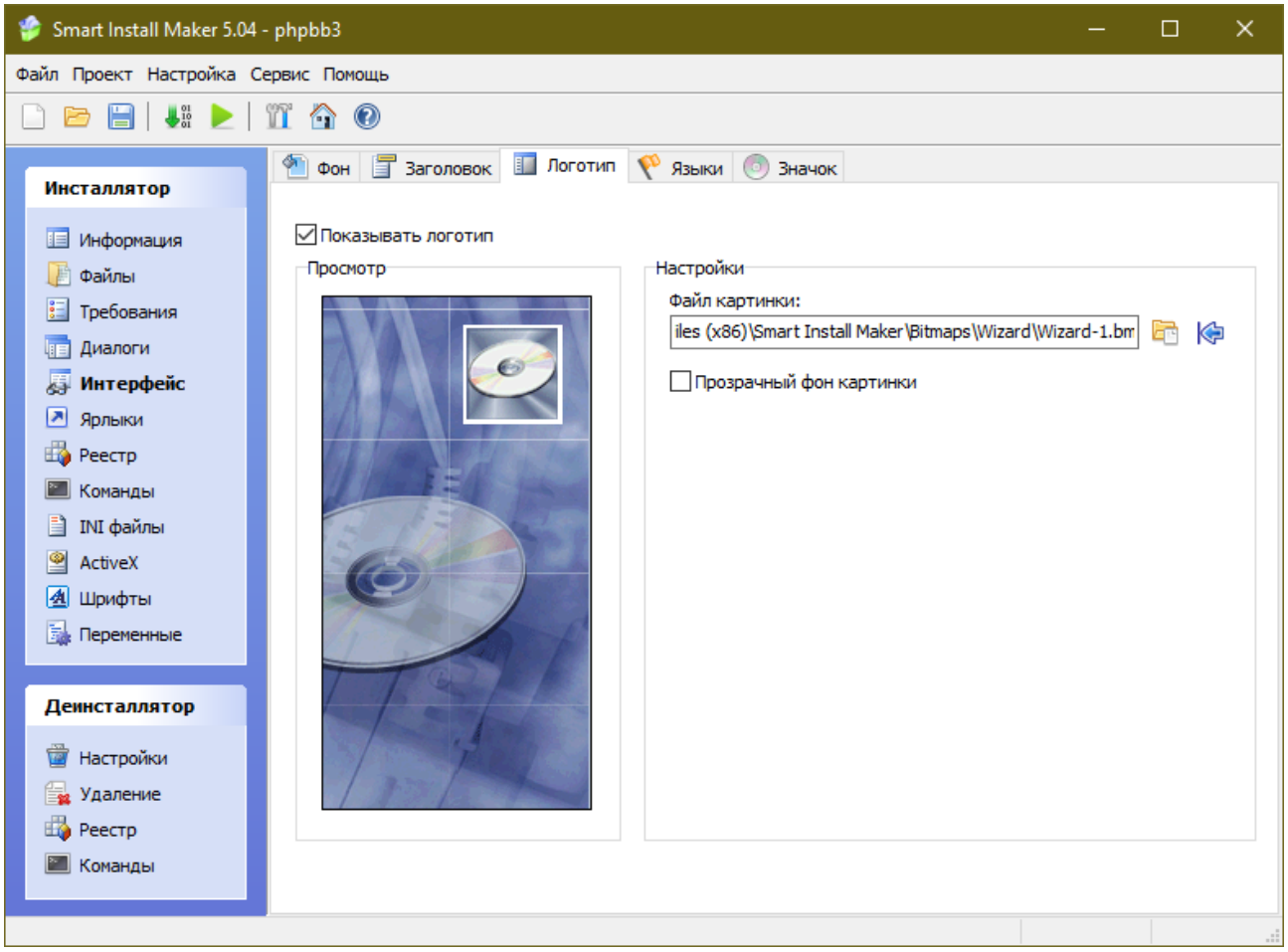

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Общая
Лицензия/Информация
Пароль
Завершение
Скрытая установка
Диалоги
Страница приветствия
Страница выбора папки установки
Страница выбора программной группы
Страница выбора дополнительных ярлыков
Страница готовности к установке
Страница завершения установки
Каталог установки
Путь:
%ProgramFiles%\%CompanyName%\%ProductName%
Имя папки:
%ProductName%
Запретить изменение
Заголовки
Заголовок окна:
Установка %ProductName% %ProductVersion%
Надпись снизу:
Copyright © 2023, %CompanyName%

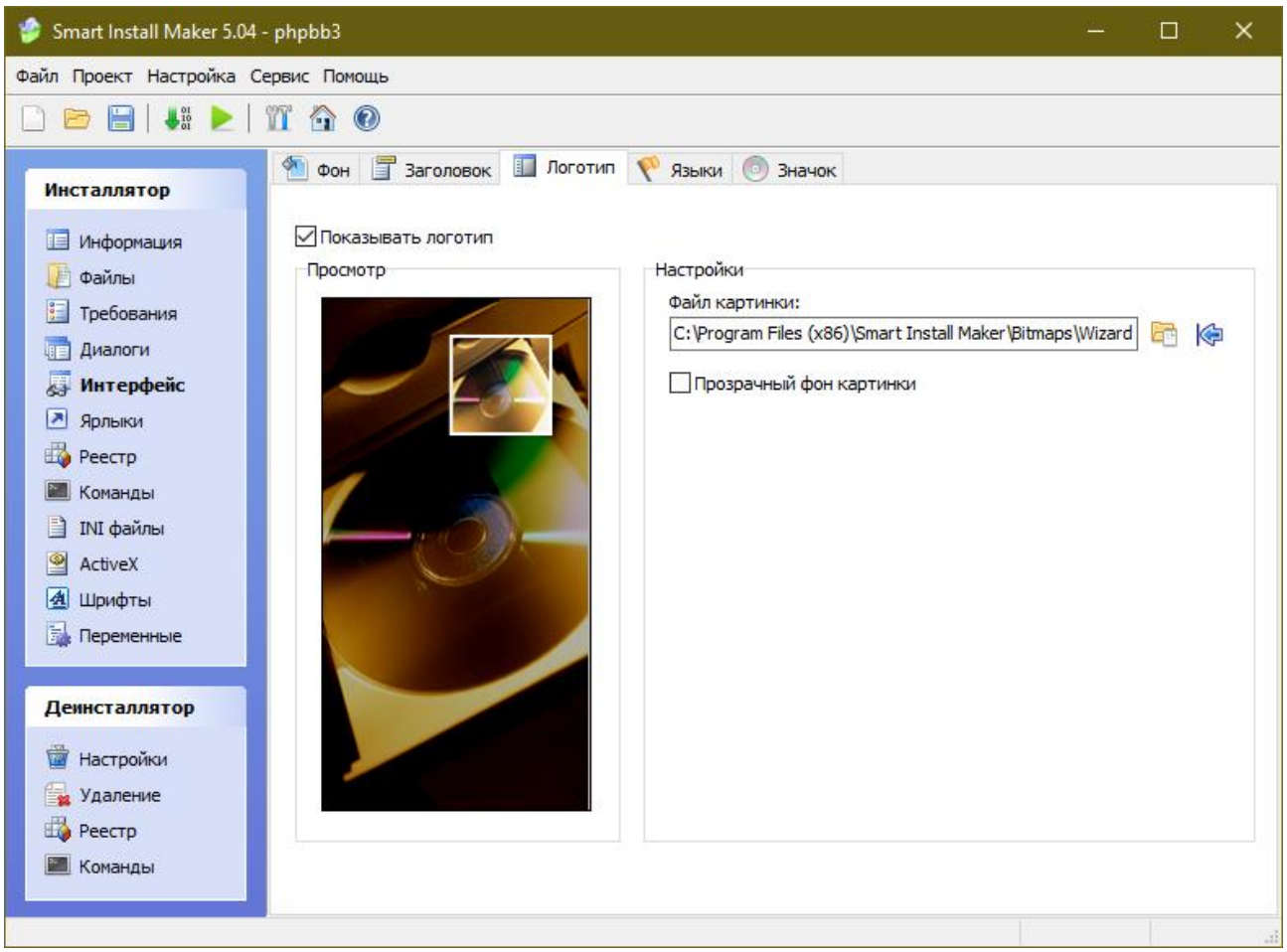

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Показывать фон
Просмотр
Градиент
Цвет заливки:
Переход цвета:
Вертикальный
Картинка
Файл картинки:
Тип заливки:
Заполнить
Текст заголовка:
Установка %ProductName% %ProductVersion%
Шрифт
Имя компании:
Copyright © 2023, %CompanyName%
Шрифт

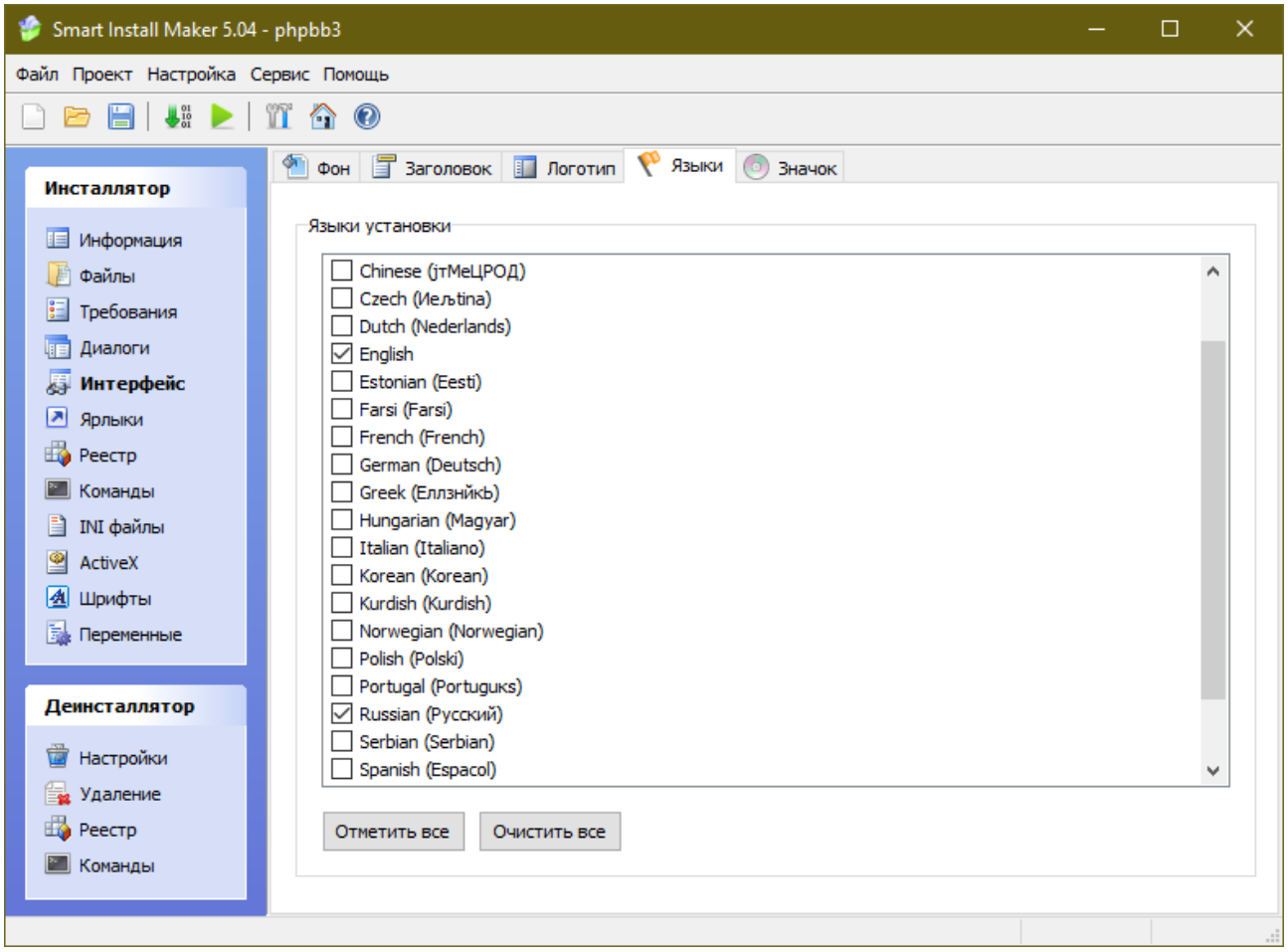
Тут мы можем настроить внешний вид инсталлятора и его иконки (инсталлятора и деинсталлятора)

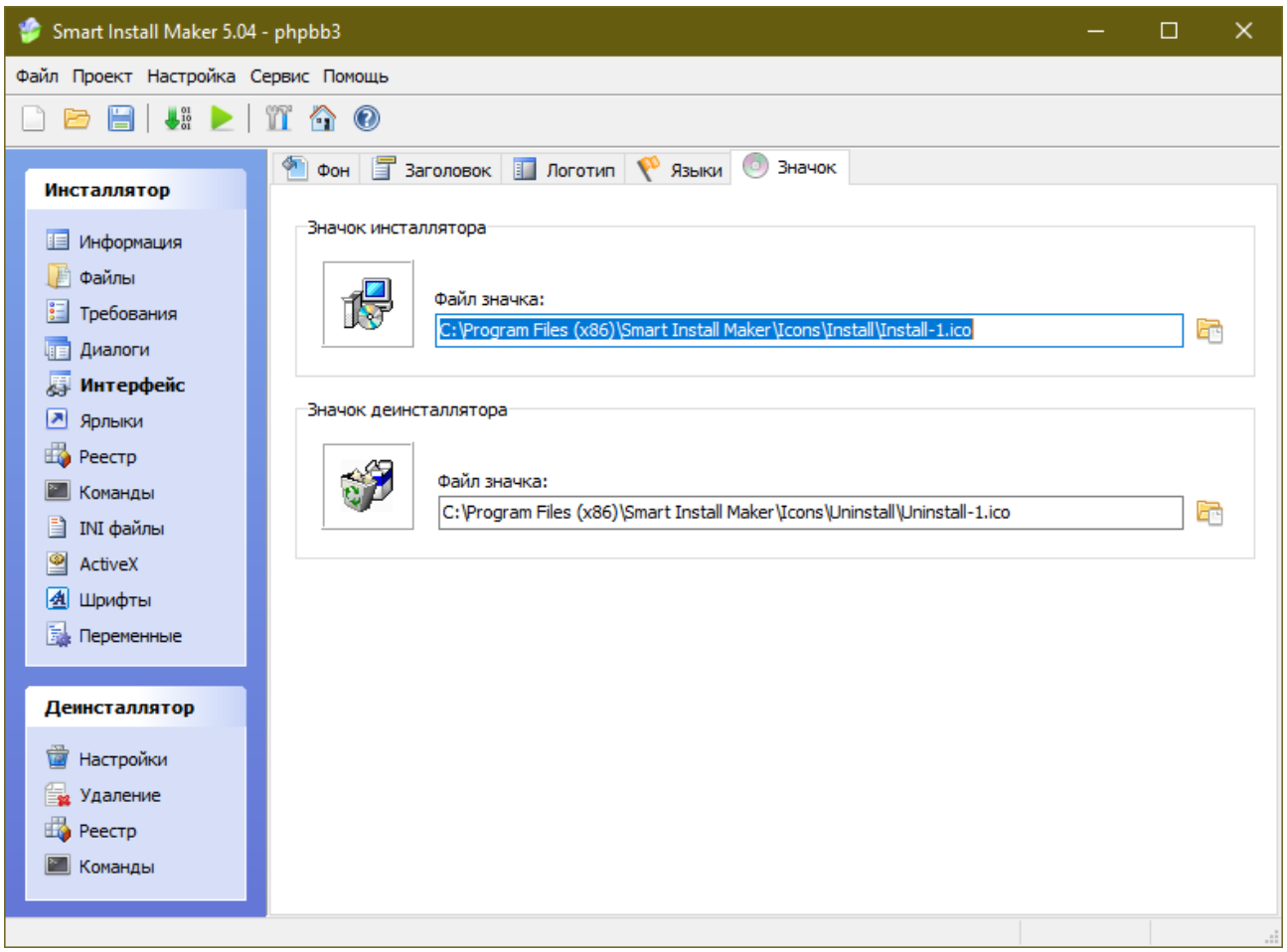



Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Просмотр
Заголовок
Описание текущего диалогового окна
Показывать картинку заголовка
Файл картинки:
C:\Program Files (x86)\Smart Install Maker\Bitmaps\Header\Header-12.png
Прозрачный фон картинки
Показывать градиентный фон
Цвет заливки:

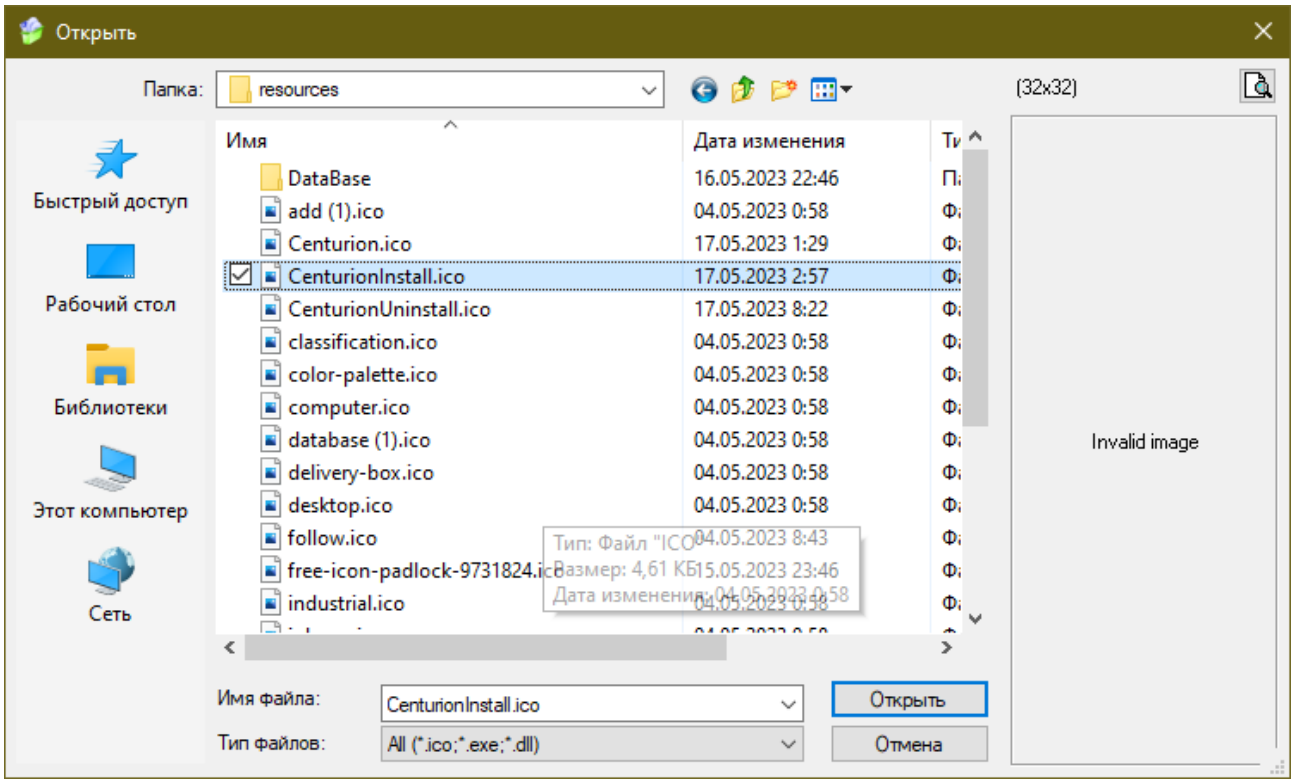

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Показывать логотип
Просмотр
Настройки
Файл картинки:
iles (x86)\Smart Install Maker\Bitmaps\Wizard\Wizard-1.bm
Прозрачный фон картинки

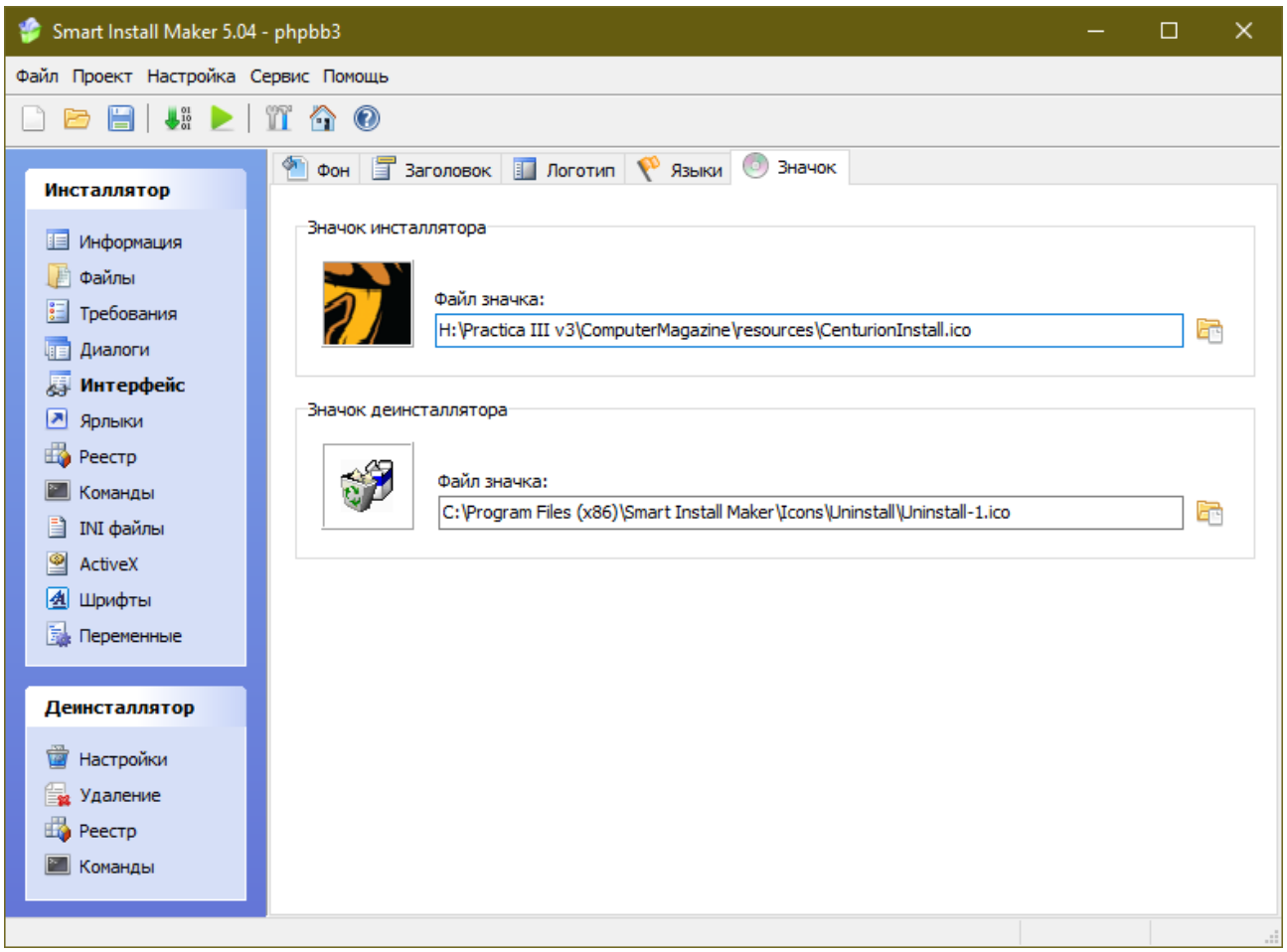

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Показывать логотип
Просмотр
Настройки
Файл картинки:
C:\Program Files (x86)\Smart Install Maker\Bitmaps\Wizard
Прозрачный фон картинки


Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Языки установки
Chinese (jтМеЦРОД)
Czech (Иелътina)
Dutch (Nederlands)
English
Estonian (Eesti)
Farsi (Farsi)
French (French)
German (Deutsch)
Greek (Еллзнйкь)
Hungarian (Magyar)
Italian (Italiano)
Korean (Korean)
Kurdish (Kurdish)
Norwegian (Norwegian)
Polish (Polski)
Portugal (Portugиκs)
Russian (Русский)
Serbian (Serbian)
Spanish (Espacol)
Отметить все
Очистить все

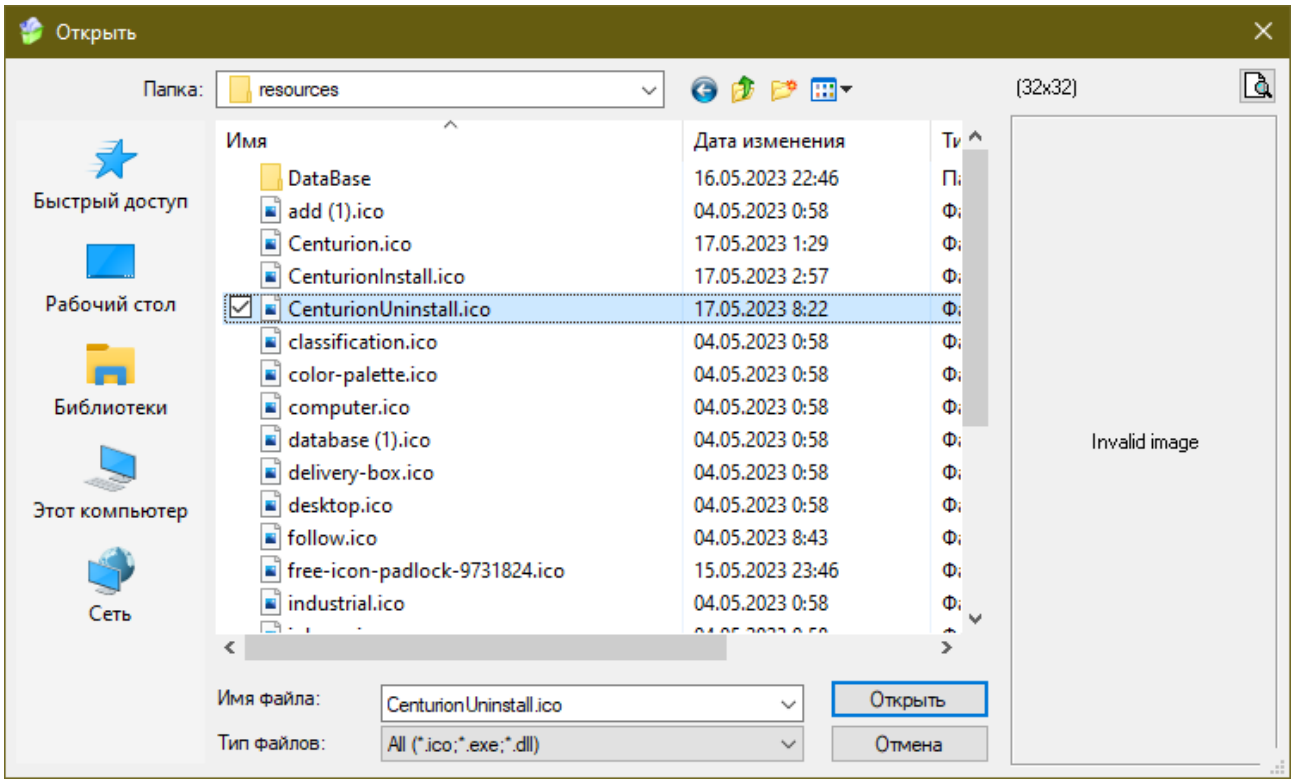


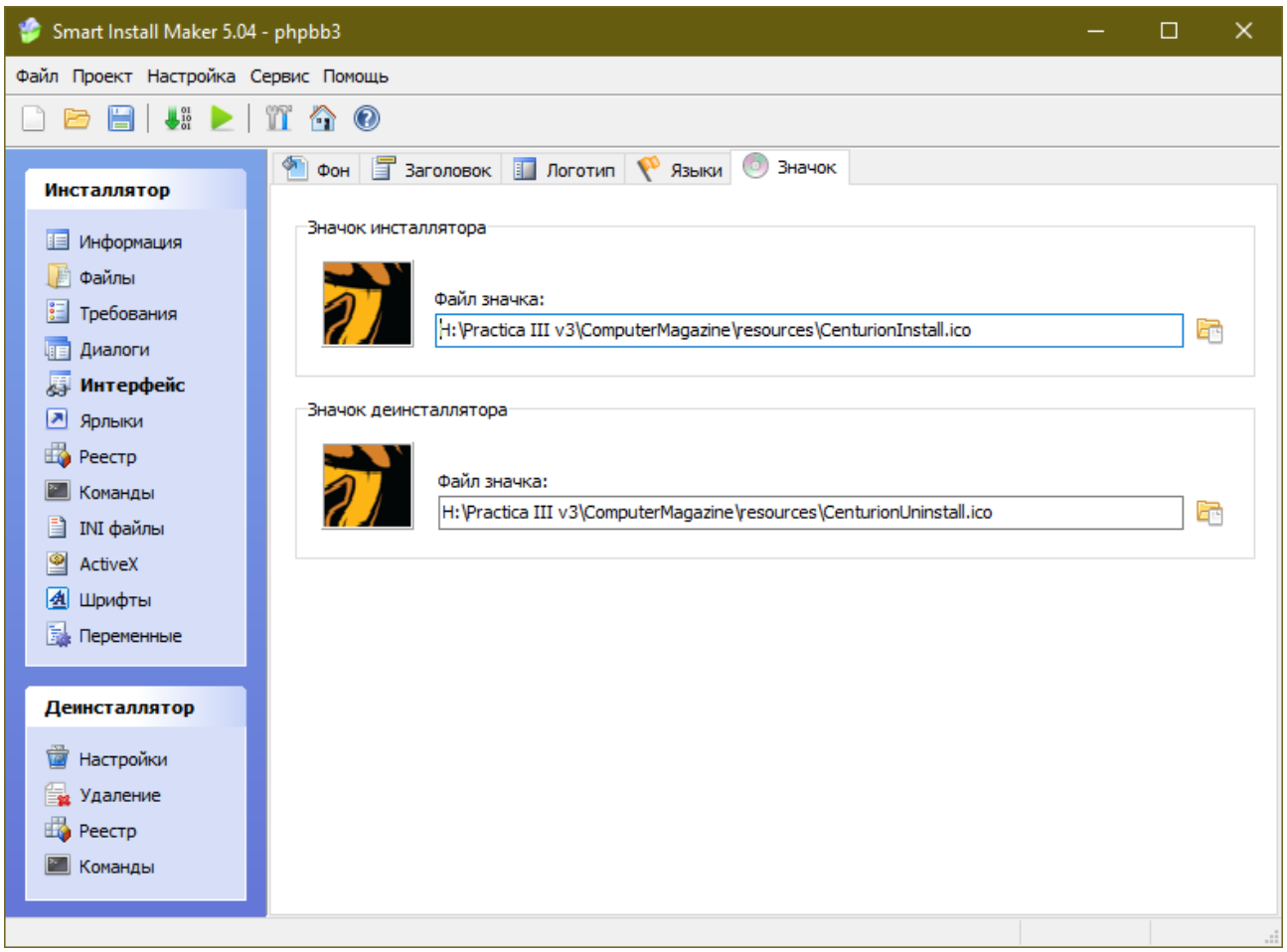
Значок Инсталлятора:

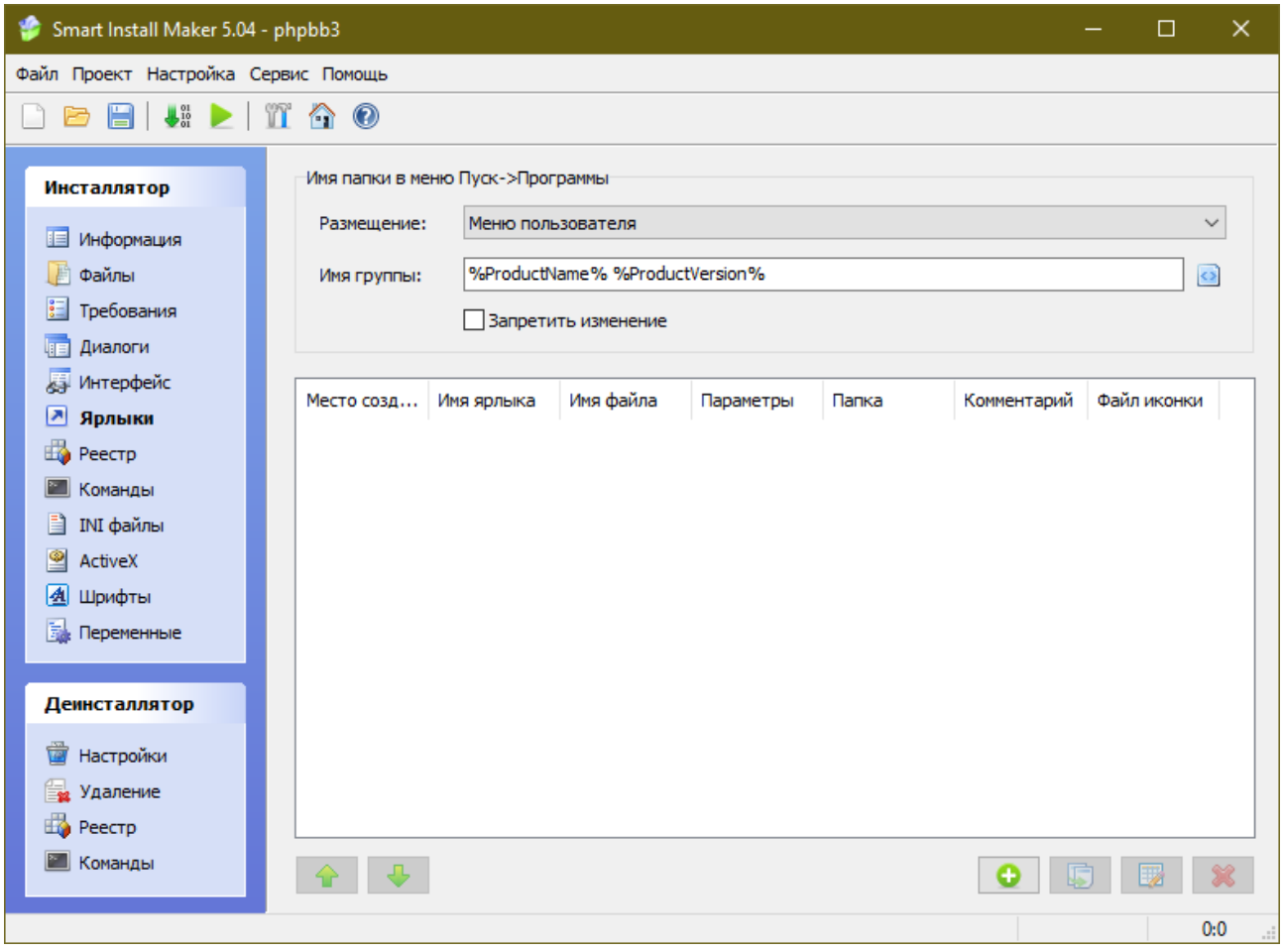




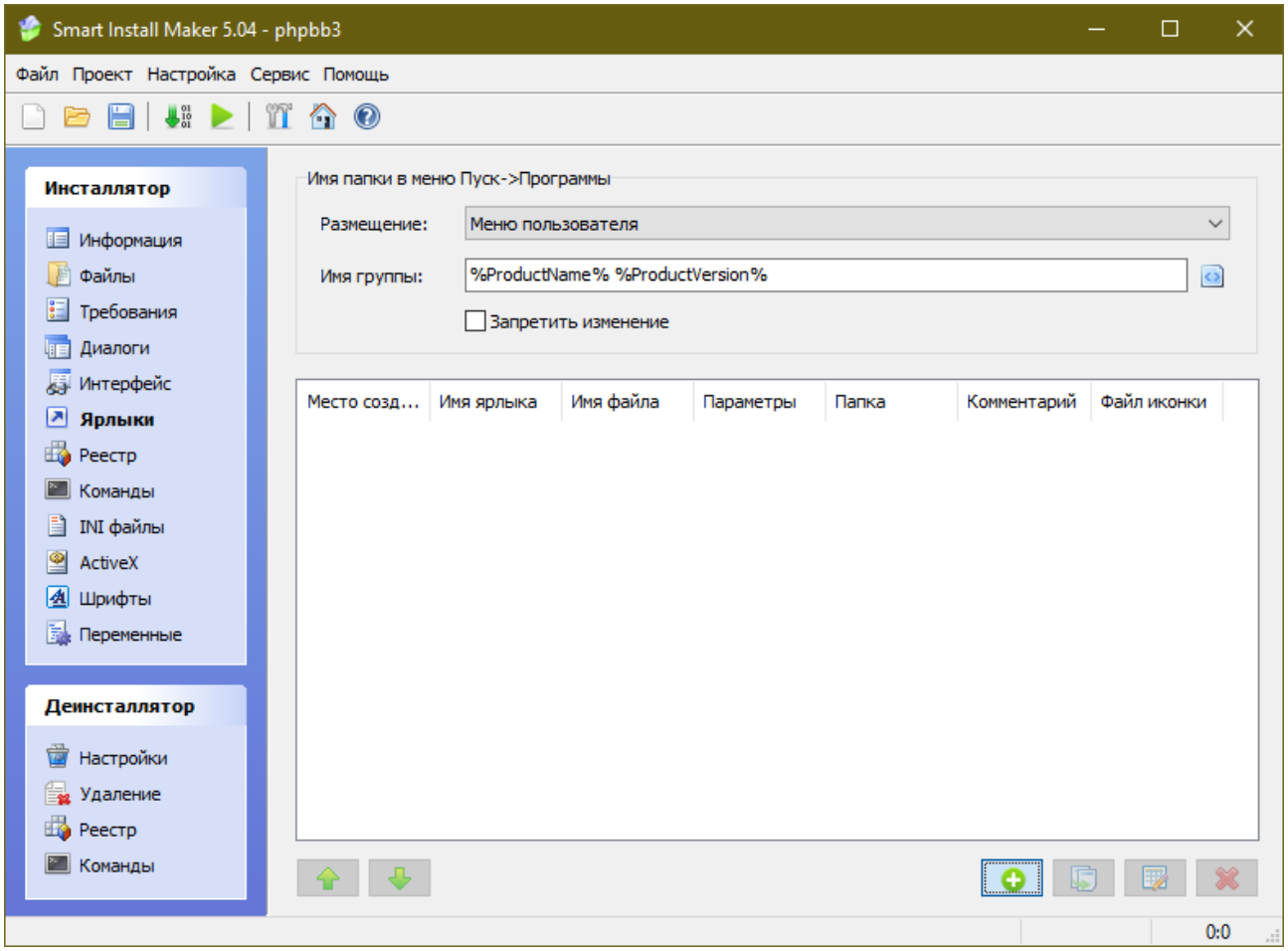
Значок Деинсталлятора:



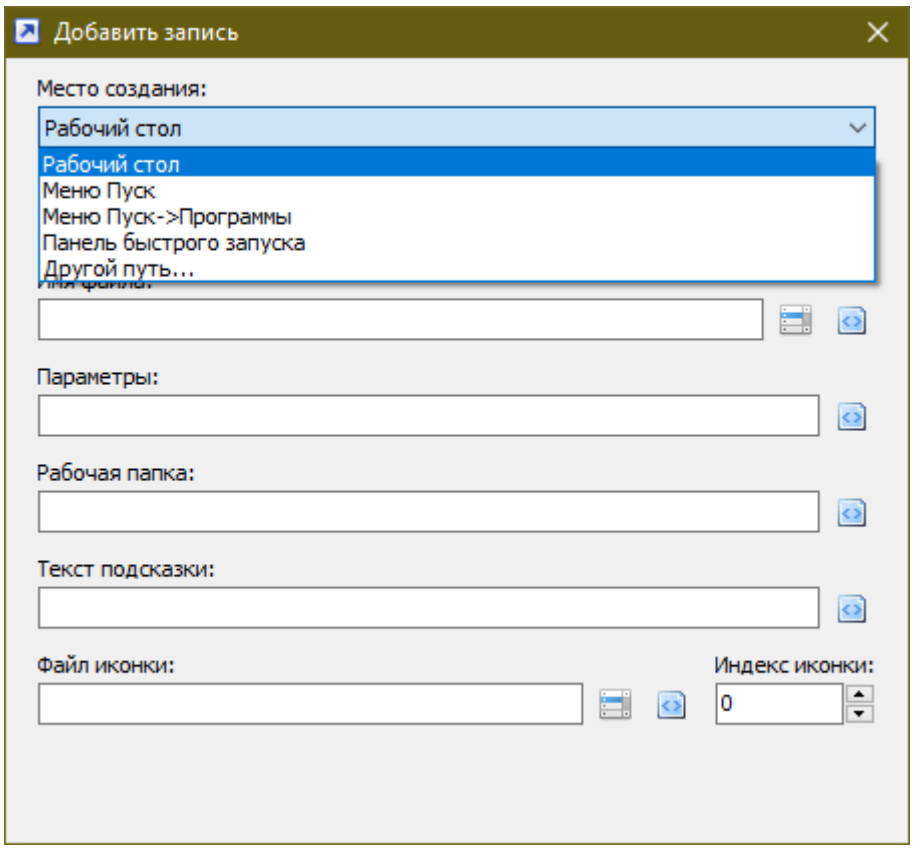

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Фон
Заголовок
Логотип
Языки
Значок
Значок инсталлятора
Файл значка:
H:\Practica III v3\ComputerMagazine\resources\CenturionInstall.ico
Значок деинсталлятора
Файл значка:
H:\Practica III v3\ComputerMagazine\resources\CenturionUninstall.ico

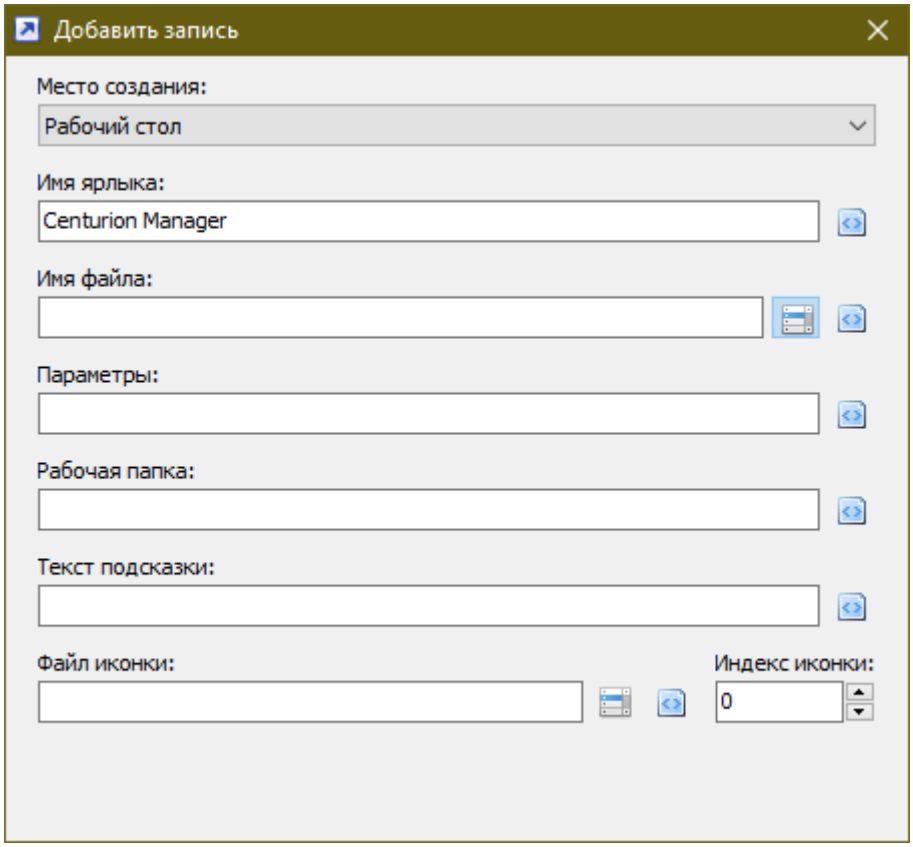

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Имя папки в меню Пуск->Программы
Размещение:
Меню пользователя
Имя группы:
%ProductName% %ProductVersion%
Запретить изменение
Место созд...
Имя ярлыка
Имя файла
Параметры
Папка
Комментарий
Файл иконки
0:0

Настраиваем иконки приложения:

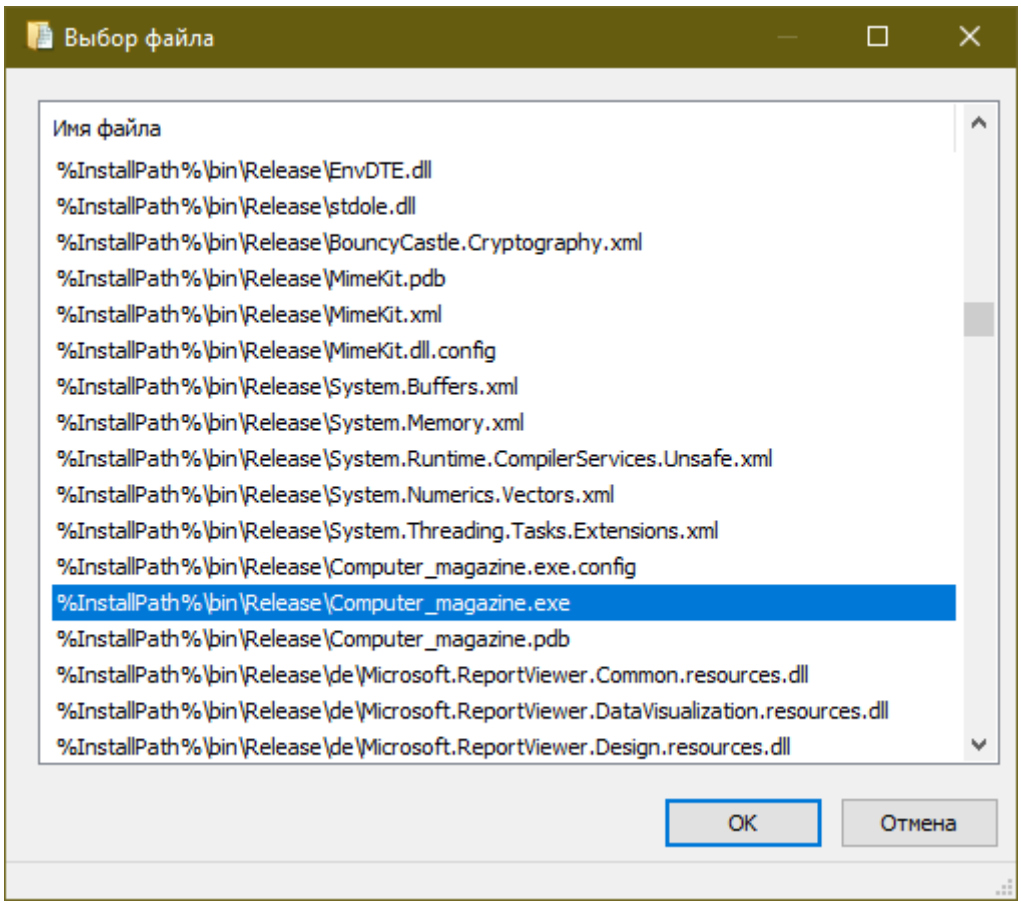


Рабочего стола:

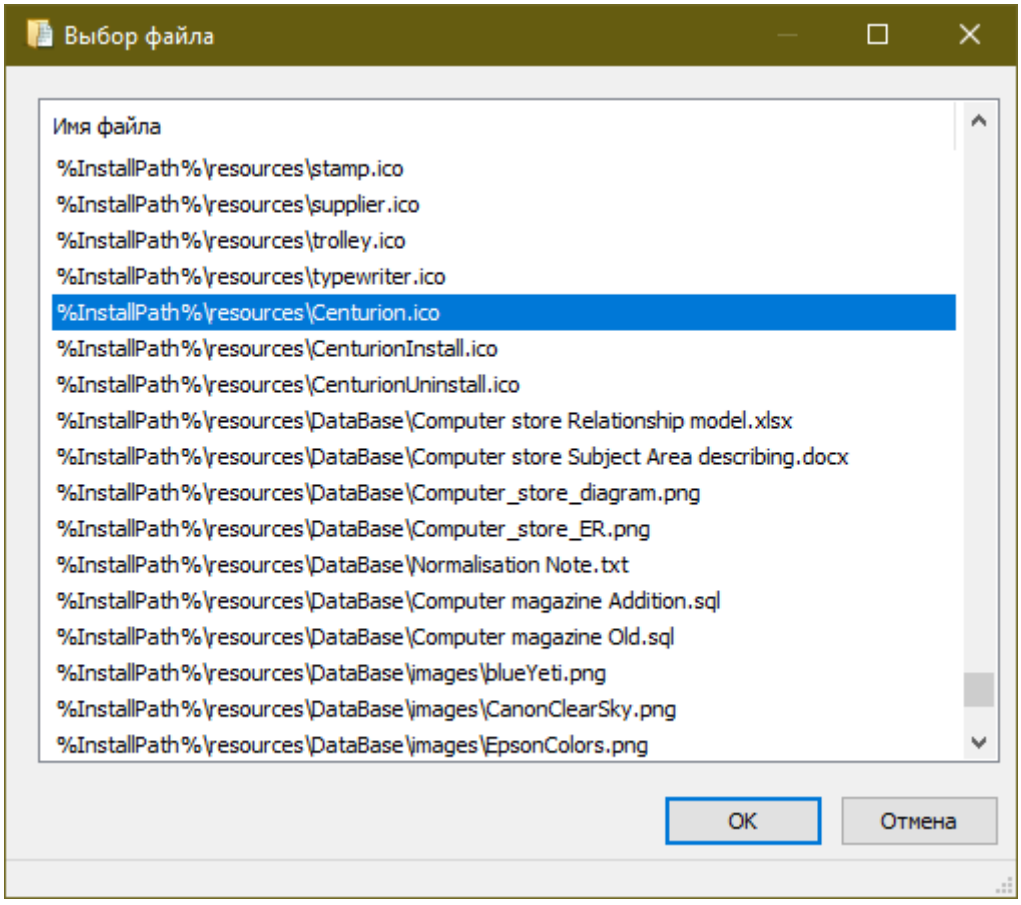


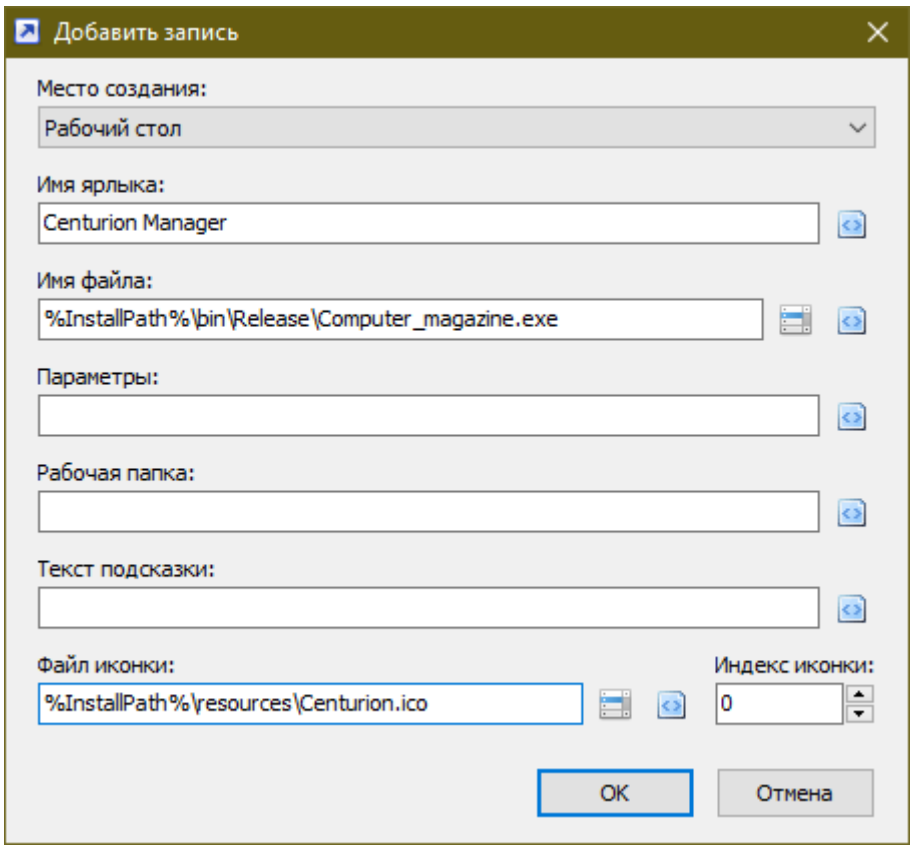


Ищем путь до .exe файла (у меня это *'папка проекта'/bin/relize/ComputerMagazine.exe*, но у вас это может быть *'папка проекта'/bin/debug/ComputerMagazine.exe*)

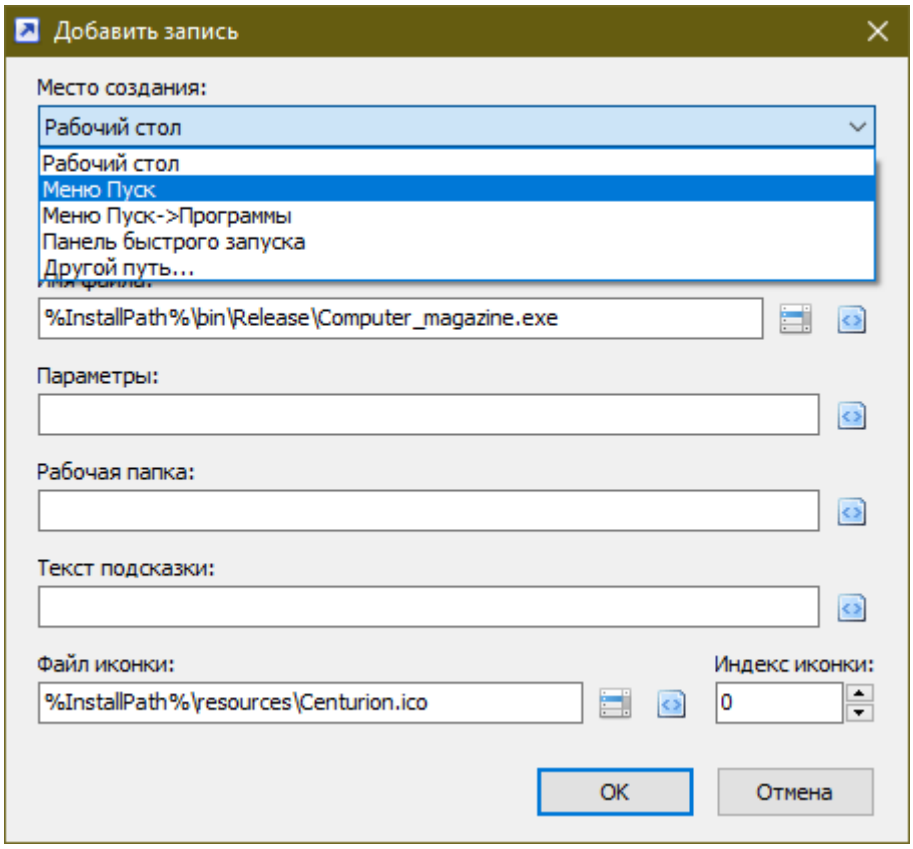


И выбираем файл иконки (у меня это *'папка проекта'/resources/Centurion.ico*):

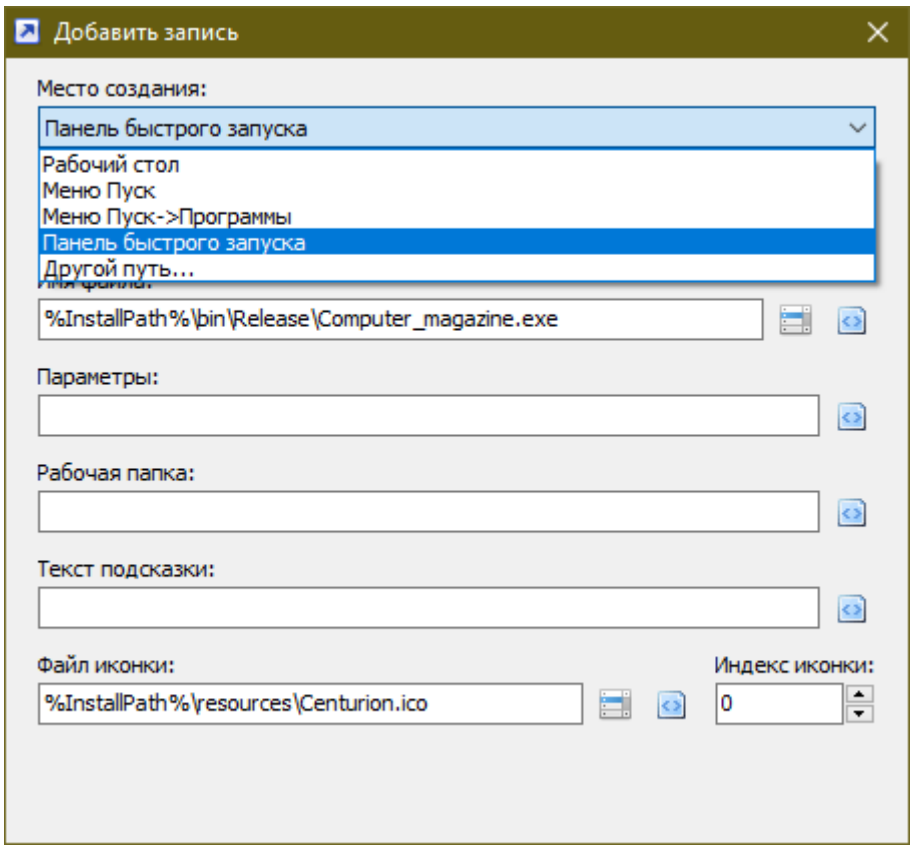


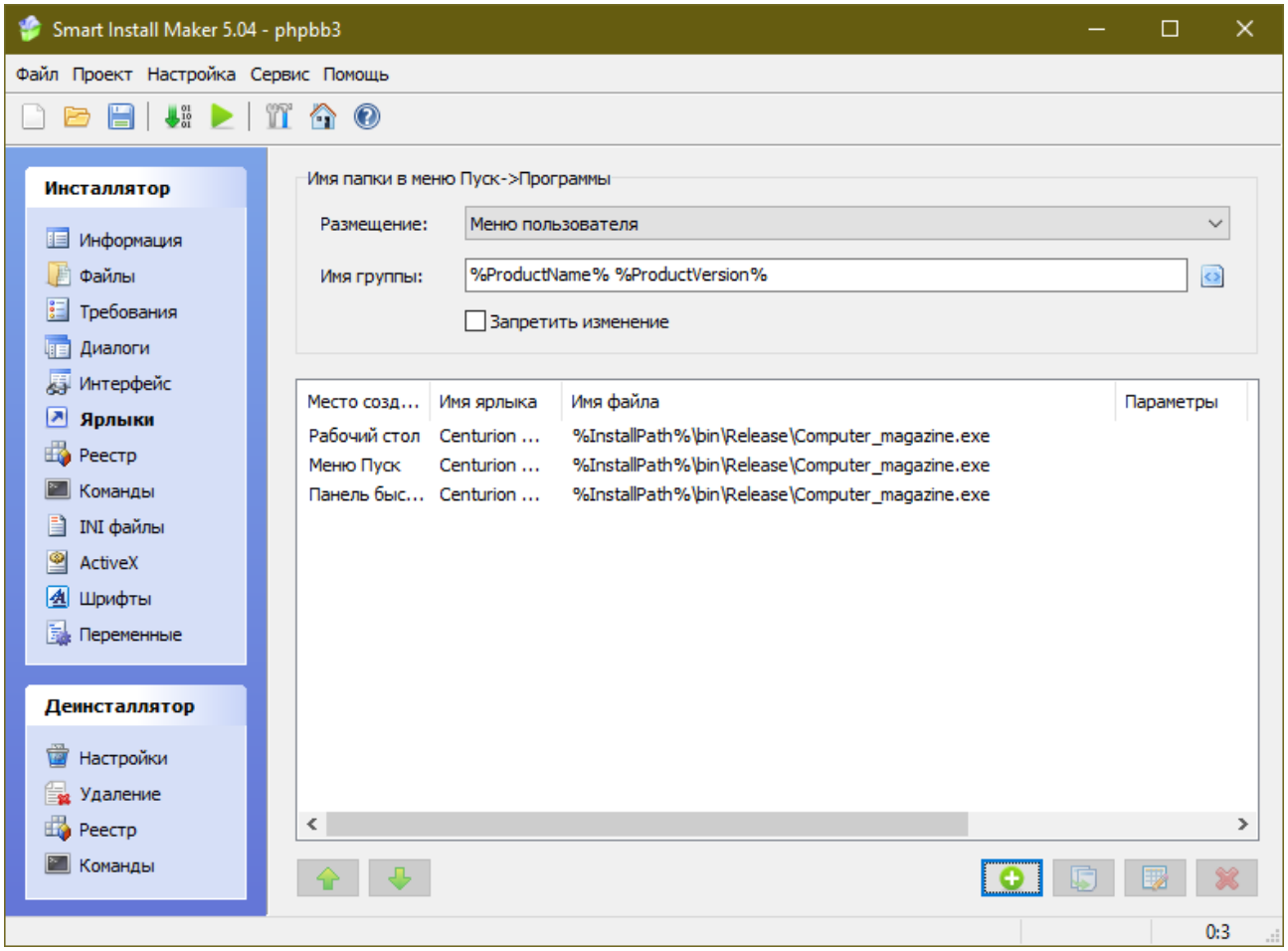


То же самое и для ярлыка в меню пуск:

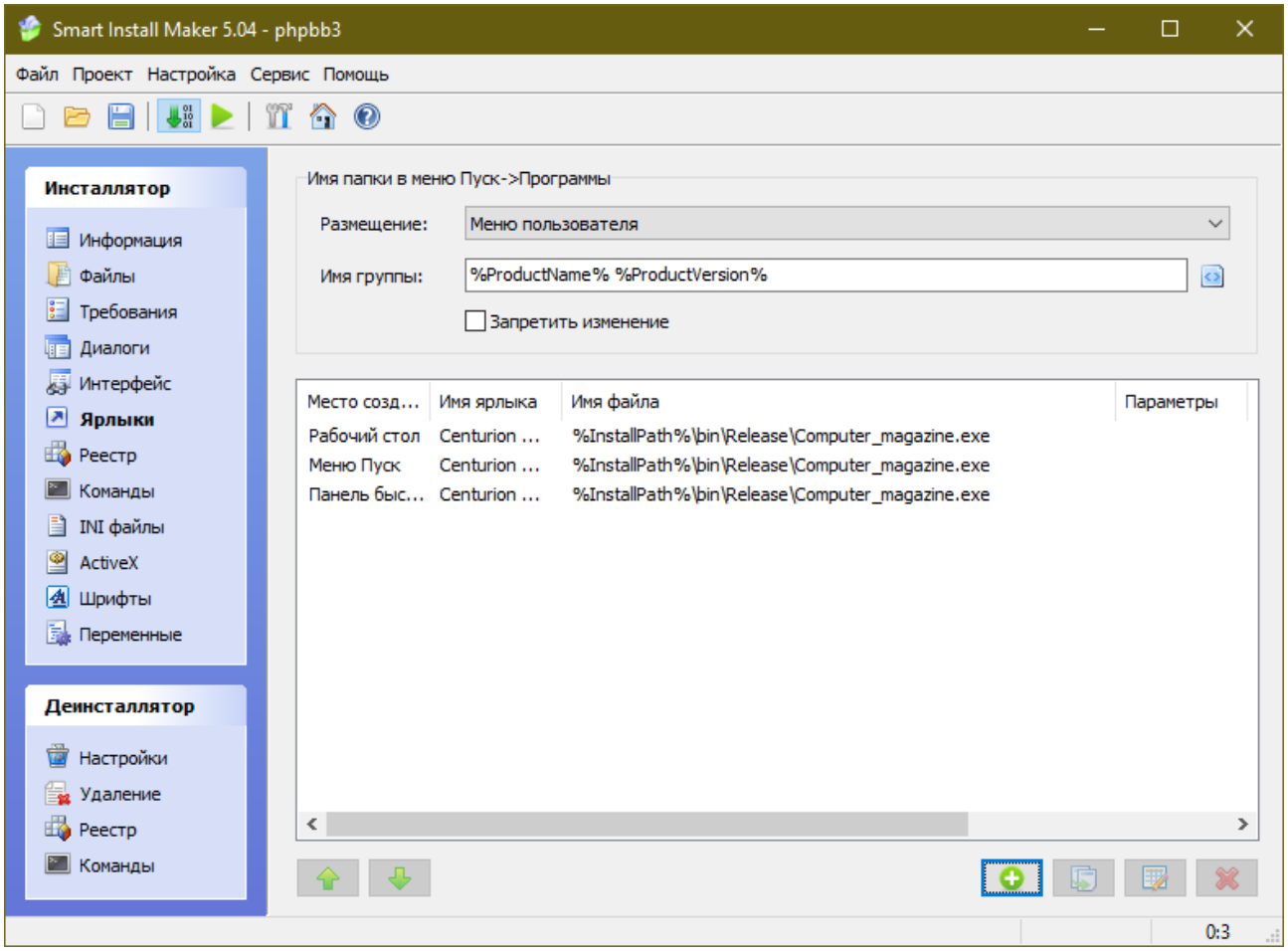


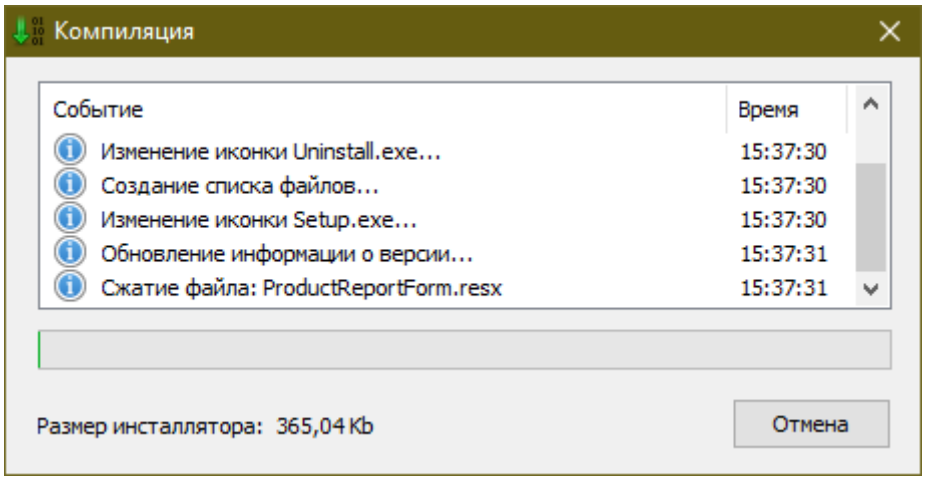
Также можем добавить и на панель инструментов:

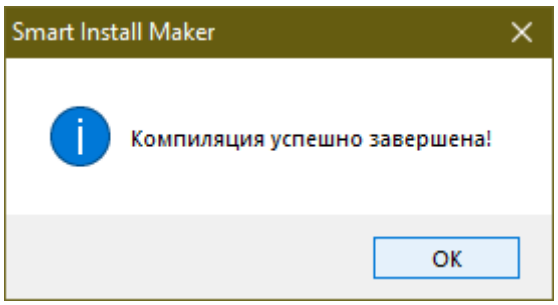


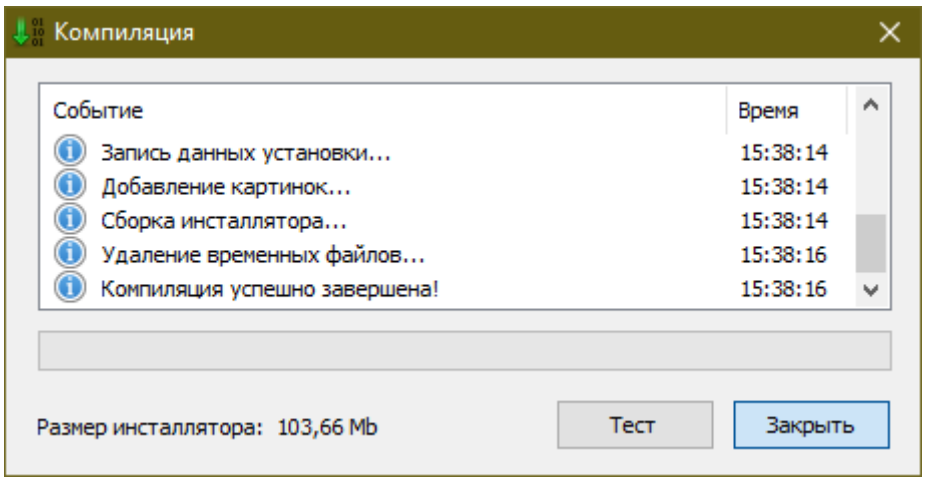


Нажимаем на «Компилировать» и собираем наш Setup.exe файл…

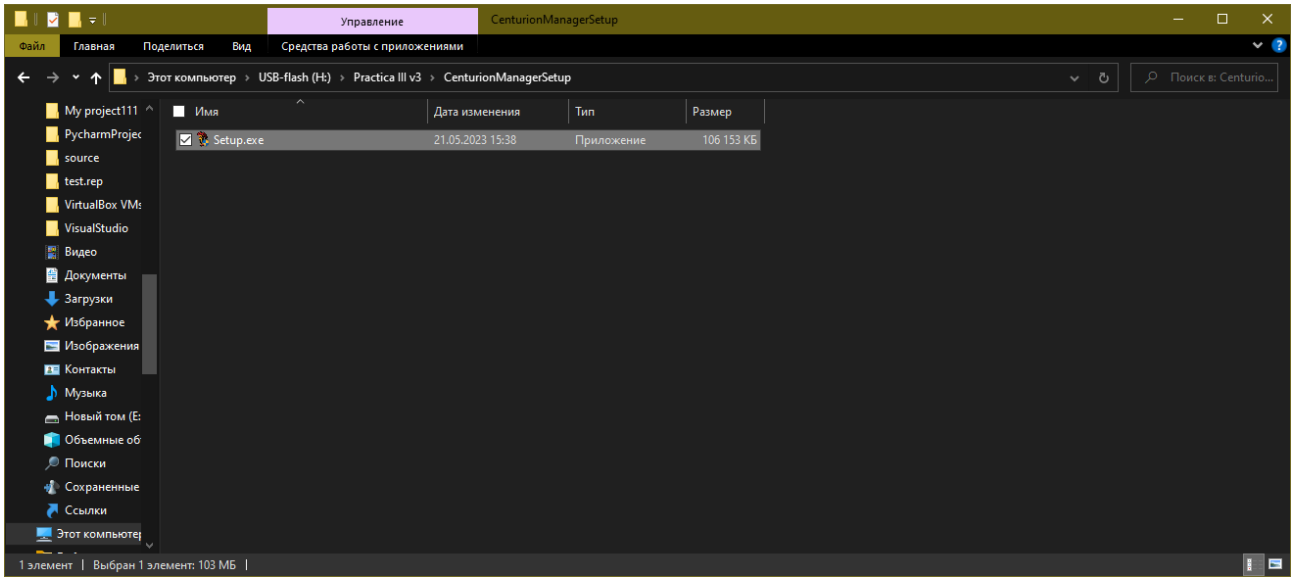

Smart Install Maker 5.04 - phpbb3
Файл Проект Настройка Сервис Помощь
Инсталлятор
Информация
Файлы
Требования
Диалоги
Интерфейс
Ярлыки
Реестр
Команды
INI файлы
ActiveX
Шрифты
Переменные
Деинсталлятор
Настройки
Удаление
Реестр
Команды
Имя папки в меню Пуск->Программы
Размещение:
Меню пользователя
Имя группы:
%ProductName% %ProductVersion%
Запретить изменение
Место созд...
Имя ярлыка
Имя файла
Параметры
Рабочий стол
Centurion ...
%InstallPath%\bin\Release\Computer_magazine.exe
Меню Пуск
Centurion ...
%InstallPath%\bin\Release\Computer_magazine.exe
Панель быс...
Centurion ...
%InstallPath%\bin\Release\Computer_magazine.exe
0:3

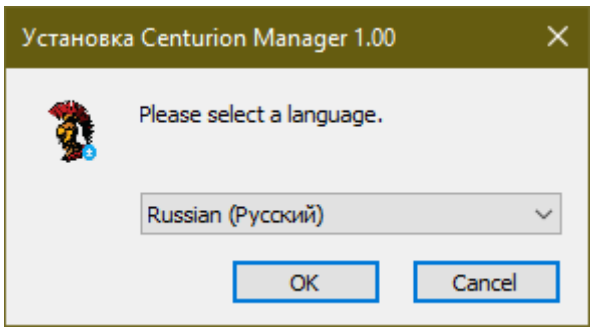

Компиляция
Событие
Время
Изменение иконки Uninstall.exe...
15:37:30
Создание списка файлов...
15:37:30
Изменение иконки Setup.exe...
15:37:30
Обновление информации о версии...
15:37:31
Сжатие файла: ProductReportForm.resx
15:37:31
Размер инсталлятора: 365,04 Kb
Отмена

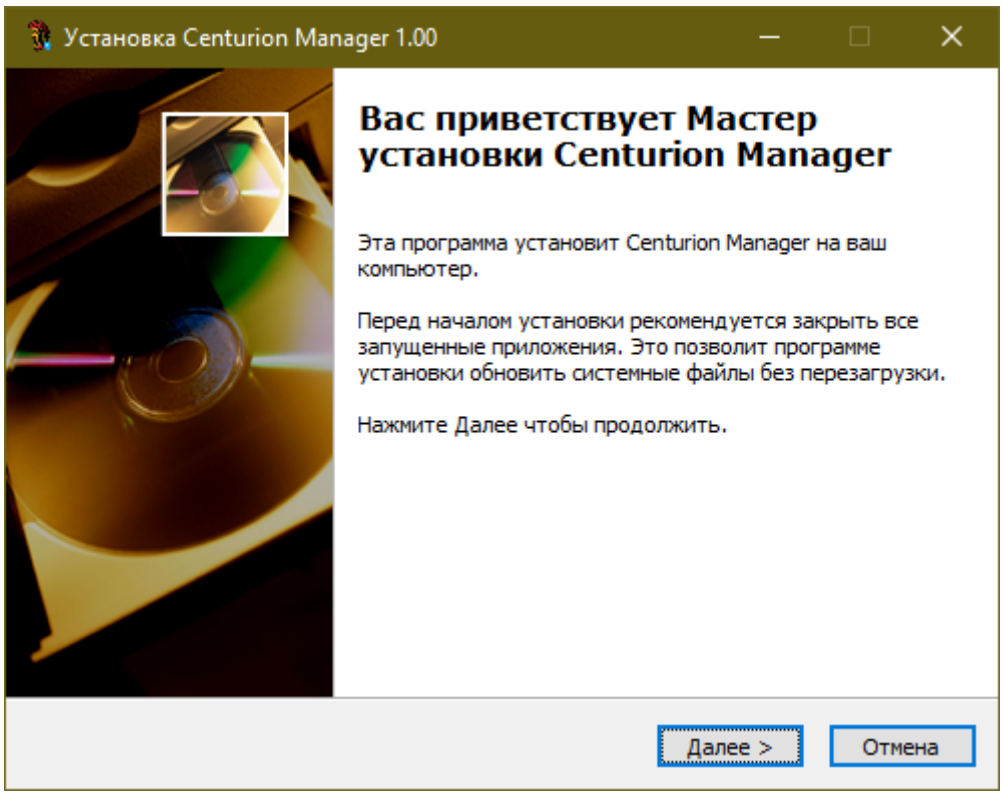

Smart Install Maker
Компиляция успешно завершена!
OK

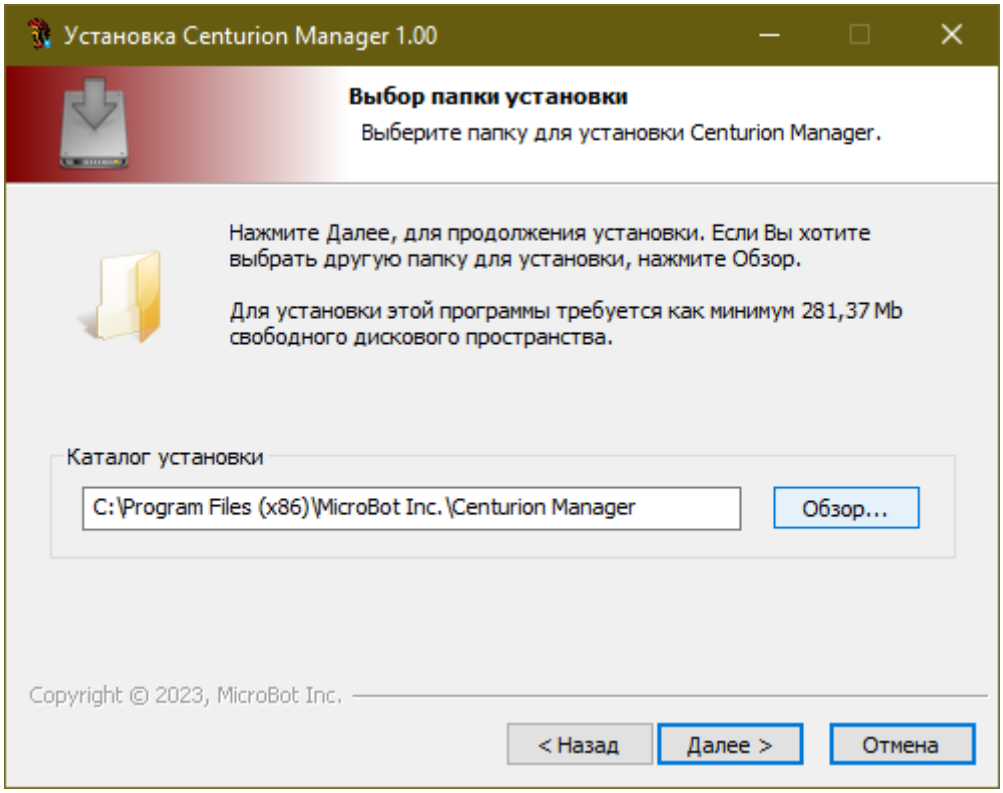


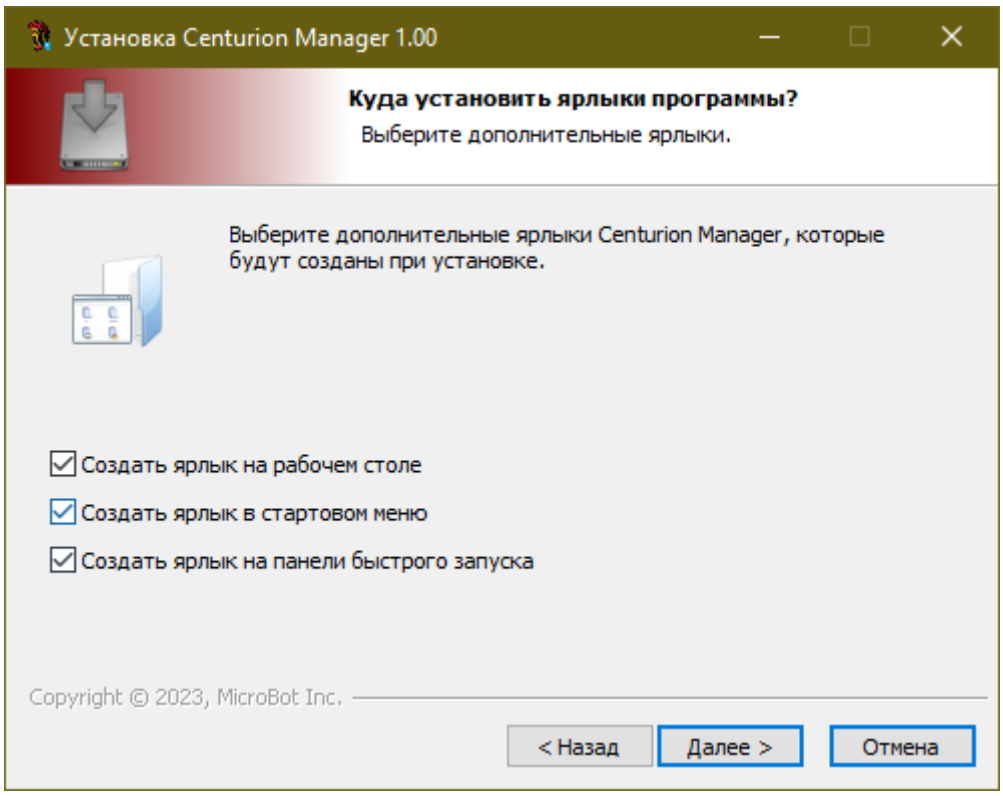
Как он работает:

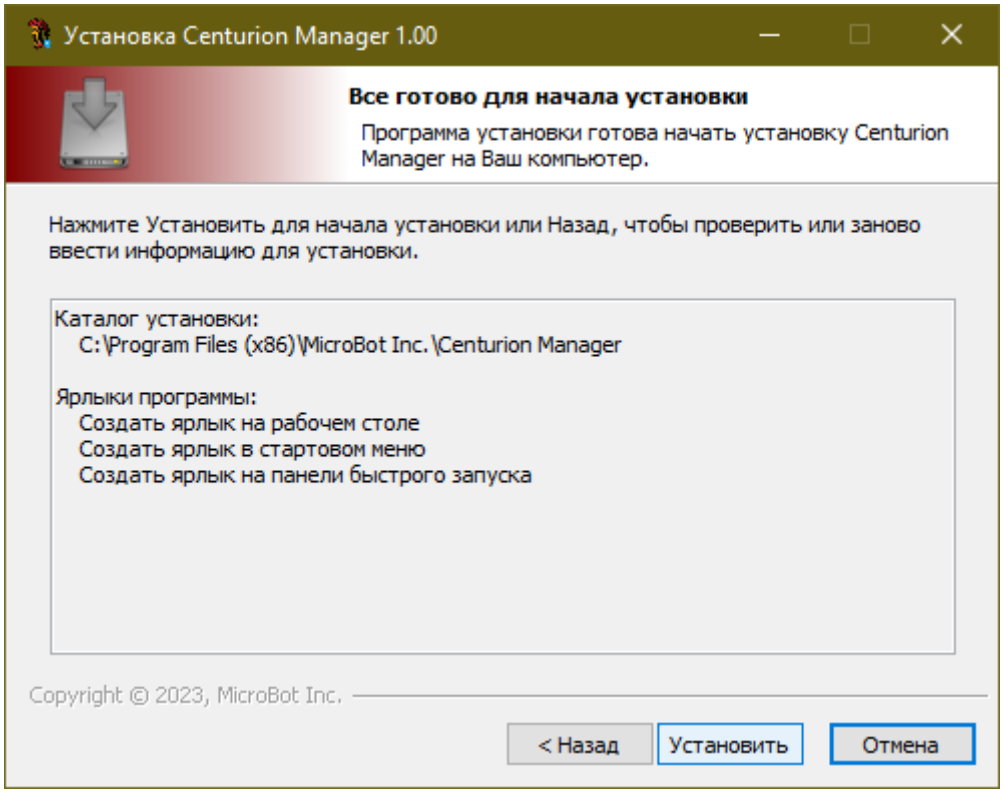


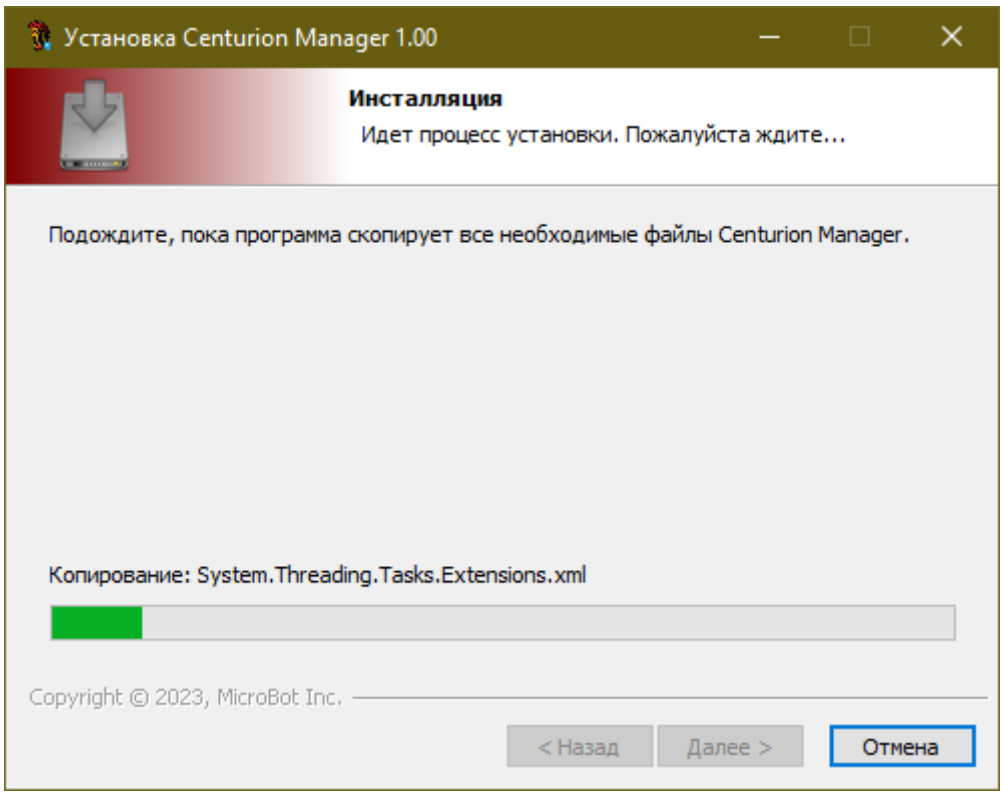


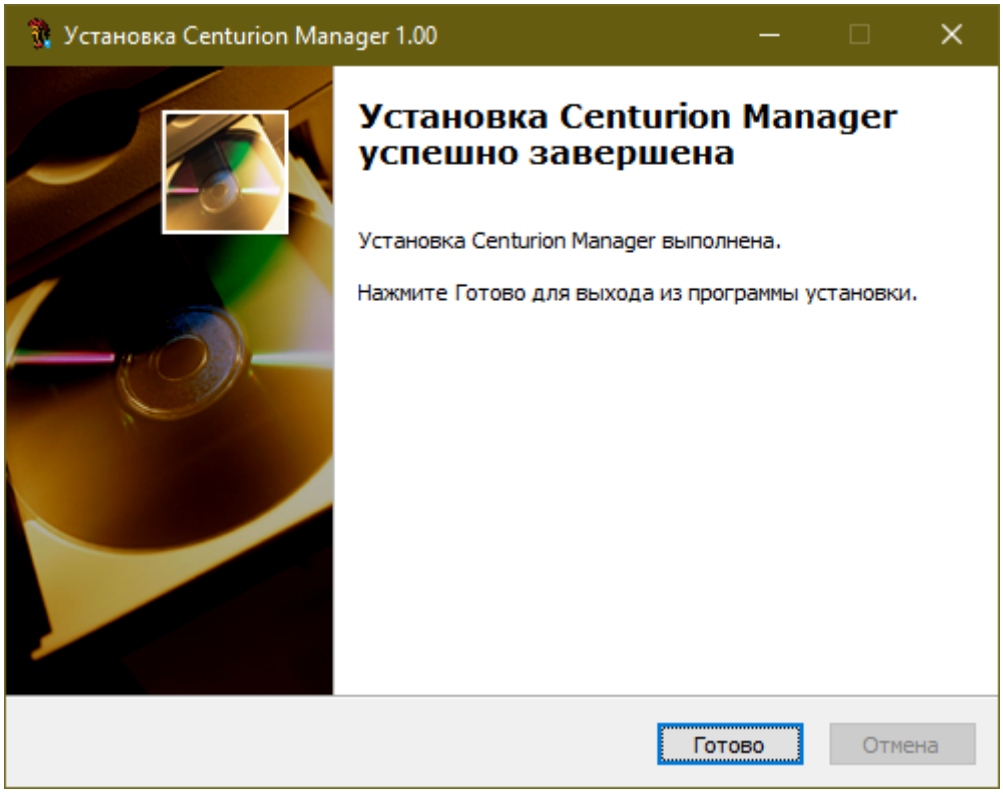

Установка Centurion Manager 1.00
Вас приветствует Мастер установки Centurion Manager
Эта программа установит Centurion Manager на ваш компьютер.
Перед началом установки рекомендуется закрыть все запущенные приложения. Это позволит программе установки обновить системные файлы без перезагрузки.
Нажмите Далее чтобы продолжить.
Далее >
Отмена


Установка Centurion Manager 1.00
Выбор папки установки
Выберите папку для установки Centurion Manager.
Нажмите Далее, для продолжения установки. Если Вы хотите выбрать другую папку для установки, нажмите Обзор.
Для установки этой программы требуется как минимум 281,37 Mb свободного дискового пространства.
Каталог установки
C:\Program Files (x86)\MicroBot Inc.\Centurion Manager
Обзор...
Copyright © 2023, MicroBot Inc.
< Назад
Далее >
Отмена

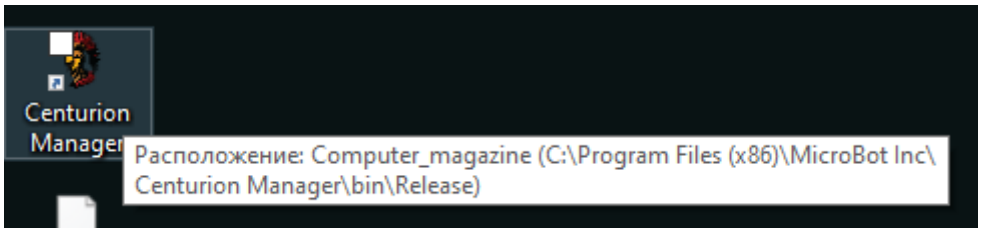

Установка Centurion Manager 1.00
Куда установить ярлыки программы?
Выберите дополнительные ярлыки.
Выберите дополнительные ярлыки Centurion Manager, которые будут созданы при установке.
Создать ярлык на рабочем столе
Создать ярлык в стартовом меню
Создать ярлык на панели быстрого запуска
Copyright © 2023, MicroBot Inc.
< Назад
Далее >
Отмена

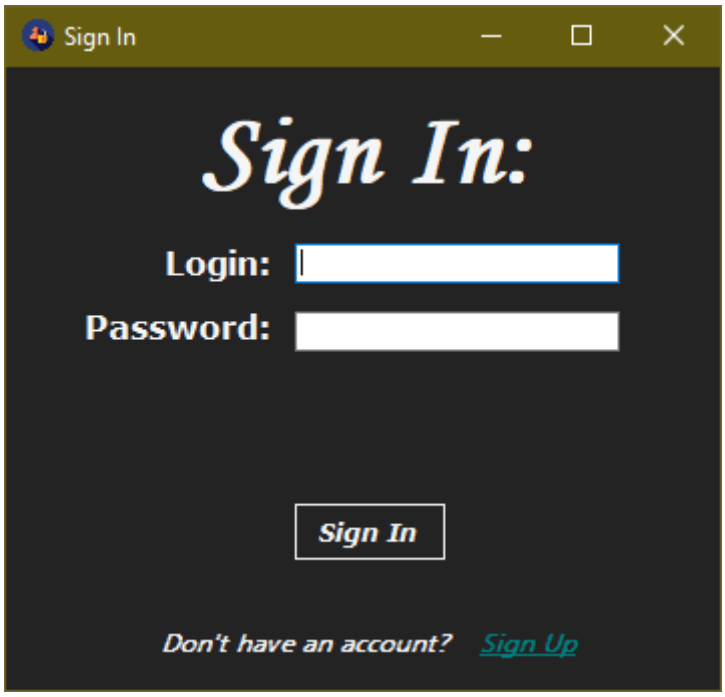

Установка Centurion Manager 1.00
Все готово для начала установки
Программа установки готова начать установку Centurion Manager на Ваш компьютер.
Нажмите Установить для начала установки или Назад, чтобы проверить или заново ввести информацию для установки.
Каталог установки:
C:\Program Files (x86)\MicroBot Inc.\Centurion Manager
Ярлыки программы:
Создать ярлык на рабочем столе
Создать ярлык в стартовом меню
Создать ярлык на панели быстрого запуска
Copyright © 2023, MicroBot Inc.
< Назад
Установить
Отмена

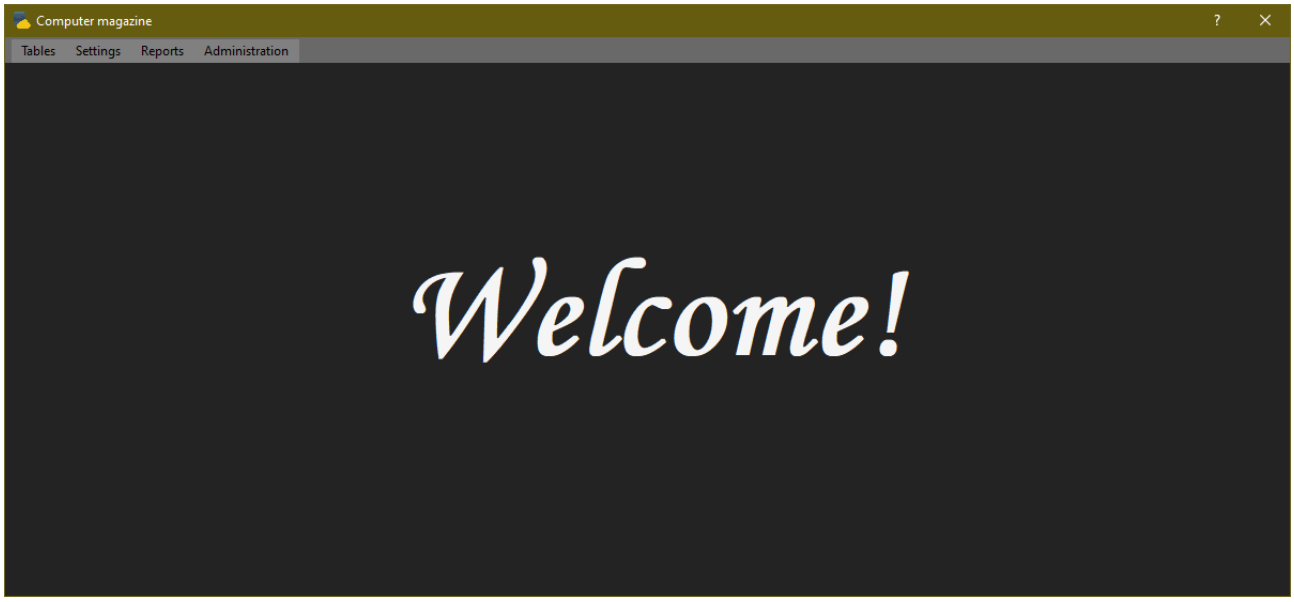

Установка Centurion Manager 1.00
Инсталляция
Идет процесс установки. Пожалуйста ждите...
Подождите, пока программа скопирует все необходимые файлы Centurion Manager.
Копирование: System.Threading.Tasks.Extensions.xml
Copyright © 2023, MicroBot Inc.
< Назад
Далее >
Отмена


Установка Centurion Manager 1.00
Установка Centurion Manager успешно завершена
Установка Centurion Manager выполнена.
Нажмите Готово для выхода из программы установки.
Готово
Отмена

Работоспособность программы после установки:

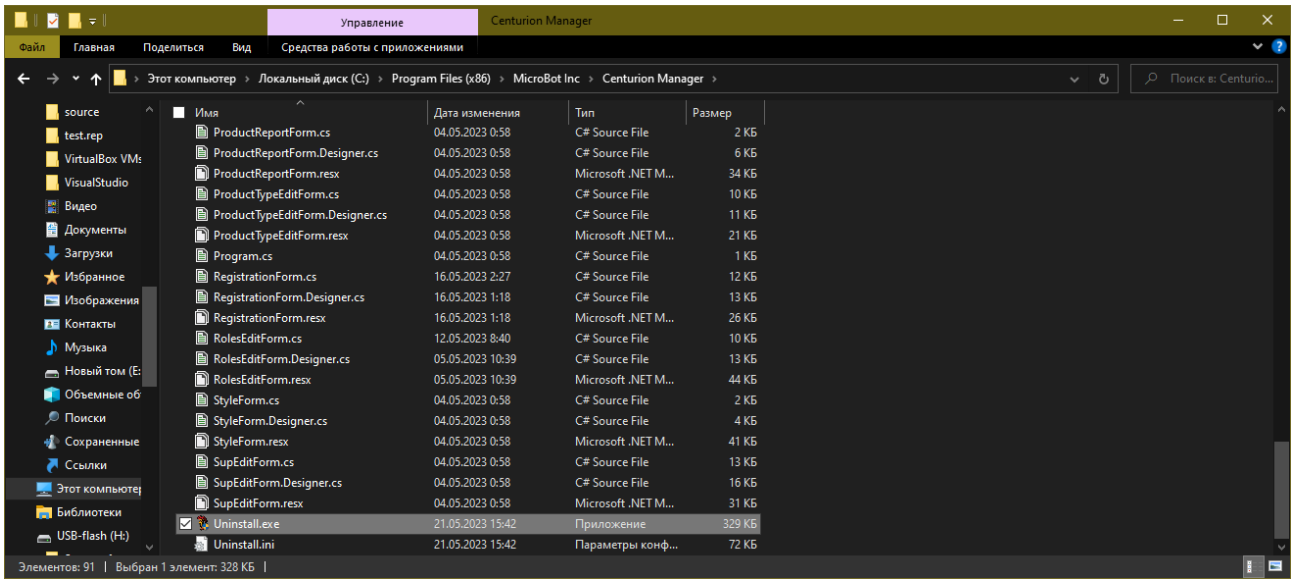




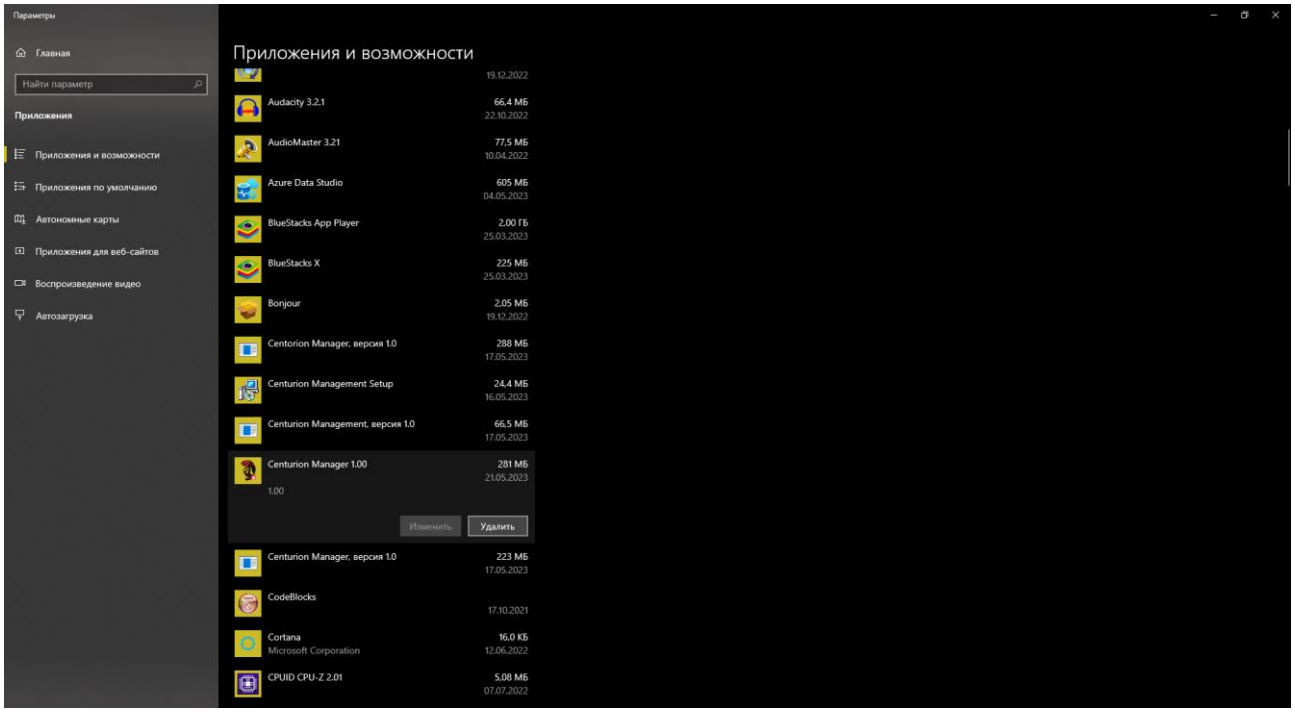


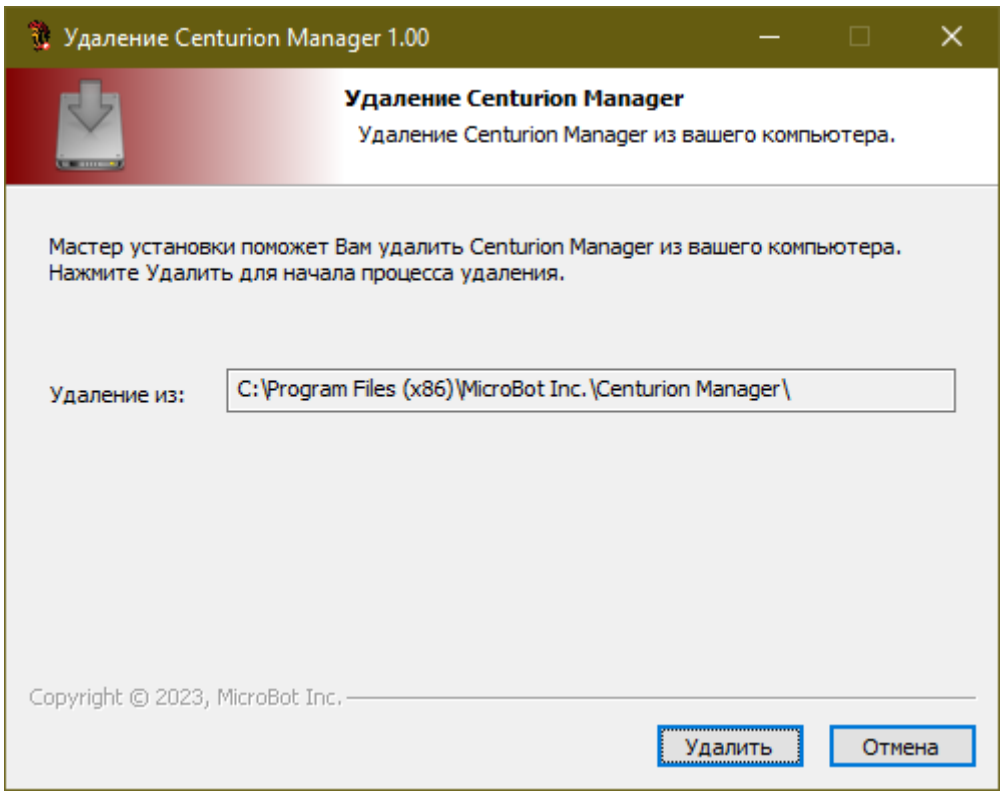
Деинсталляция приложения:

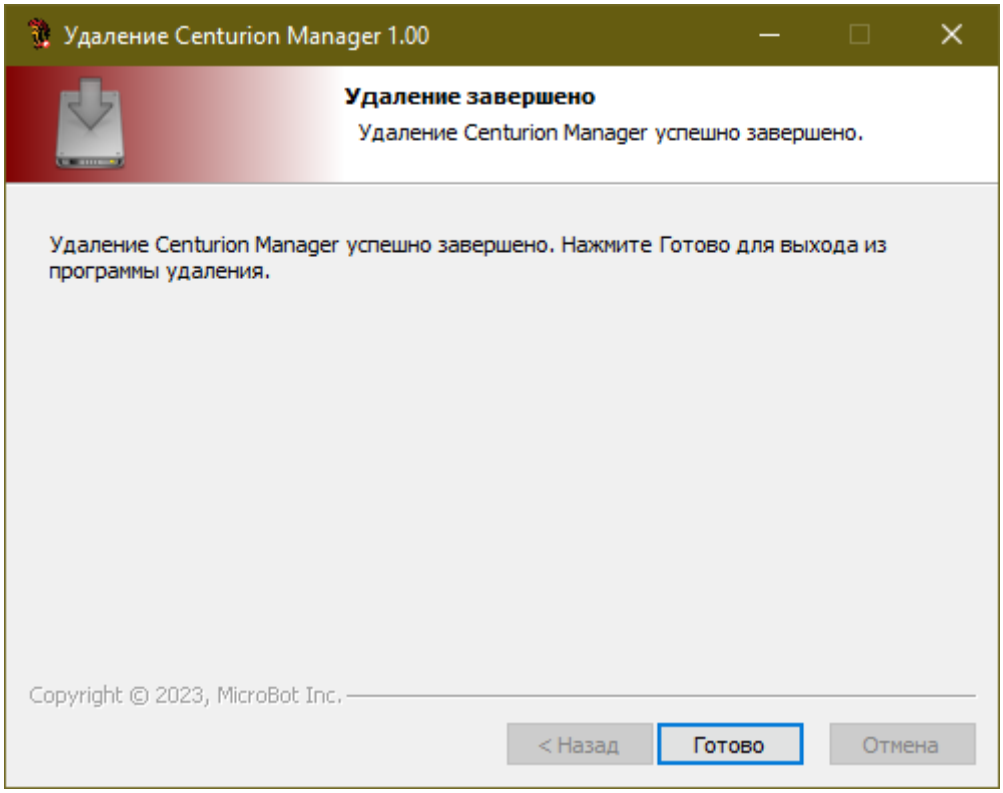
Либо:



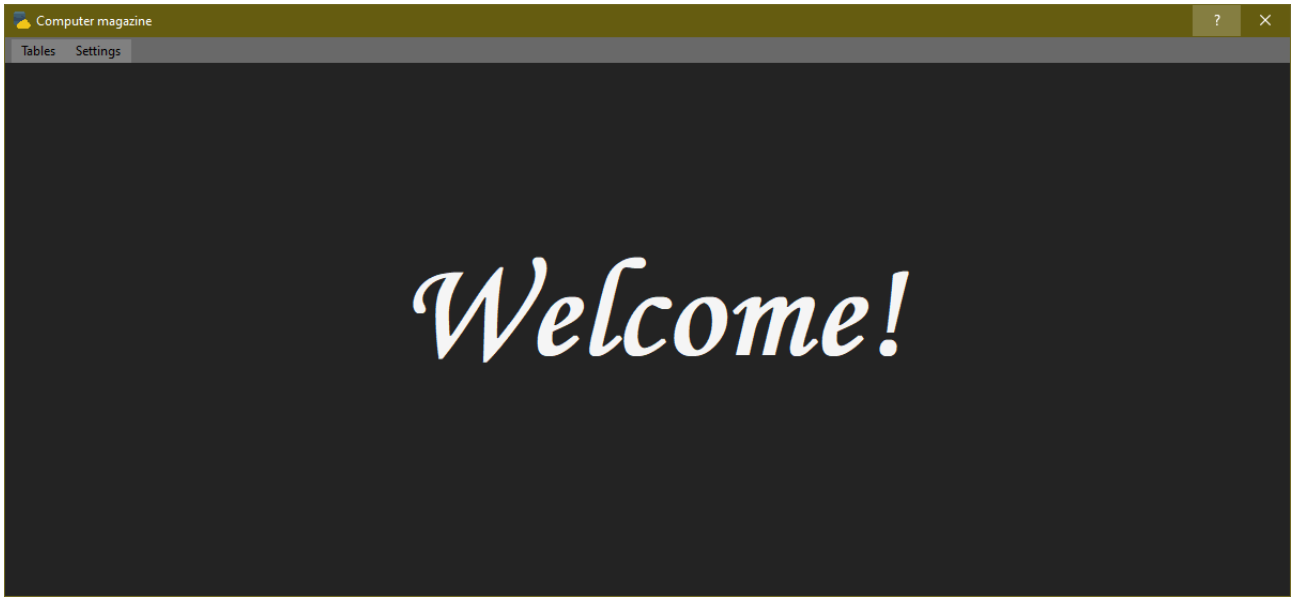
Либо:

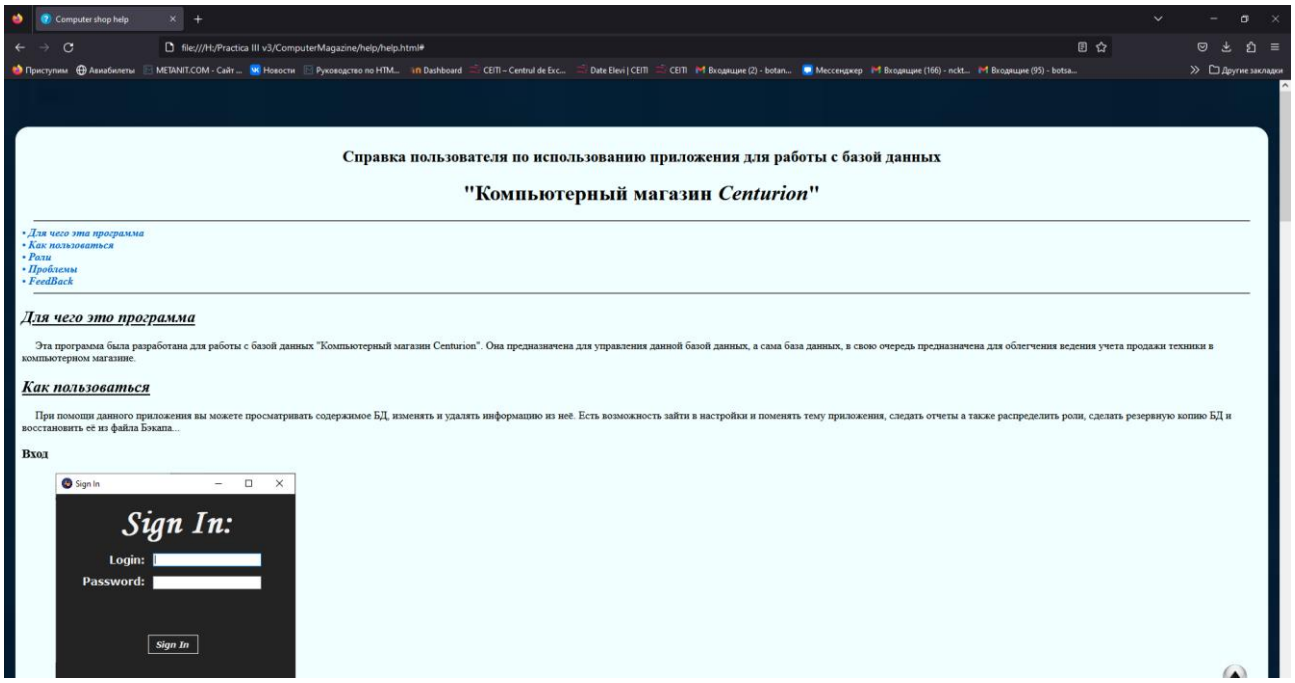


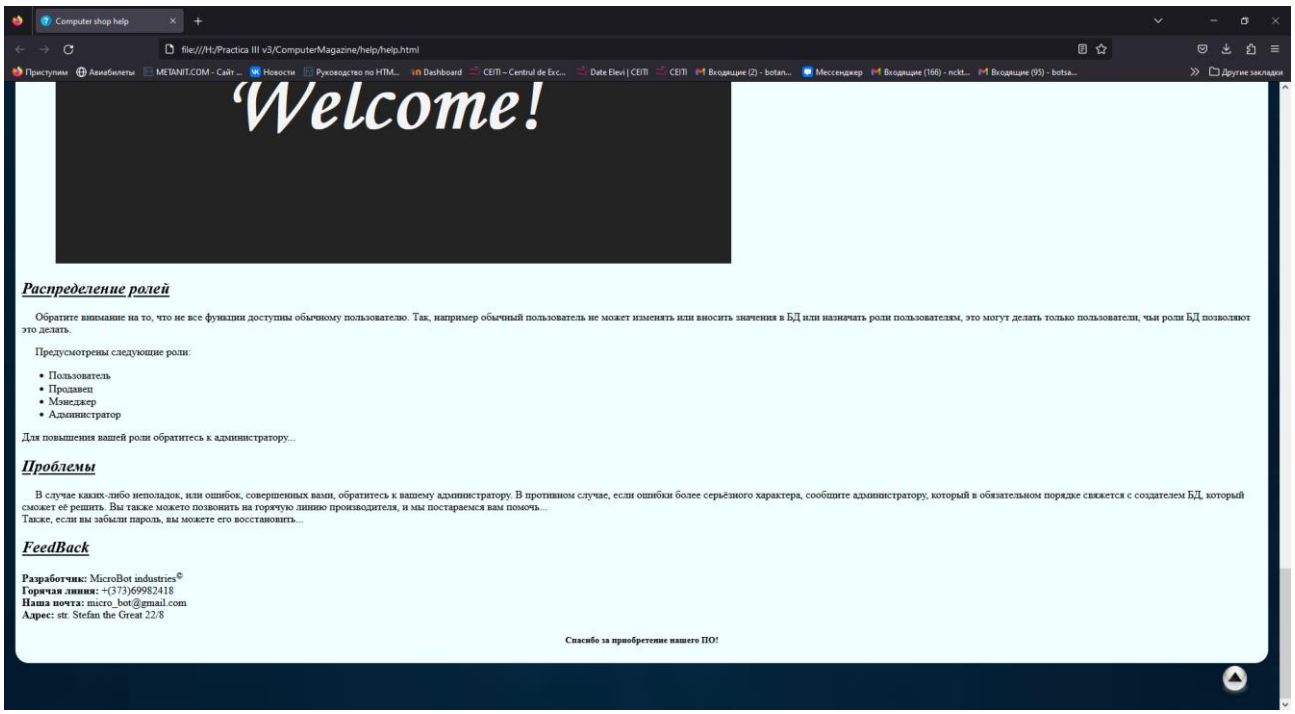

Удаление Centurion Manager 1.00
Удаление Centurion Manager
Удаление Centurion Manager из вашего компьютера.
Мастер установки поможет Вам удалить Centurion Manager из вашего компьютера.
Нажмите Удалить для начала процесса удаления.
Удаление из:
C:\Program Files (x86)\MicroBot Inc.\Centurion Manager\
Copyright © 2023, MicroBot Inc.
Удалить
Отмена

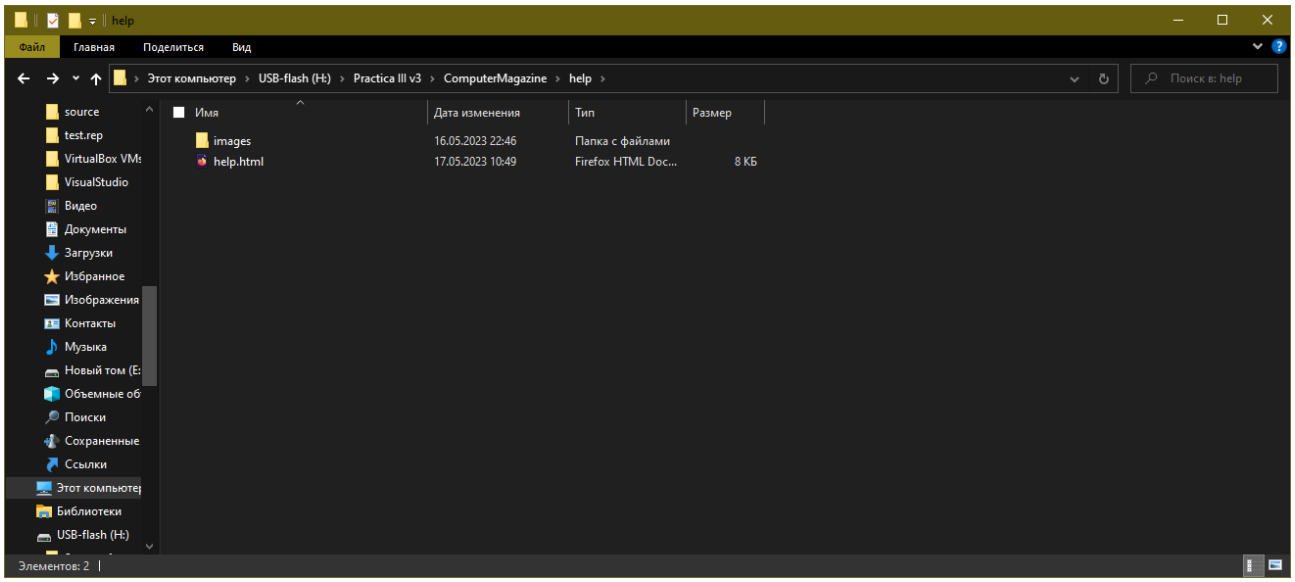

Удаление Centurion Manager 1.00
Удаление завершено
Удаление Centurion Manager успешно завершено.
Удаление Centurion Manager успешно завершено. Нажмите Готово для выхода из программы удаления.
Copyright © 2023, MicroBot Inc.
< Назад
Готово
Отмена

Руководство пользователя









Код HTML:

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Computer shop help</title>
    <link rel="icon" type="images/png" href="../resources/question-mark.png">

    <style>
        body{
            background-image: url("https://img.freepik.com/free-photo/solid-navy-blue-concrete-textured-wall_53876-124584.jpg");
            background-size: cover;
            background-repeat: repeat-y;
            background-attachment: fixed;
```

```
}

.contaner{
    display: flex;
    flex-direction: column;
    align-items: center;
}

.main{
    background-color: azure;
    border-radius: 20px;
    width: 98%;
    margin:70px auto;
    padding: 10px;
}

.title{
    text-align: center;
}

.subtitle{
    font-style: italic;
    text-decoration: underline;
}

p{
    text-indent: 20px;
}

#up{
    position: fixed;
    left: 95%;
    top: 93%;
}
.samples img{
    margin-left: 50px;
}

.links a{
    text-decoration: none;
    font-weight: bold;
    font-style: italic;
}

.links a::before{
    content: "• ";
}

.links{
    display: flex;
    flex-direction: column;
```

```
        }

    </style>

</head>
<body>
    <div class="container">
        <div class="main">
            <h2 class="title">Справка пользователя по использованию приложения для работы с базой данных</h2>
            <h1 class="title">"Компьютерный магазин <i>Centurion</i>"</h1>

            <hr width="98%">

            <div class="links">
                <a href="#for_what">Для чего эта программа</a>
                <a href="#how_to">Как пользоваться</a>
                <a href="#roles">Роли</a>
                <a href="#troubles">Проблемы</a>
                <a href="#feedback">FeedBack</a>

            </div>

            <hr width="98%">

            <a id="up" href="#"><img src="https://cdn.pixabay.com/photo/2012/04/02/15/49/button-24807_1280.png" width="60%" alt="Up"></a>

            <h2 id="for_what" class="subtitle">Для чего это программа</h2>
            <p>
                Эта программа была разработана для работы с базой данных "Компьютерный магазин Centurion". Она предназначена для управления данной базой данных, а сама база данных, в свою очередь предназначена для облегчения ведения учета продажи техники в компьютерном магазине.
            </p>


            <h2 id="how_to" class="subtitle">Как пользоваться</h2>
            <p>
                При помощи данного приложения вы можете просматривать содержимое БД, изменять и удалять информацию из неё. Есть возможность зайти в настройки и поменять тему приложения, следать отчеты а также распределить роли, сделать резервную копию БД и восстановить её из файла Бэкапа...
                <div class="samples">
                    <h3>Вход</h3>
                        <img src="./images/1.png" alt="Вход">
                    <h3>Регистрация</h3>
                        <img src="./images/2.png" alt="Регистрация">
                    <h3>Просмотр, редактирование и удаление</h3>
                        <img src="./images/3.png" alt="Просмотр, редактирование и удаление">
                    <h3>Настройки</h3>
                        <img src="./images/4.png" alt="Настройки">
                    <h3>Отчеты</h3>
```

```
                    <img src="./images/5.png" alt="Отчеты">
                <h3>Администрирование</h3>
                    <img src="./images/6.png" alt="Администрирование">
            </div>
        </p>

        <h2 id="roles" class="subtitle">Распределение ролей</h2>
        <p>
            Обратите внимание на то, что не все функции доступны обычному пользователю. Так, например обычный пользователь не может изменять или вносить значения в БД или назначать роли пользователям, это могут делать только пользователи, чьи роли БД позволяют это делать.
        </p>
        <p>
            Предусмотрены следующие роли:
            <ul>
                <li>Пользователь</li>
                <li>Продавец</li>
                <li>Мэнеджер</li>
                <li>Администратор</li>
            </ul>

            Для повышения вашей роли обратитесь к администратору...
        </p>

        <h2 id="troubles" class="subtitle">Проблемы</h2>
        <p>
            В случае каких-либо неполадок, или ошибок, совершенных вами, обратитесь к вашему администратору. В противном случае, если ошибки более серьёзного характера, сообщите администратору, который в обязательном порядке свяжется с создателем БД, который сможет её решить. Вы также можето позвонить на горячую линию производителя, и мы постараемся вам помочь...
<br>Также, если вы забыли пароль, вы можете его восстановить...
        </p>

        <h2 id="feedback" class="subtitle">FeedBack</h2>
        <p style="text-indent: 0px;">
            <b>Разработчик:</b> MicroBot industries<sup>©</sup><br>
            <b>Горячая линия: </b> +(373)69982418<br>
            <b>Наша почта: </b> micro_bot@gmail.com<br>
            <b>Адрес: </b> str. Stefan the Great 22/8<br>
            <div style="text-align: center;"><sub><b>Спасибо за приобретение нашего ПО!</b></sub></div>
        </p>


    </div>
</div>
</body>
</html>
```

Вывод

Общие впечатления от практики

Подводя итоги практики, ч могу сказать, что она мне очень сильно понравилась. Всё было замечательно, работа спорилась, а если возникали какие-либо вопросы, можно было задать их преподавателю. Разрабатывать данное мне приложение было безумно интересно, а сама практика, несомненно, пойдет мне на пользу, так как это бесценный опыт, который мне пригодится в будущем…

Что понравилось

Преподаватель отзывчивый и пунктуальный. Никогда не отказывает в помощи, может дать необходимые для проекта советы, может указать на недочеты.

Что не понравилось

Класс был немного шумным.

Какие трудности встретил

В процессе проектирования БД, пришлось вспоминать нормализацию, что мне вполне удалось… Также трудности возникли с составлением инсталлятора приложения, однако в конце концов, мне всё же удалось найти решение…

Что помогло

Интернет, Книги, Видео из YouTube-а и, конечно же советы преподавателя…

Какие есть предложения по практике

Мне всё понравилось, я не могу выдвинуть какие-либо предложения… Всё было здорово!!!

Библиография

- Васильев, А. (б.д.). *Программирование на C# для начинающих. Основные сведения.*
- Культин, Н. (б.д.). *C# в задачах и примерах.*
- https://metanit.com/sharp/adonetcore/
- https://www.youtube.com/watch?v=KyFWqbRfWIA&list=PLQOaTSbfxUtD6kMmAYc8Fooqya3pjLs1N
- https://www.w3schools.com/cs/index.php

Приложение

RegistrationForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class RegistrationForm : Form
    {
        Form caller;
        string mode = "signIn";
        public RegistrationForm(Form caller)
        {
            InitializeComponent();
            this.caller = caller;

            if(((Form1)caller).getStyle() == "white")
            {
                this.BackColor = Color.WhiteSmoke;

                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                label7.ForeColor = Color.Black;
                label9.ForeColor = Color.Black;
                button1.ForeColor = Color.Black;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);

                label1.ForeColor = Color.WhiteSmoke;
                label2.ForeColor = Color.WhiteSmoke;
                label3.ForeColor = Color.WhiteSmoke;
                label5.ForeColor = Color.WhiteSmoke;
                label6.ForeColor = Color.WhiteSmoke;
                label7.ForeColor = Color.WhiteSmoke;
                label9.ForeColor = Color.WhiteSmoke;
                button1.ForeColor = Color.WhiteSmoke;
            }
        }

        private void RegistrationForm_FormClosed(object sender, FormClosedEventArgs e)
        {
            caller.Close();
        }

        private void label4_Click(object sender, EventArgs e)
        {
            if (mode == "signIn")
            {
                mode = "signUp";
                label5.Text = "Sign Up";
                label3.Text = "Already have an acount?";
                label4.Text = "Sign In";
```

```
                button1.Text = "Sign Up";
                label6.Visible = true;
                textBox2.Visible = true;
                label7.Visible= true;
                textBox4.Visible = true;
                label8.Visible = false;
                label9.Visible = false;

                textBox1.Text = "";
                textBox2.Text = "";
                textBox3.Text = "";
                textBox4.Text = "";

            }
            else {
                mode = "signIn";
                label5.Text = "Sign In";
                label3.Text = "Don't have an acount?";
                label4.Text = "Sign Up";
                button1.Text = "Sign In";
                label6.Visible = false;
                textBox2.Visible = false;
                label7.Visible = false;
                textBox4.Visible = false;
                label8.Visible = false;
                label9.Visible = false;

                textBox1.Text = "";
                textBox2.Text = "";
                textBox3.Text = "";
                textBox4.Text = "";
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (mode == "signIn")
                {
                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from Users where user_login
=  @user_login collate Latin1_General_CS_AI", connection);
                        command.Parameters.Add("@user_login", textBox3.Text);

                        SqlDataReader reader = command.ExecuteReader();

                        //MessageBox.Show(reader.HasRows.ToString());

                        if (reader.HasRows)
                        {
                            reader.Read();
                            //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                            if(textBox1.Text == reader.GetValue(1).ToString())
                            {
                                //MessageBox.Show(reader.GetValue(0).ToString() + " " +
reader.GetValue(1).ToString() + " " + reader.GetValue(2).ToString());
                                ((Form1)caller).setUser(reader.GetValue(2).ToString());
                                ((Form1)caller).setUserLogin(reader.GetValue(0).ToString());
                                caller.Show();
                                this.Hide();
                            }
                            else
                            {
```

```
                                MessageBox.Show("Wrong password!", "Wrong password!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                label8.Visible = true;
                                label9.Visible = true;
                            }
                        }
                        else
                        {
                            MessageBox.Show("Such user has not been found!", "User no found",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }

                        reader.Close();
                    }
                }
                else
                {
                    if(textBox3.Text != "")
                    {
                        if (textBox1.Text.Length >= 5)
                        {
                            if (textBox1.Text == textBox2.Text)
                            {
                                if (Regex.IsMatch(textBox4.Text, "^.+\\@.+\\..+$"))
                                {
                                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                                    {
                                        connection.Open();

                                        SqlCommand command = new SqlCommand("select * from Users
where user_login = @user_login collate Latin1_General_CS_AI", connection);
                                        command.Parameters.Add("@user_login", textBox3.Text);

                                        SqlDataReader reader = command.ExecuteReader();

                                        //MessageBox.Show(reader.HasRows.ToString());

                                        if (!reader.HasRows)
                                        {
                                            reader.Close();

                                            command.CommandText = "select * from Users where
user_email = @user_email";
                                            command.Parameters.Clear();
                                            command.Parameters.Add("user_email", textBox4.Text);

                                            reader = command.ExecuteReader();

                                            if (!reader.HasRows)
                                            {
                                                reader.Close();

                                                command.CommandText = "insert into Users values
(@user_login, @user_password, 'user', @user_email)";
                                                command.Parameters.Clear();
                                                command.Parameters.Add("@user_login",
textBox3.Text);
                                                command.Parameters.Add("@user_password",
textBox1.Text);
                                                command.Parameters.Add("@user_email",
textBox4.Text);

                                                if (command.ExecuteNonQuery() > 0)
                                                {
                                                    ((Form1)caller).setUser("user");
                                                    ((Form1)caller).setUserLogin(textBox3.Text);
                                                    this.Hide();
                                                    caller.Show();
```

```
                                                }
                                                else
                                                {
                                                    MessageBox.Show("An error occured!", "User
adding", MessageBoxButtons.OK, MessageBoxIcon.Error);
                                                }
                                            }
                                            else
                                            {
                                                MessageBox.Show("This email is already used!",
"Unavailable email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                            }
                                        }
                                        else
                                        {
                                            MessageBox.Show("Such user already exists!", "User
exists", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                        }

                                        reader.Close();
                                    }
                                }
                                else
                                {
                                    MessageBox.Show("Please enter the email in the folowing
format: example@gmail.com!", "Wrong email", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                }
                            }
                            else
                            {
                                MessageBox.Show("Passwords don't match!", "Passwords don't
match", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                        }
                        else
                        {
                            MessageBox.Show("Password should contain at least 5 chars!",
"Password's to short", MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    }
                    else
                    {
                        MessageBox.Show("Enter User name!", "Empty user name",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }

                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void label8_Click(object sender, EventArgs e)
        {
            PasswordRestoreFrorm passwordRestoreFrorm = new PasswordRestoreFrorm(textBox3.Text,
((Form1)caller).getStyle(), this);
            passwordRestoreFrorm.ShowDialog();
        }



        public void setLogin(String login)
        {
            textBox3.Text = login;
        }
```

```
        public void setPassword(String password)
        {
            textBox1.Text = password;
        }
    }
}
```

PasswordRestoreForm.cs

```
using System;
using System.Data.SqlClient;
using System.Drawing;
using System.Net;
using System.Net.Mail;
using System.Reflection.Emit;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Computer_magazine
{
    public partial class PasswordRestoreFrorm : Form
    {

        private int code;
        private string oldPassword;
        private string email;
        private string login;
        Form caller;
        public PasswordRestoreFrorm(String login, String style, Form caller)
        {
            InitializeComponent();

            this.caller = caller;

            label3.Text += "\"" + login + "\"";

            if (style.Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;

                label1.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);

                label1.ForeColor = Color.WhiteSmoke;
                label3.ForeColor = Color.WhiteSmoke;
                label4.ForeColor = Color.WhiteSmoke;
                label5.ForeColor = Color.WhiteSmoke;
            }

            this.login = login;

            try
            {
                using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                {
                    connection.Open();

                    String query = "select * from Users where user_login = @user_login collate
Latin1_General_CS_AI";

                    SqlCommand command = new SqlCommand(query, connection);
                    command.Parameters.Add("@user_login", login);
```

```
                    SqlDataReader reader = command.ExecuteReader();

                    if (reader.HasRows)
                    {
                        reader.Read();
                        oldPassword = reader.GetValue(1).ToString();
                        email = reader.GetValue(3).ToString();

                        reader.Close();
                        command.Dispose();


                        sendCodeMessage();
                    }
                    else
                    {
                        MessageBox.Show($"User not found!", "User not found",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
            }
            catch(Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (!oldPassword.Equals(textBox2.Text))
            {
                if (textBox2.Text.Length >= 5)
                {

                    if (textBox2.Text.Equals(textBox3.Text))
                    {
                        try
                        {
                            using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                            {
                                connection.Open();

                                String query = "update Users set user_password = @user_passord
where user_login = @user_login";

                                SqlCommand command = new SqlCommand(query, connection);
                                command.Parameters.Add("@user_passord", textBox2.Text);
                                command.Parameters.Add("@user_login", login);

                                //auto regiistration
                                ((RegistrationForm)caller).setLogin(login);
                                ((RegistrationForm)caller).setPassword(textBox2.Text);

                                command.ExecuteNonQuery();

                                MessageBox.Show("Password successfully changed!", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                sendResultMessage();
                                this.Close();
                            }
                        }
                        catch (Exception ex)
                        {
                            MessageBox.Show($"Error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
                    }
                }
                else
                {
                    MessageBox.Show($"Passwords are not matching!", "Invalind confirmation",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
            }
            else
            {
                MessageBox.Show($"Passwords should be 5 or more symbols!", "Short password",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
        else
        {
            MessageBox.Show($"Enter new password!", "New Password", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (textBox1.Text.Equals(code.ToString())){
            textBox1.Visible= false;
            button1.Visible= false;
            label1.Visible= false;

            label3.Visible= true;
            label4.Visible= true;
            label5.Visible= true;

            textBox2.Visible= true;
            textBox3.Visible= true;

            button2.Visible= true;
        }
        else
        {
            MessageBox.Show($"Verification code is incorrect! Resanding...", "Wrong
Verification code", MessageBoxButtons.OK, MessageBoxIcon.Information);
            sendCodeMessage();
            textBox1.Text = "";
        }
    }



    private void sendCodeMessage()
    {
        Random randCode = new Random();
        code = randCode.Next(1000, 9999);

        //Создаем письмо
        MailAddress from = new MailAddress("microbot2033@gmail.com", "MicroBotSupport");
        //MailAddress to = new MailAddress("botannicolai22@gmail.com", login);
        MailAddress to = new MailAddress(email, login);
        MailMessage message = new MailMessage(from, to);

        message.Subject = "Restoring your passsword";
        message.Body = $"<h1>We are restoring your password now</h1><br><br><b>Dear
{login}!</b><br>To continue restoring the password enter the following code in your App:
<b>{code}</b>";
        message.IsBodyHtml = true;


        //Создаем клиента (не пользователя), через который мы будем отправлять письмо
        SmtpClient client = new SmtpClient("smtp.gmail.com", 587); //smtp.gmail.com - сервис
Gmail; 587 - порт для писем
```

```
            client.Credentials = new NetworkCredential("microbot2033@gmail.com",
"gohxzhvswujhezie"); //тут указываем почту, с которой будем отправлять и указываем пароль.
!ACHTUNG! Это пароль - это не простой пароль от почты, а пароль приложения. Его можно создать в
настройках своего Гугл аккаунта. Просто в строке поиска пишешь "Пароли приложений" -> из
выпадающего списсска с приложения ми выбираешь "другое", пишешь имя и копируешь пароль в
оранджевом окошке. Вот его и надо использовать...

     //Подробнее о том, как это сделать смотри тут: https://www.youtube.com/watch?v=tu9QS_t5o1k

     //Ранние методы не рабочие, так как поилитики безопасности Google изменились!!!
            client.EnableSsl = true; //Для безопасного подключения
            client.Send(message);

            //MessageBox.Show("Sent!");
        }

        private void sendResultMessage()
        {
            MailAddress from = new MailAddress("microbot2033@gmail.com", "MicroBotSupport");
            MailAddress to = new MailAddress(email, login);
            MailMessage message = new MailMessage(from, to);

            message.Subject = "Success!";
            message.Body = $"<h1>Password Succcessfully changed!</h1><br><br>Dear
{login}!<br>Wour password has been successfully changed to <b>{textBox2.Text}</b>...<br><br>Have
a good day!";
            message.IsBodyHtml = true;

            SmtpClient client = new SmtpClient("smtp.gmail.com", 587);
            client.EnableSsl= true;
            client.Credentials = new NetworkCredential("microbot2033@gmail.com",
"gohxzhvswujhezie");

            client.Send(message);
        }
    }
}
```

Form1.cs

```
using Microsoft.ReportingServices.ReportProcessing.ReportObjectModel;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Common;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Net.Mail;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class Form1 : Form
    {
        private string cfg = "../../config/config.cfg";
        Dictionary<string, string> config;
        private string table;
        private ErrorProvider error = new ErrorProvider();
        private string user;
```

```
private string user_login;

Computer_magazineDataSet ds;

private string style;

public Form1()
{
    InitializeComponent();


    config = new Dictionary<string, string>();


    //reading saved config
    using (StreamReader file = new StreamReader(cfg))
    {
        string[] lines = file.ReadToEnd().Split('\n');

        foreach(string line in lines)
        {
            //MessageBox.Show(line);
            if(line != "")
                config.Add((line.Split(':'))[0], (line.Split(':'))[1]);
        }
    }

    if (config.ContainsKey("style"))
    {
        //style = config["style"];
        config.TryGetValue("style", out style);
        style = style.Replace("\r", ""); //\r - perenos karetki

        //MessageBox.Show(style);

        if (style.Equals("white"))
        {
            this.BackColor = Color.WhiteSmoke;
            dataGridView1.BackgroundColor= Color.WhiteSmoke;
            menuStrip1.BackColor= Color.Gainsboro;
            panel1.BackColor= Color.LightGray;
            panel2.BackColor= Color.LightGray;
            panel3.BackColor= Color.LightGray;
            tablesToolStripMenuItem.BackColor= Color.LightGray;
            settingsToolStripMenuItem.BackColor= Color.LightGray;
            reportsToolStripMenuItem.BackColor= Color.LightGray;
            rolesToolStripMenuItem.BackColor = Color.LightGray;

            label1.ForeColor= Color.Black;
            label2.ForeColor= Color.Black;
            label3.ForeColor= Color.Black;
            label4.ForeColor= Color.Black;
            label5.ForeColor= Color.Black;
            label6.ForeColor= Color.Black;
            label7.ForeColor= Color.Black;
            checkBox1.ForeColor= Color.Black;
            label8.ForeColor= Color.Black;
            label9.ForeColor= Color.Black;
            label10.ForeColor= Color.Black;
            label11.ForeColor= Color.Black;
            label12.ForeColor= Color.Black;
            label13.ForeColor= Color.Black;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            menuStrip1.BackColor = Color.DimGray;
            panel1.BackColor = Color.Gray;
```

```
                panel2.BackColor = Color.Gray;
                panel3.BackColor = Color.Gray;
                tablesToolStripMenuItem.BackColor = Color.Gray;
                settingsToolStripMenuItem.BackColor = Color.Gray;
                reportsToolStripMenuItem.BackColor = Color.Gray;
                rolesToolStripMenuItem.BackColor = Color.Gray;

                label1.ForeColor = Color.WhiteSmoke;
                label2.ForeColor = Color.WhiteSmoke;
                label3.ForeColor = Color.WhiteSmoke;
                label4.ForeColor = Color.WhiteSmoke;
                label5.ForeColor = Color.WhiteSmoke;
                label6.ForeColor = Color.WhiteSmoke;
                label7.ForeColor = Color.WhiteSmoke;
                checkBox1.ForeColor = Color.WhiteSmoke;
                label8.ForeColor = Color.WhiteSmoke;
                label9.ForeColor = Color.WhiteSmoke;
                label10.ForeColor = Color.WhiteSmoke;
                label11.ForeColor = Color.WhiteSmoke;
                label12.ForeColor = Color.WhiteSmoke;
                label13.ForeColor = Color.WhiteSmoke;
            }
        }


        comboBox1.Items.Add("Consign.№");
        comboBox1.Items.Add("Date");

        //ds = new Computer_magazineDataSet();

        //string connection_string = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

        //using (SqlConnection connection = new SqlConnection(connection_string))
        //{
        //    connection.Open();

        //    string query = "select * from consignment_view";

        //    SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);

        //    dataAdapter.Fill(ds.Tables[0]);


        //    query = "select * from Chek_view";

        //    dataAdapter.SelectCommand = new SqlCommand(query, connection);

        //    dataAdapter.Fill(ds.Tables[1]);

        //    query = "select * from Product_view";

        //    dataAdapter.SelectCommand = new SqlCommand(query, connection);

        //    dataAdapter.Fill(ds.Tables[2]);



        //    //// перебор всех таблиц
        //    //foreach (DataTable dt in ds.Tables)
        //    //{
        //    //    Console.WriteLine(dt.TableName); // название таблицы
        //    //                                     // перебор всех столбцов
        //    //    foreach (DataColumn column in dt.Columns)
        //    //        Console.Write("\t{0}", column.ColumnName);
        //    //    Console.WriteLine();
        //    //    // перебор всех строк таблицы
        //    //    foreach (DataRow row in dt.Rows)
        //    //    {
```

```
    //    //        // получаем все ячейки строки
    //    //        var cells = row.ItemArray;
    //    //        foreach (object cell in cells)
    //    //            Console.Write("\t{0}", cell);
    //    //        Console.WriteLine();
    //    //    }
    //    //}

    //}

    helpProvider1.HelpNamespace = "../../help/help.html";
}

private void Form1_Load(object sender, EventArgs e)
{
    RegistrationForm rf = new RegistrationForm(this);
    this.Hide();
    rf.ShowDialog();


    if (user == "user")
    {
        editToolStripMenuItem.Visible = false;
        editToolStripMenuItem1.Visible = false;
        editToolStripMenuItem2.Visible = false;
        editToolStripMenuItem3.Visible = false;
        editToolStripMenuItem4.Visible = false;
        editToolStripMenuItem5.Visible = false;
        editToolStripMenuItem6.Visible = false;
        editToolStripMenuItem7.Visible = false;
        editToolStripMenuItem8.Visible = false;
        editToolStripMenuItem9.Visible = false;
        editToolStripMenuItem10.Visible = false;
        editToolStripMenuItem11.Visible = false;
        editToolStripMenuItem12.Visible = false;

        chekToolStripMenuItem.Visible = false;
        consignmentToolStripMenuItem.Visible = false;
        chekInfoToolStripMenuItem.Visible = false;
        pasportDataToolStripMenuItem.Visible = false;


        addToolStripMenuItem.Visible = false;
        toolStripMenuItem1.Visible = false;
        toolStripMenuItem2.Visible = false;
        toolStripMenuItem3.Visible = false;
        toolStripMenuItem4.Visible = false;
        toolStripMenuItem5.Visible = false;
        toolStripMenuItem6.Visible = false;
        toolStripMenuItem7.Visible = false;
        toolStripMenuItem8.Visible = false;
        toolStripMenuItem9.Visible = false;
        toolStripMenuItem10.Visible = false;
        toolStripMenuItem11.Visible = false;
        addToolStripMenuItem1.Visible = false;

        reportsToolStripMenuItem.Visible = false;

        rolesToolStripMenuItem.Visible = false;
    }

    if (user == "admin")
    {
        editToolStripMenuItem.Visible = true;
        editToolStripMenuItem1.Visible = true;
        editToolStripMenuItem2.Visible = true;
        editToolStripMenuItem3.Visible = true;
        editToolStripMenuItem4.Visible = true;
        editToolStripMenuItem5.Visible = true;
```

```
        editToolStripMenuItem6.Visible = true;
        editToolStripMenuItem7.Visible = true;
        editToolStripMenuItem8.Visible = true;
        editToolStripMenuItem9.Visible = true;
        editToolStripMenuItem10.Visible = true;
        editToolStripMenuItem11.Visible = true;
        editToolStripMenuItem12.Visible = true;

        addToolStripMenuItem.Visible = true;
        toolStripMenuItem1.Visible = true;
        toolStripMenuItem2.Visible = true;
        toolStripMenuItem3.Visible = true;
        toolStripMenuItem4.Visible = true;
        toolStripMenuItem5.Visible = true;
        toolStripMenuItem6.Visible = true;
        toolStripMenuItem7.Visible = true;
        toolStripMenuItem8.Visible = true;
        toolStripMenuItem9.Visible = true;
        toolStripMenuItem10.Visible = true;
        toolStripMenuItem11.Visible = true;
        addToolStripMenuItem1.Visible = true;

        reportsToolStripMenuItem.Visible = true;

        rolesToolStripMenuItem.Visible = true;
    }

    if (user == "seller")
    {
        editToolStripMenuItem.Visible = false;
        editToolStripMenuItem1.Visible = false;
        editToolStripMenuItem2.Visible = false;
        editToolStripMenuItem3.Visible = false;
        editToolStripMenuItem4.Visible = false;
        editToolStripMenuItem5.Visible = false;
        editToolStripMenuItem6.Visible = false;
        editToolStripMenuItem7.Visible = false;
        editToolStripMenuItem8.Visible = false;
        editToolStripMenuItem9.Visible = false;
        editToolStripMenuItem10.Visible = false;
        editToolStripMenuItem11.Visible = false;
        editToolStripMenuItem12.Visible = false;

        addToolStripMenuItem.Visible = false;
        toolStripMenuItem1.Visible = false;
        toolStripMenuItem2.Visible = false;
        toolStripMenuItem3.Visible = false;
        toolStripMenuItem4.Visible = false;
        toolStripMenuItem5.Visible = false;
        toolStripMenuItem6.Visible = false;
        toolStripMenuItem7.Visible = false;
        toolStripMenuItem8.Visible = true;
        toolStripMenuItem9.Visible = false;
        toolStripMenuItem10.Visible = true;
        toolStripMenuItem11.Visible = false;
        addToolStripMenuItem1.Visible = true;

        reportsToolStripMenuItem.Visible = false;

        rolesToolStripMenuItem.Visible = false;
    }

    if (user == "manager")
    {
        editToolStripMenuItem.Visible = false;
        editToolStripMenuItem1.Visible = false;
        editToolStripMenuItem2.Visible = false;
        editToolStripMenuItem3.Visible = false;
        editToolStripMenuItem4.Visible = false;
```

```
            editToolStripMenuItem5.Visible = false;
            editToolStripMenuItem6.Visible = false;
            editToolStripMenuItem7.Visible = false;
            editToolStripMenuItem8.Visible = true;
            editToolStripMenuItem9.Visible = false;
            editToolStripMenuItem10.Visible = true;
            editToolStripMenuItem11.Visible = false;
            editToolStripMenuItem12.Visible = true;

            addToolStripMenuItem.Visible = false;
            toolStripMenuItem1.Visible = false;
            toolStripMenuItem2.Visible = false;
            toolStripMenuItem3.Visible = false;
            toolStripMenuItem4.Visible = false;
            toolStripMenuItem5.Visible = false;
            toolStripMenuItem6.Visible = false;
            toolStripMenuItem7.Visible = false;
            toolStripMenuItem8.Visible = true;
            toolStripMenuItem9.Visible = true;
            toolStripMenuItem10.Visible = true;
            toolStripMenuItem11.Visible = true;
            addToolStripMenuItem1.Visible = true;

            reportsToolStripMenuItem.Visible = true;

            rolesToolStripMenuItem.Visible = false;
        }
    }

    public string getStyle()
    {
        return style;
    }

    public void setStyle(string style)
    {
        config.Remove("style");
        config.Add("style", style);

        using (StreamWriter file = new StreamWriter(cfg))
        {
            foreach(string key in config.Keys)
            {
                config.TryGetValue(key, out string value);
                file.WriteLine(key+":"+value);
            }
        }

        this.style = style;

        if (style == "white")
        {
            this.BackColor = Color.WhiteSmoke;
            dataGridView1.BackgroundColor= Color.WhiteSmoke;
            menuStrip1.BackColor = Color.Gainsboro;
            panel1.BackColor = Color.LightGray;
            panel2.BackColor = Color.LightGray;
            panel3.BackColor = Color.LightGray;
            tablesToolStripMenuItem.BackColor = Color.LightGray;
            settingsToolStripMenuItem.BackColor = Color.LightGray;
            reportsToolStripMenuItem.BackColor = Color.LightGray;
            rolesToolStripMenuItem.BackColor = Color.LightGray;

            label1.ForeColor = Color.Black;
            label2.ForeColor = Color.Black;
            label3.ForeColor = Color.Black;
            label4.ForeColor = Color.Black;
            label5.ForeColor = Color.Black;
            label6.ForeColor = Color.Black;
```

```
            label7.ForeColor = Color.Black;
            checkBox1.ForeColor = Color.Black;

            label8.ForeColor = Color.Black;
            label9.ForeColor = Color.Black;
            label10.ForeColor = Color.Black;
            label11.ForeColor = Color.Black;
            label12.ForeColor = Color.Black;
            label13.ForeColor = Color.Black;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            menuStrip1.BackColor = Color.DimGray;
            panel1.BackColor = Color.Gray;
            panel2.BackColor = Color.Gray;
            panel3.BackColor = Color.Gray;
            tablesToolStripMenuItem.BackColor = Color.Gray;
            settingsToolStripMenuItem.BackColor = Color.Gray;
            reportsToolStripMenuItem.BackColor = Color.Gray;
            rolesToolStripMenuItem.BackColor = Color.Gray;

            label1.ForeColor = Color.WhiteSmoke;
            label2.ForeColor = Color.WhiteSmoke;
            label3.ForeColor = Color.WhiteSmoke;
            label4.ForeColor = Color.WhiteSmoke;
            label5.ForeColor = Color.WhiteSmoke;
            label6.ForeColor = Color.WhiteSmoke;
            label7.ForeColor = Color.WhiteSmoke;
            checkBox1.ForeColor = Color.WhiteSmoke;

            label8.ForeColor = Color.WhiteSmoke;
            label9.ForeColor = Color.WhiteSmoke;
            label10.ForeColor = Color.WhiteSmoke;
            label11.ForeColor = Color.WhiteSmoke;
            label12.ForeColor = Color.WhiteSmoke;
            label13.ForeColor = Color.WhiteSmoke;
        }
    }



    public string getUser()
    {
        return this.user;
    }

    public void setUser(string user)
    {
        this.user = user;
    }

    public string getUserLogin()
    {
        return this.user_login;
    }

    public void setUserLogin(String user_login)
    {
        this.user_login = user_login;
    }



    private void styleToolStripMenuItem_Click(object sender, EventArgs e)
    {
        StyleForm style = new StyleForm(this);
        style.ShowDialog();
```

```
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    updateManufacturer();
}

private void openToolStripMenuItem1_Click(object sender, EventArgs e)
{
    updateSupplier();
}

private void openToolStripMenuItem2_Click(object sender, EventArgs e)
{
    updateMonitor();
}

private void openToolStripMenuItem3_Click(object sender, EventArgs e)
{
    updatePhone();
}

private void openToolStripMenuItem4_Click(object sender, EventArgs e)
{
    updateComputer();
}

private void openToolStripMenuItem5_Click(object sender, EventArgs e)
{
    updatePrinter();
}

private void openToolStripMenuItem6_Click(object sender, EventArgs e)
{
    updateProductType();
}

private void openToolStripMenuItem7_Click(object sender, EventArgs e)
{
    updatePayment();
}

private void openToolStripMenuItem8_Click(object sender, EventArgs e)
{
    updateChek();
}

private void openToolStripMenuItem9_Click(object sender, EventArgs e)
{
    updateConsignment();
}

private void openToolStripMenuItem10_Click(object sender, EventArgs e)
{
    updateChekInfoChek();
}

private void openToolStripMenuItem11_Click(object sender, EventArgs e)
{
    updateProduct();
}

private void openToolStripMenuItem12_Click(object sender, EventArgs e)
{
    updatePasportData();
}


private void addToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    ManEditForm add = new ManEditForm(this, "add");
    add.ShowDialog();
    updateManufacturer();
}
private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    ManEditForm edit = new ManEditForm(this, "edit");
    edit.ShowDialog();
    updateManufacturer();
}



private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    SupEditForm add = new SupEditForm(this, "add");
    add.ShowDialog();
    updateSupplier();
}

private void editToolStripMenuItem1_Click(object sender, EventArgs e)
{
    SupEditForm edit = new SupEditForm(this, "edit");
    edit.ShowDialog();
    updateSupplier();
}



private void toolStripMenuItem2_Click(object sender, EventArgs e)
{
    MonitorEditForm add = new MonitorEditForm(this, "add");
    add.ShowDialog();
    updateMonitor();
}

private void editToolStripMenuItem2_Click(object sender, EventArgs e)
{
    MonitorEditForm edit = new MonitorEditForm(this, "edit");
    edit.ShowDialog();
    updateMonitor();
}



private void toolStripMenuItem3_Click(object sender, EventArgs e)
{
    PhoneEditForm add = new PhoneEditForm(this, "add");
    add.ShowDialog();
    updatePhone();
}

private void editToolStripMenuItem3_Click(object sender, EventArgs e)
{
    PhoneEditForm edit = new PhoneEditForm(this, "edit");
    edit.ShowDialog();
    updatePhone();
}



private void toolStripMenuItem4_Click(object sender, EventArgs e)
{
    ComputerEditForm add = new ComputerEditForm(this, "add");
    add.ShowDialog();
    updateComputer();
}
```

```
private void editToolStripMenuItem4_Click(object sender, EventArgs e)
{
    ComputerEditForm edit = new ComputerEditForm(this, "edit");
    edit.ShowDialog();
    updateComputer();
}



private void toolStripMenuItem5_Click(object sender, EventArgs e)
{
    PrinterEditForm add = new PrinterEditForm(this, "add");
    add.ShowDialog();
    updatePrinter();
}

private void editToolStripMenuItem5_Click(object sender, EventArgs e)
{
    PrinterEditForm edit = new PrinterEditForm(this, "edit");
    edit.ShowDialog();
    updatePrinter();
}



private void toolStripMenuItem6_Click(object sender, EventArgs e)
{
    ProductTypeEditForm add = new ProductTypeEditForm(this, "add");
    add.ShowDialog();
    updateProductType();
}

private void editToolStripMenuItem6_Click(object sender, EventArgs e)
{
    ProductTypeEditForm edit = new ProductTypeEditForm(this, "edit");
    edit.ShowDialog();
    updateProductType();
}



private void toolStripMenuItem7_Click(object sender, EventArgs e)
{
    PaymentEditForm add = new PaymentEditForm(this, "add");
    add.ShowDialog();
    updatePayment();
}

private void editToolStripMenuItem7_Click(object sender, EventArgs e)
{
    PaymentEditForm edit = new PaymentEditForm(this, "edit");
    edit.ShowDialog();
    updatePayment();
}



private void toolStripMenuItem8_Click(object sender, EventArgs e)
{
    ChekEditForm add = new ChekEditForm(this, "add");
    add.ShowDialog();
    updateChek();
}

private void editToolStripMenuItem8_Click(object sender, EventArgs e)
{
    if (user == "manager")
    {
        ChekEditForm edit = new ChekEditForm(this, "edit-only");
```

```
            edit.ShowDialog();
            updateChek();
        }
        else
        {
            ChekEditForm edit = new ChekEditForm(this, "edit");
            edit.ShowDialog();
            updateChek();
        }

    }



    private void toolStripMenuItem9_Click(object sender, EventArgs e)
    {
        ConsignmentEditForm add = new ConsignmentEditForm(this, "add");
        add.ShowDialog();
        updateConsignment();
    }

    private void editToolStripMenuItem9_Click(object sender, EventArgs e)
    {
        ConsignmentEditForm edit = new ConsignmentEditForm(this, "edit");
        edit.ShowDialog();
        updateConsignment();
    }



    private void toolStripMenuItem10_Click(object sender, EventArgs e)
    {
        ChekInfoChekEditForm add = new ChekInfoChekEditForm(this, "add");
        add.ShowDialog();
        updateChekInfoChek();
    }

    private void editToolStripMenuItem10_Click(object sender, EventArgs e)
    {
        if (user == "manager")
        {
            ChekInfoChekEditForm edit = new ChekInfoChekEditForm(this, "edit-only");
            edit.ShowDialog();
            updateChekInfoChek();
        }
        else
        {
            ChekInfoChekEditForm edit = new ChekInfoChekEditForm(this, "edit");
            edit.ShowDialog();
            updateChekInfoChek();
        }


    }



    private void toolStripMenuItem11_Click(object sender, EventArgs e)
    {
        ProductEditForm add = new ProductEditForm(this, "add");
        add.ShowDialog();
        updateProduct();
    }

    private void editToolStripMenuItem11_Click(object sender, EventArgs e)
    {
        ProductEditForm edit = new ProductEditForm(this, "edit");
        edit.ShowDialog();
        updateProduct();
```

```
        }



        private void updateManufacturer()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
            table = "manufacturer";
            //search
            label1.Visible= true;
            label2.Visible= false;
            textBox1.Visible= false;
            button1.Visible= false;
            button2.Visible= false;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;

            //filter
            label3.Visible=true;
            label4.Visible = false;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox2.Items.Clear();
            comboBox2.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            button3.Visible = false;
            button4.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;

            //calculs

            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;

            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
```

```
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from manufacturer";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 4;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Manufacturer id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Manufacturer name";
                    dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Production price";
                    if (user == "user") dataGridView1.Columns[0].Visible = false;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        public void updateSupplier()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
            //searchs
            label1.Visible = true;
            label2.Visible = true;
            label2.Text = "Telephone: ";
            table = "supplier";
            textBox1.Visible = true;
            button1.Visible = true;
            button2.Visible = true;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;


            //filter
            label3.Visible = true;
            label4.Visible = true;
            label4.Text = "Prod. type:";
            comboBox2.Visible = true;
            comboBox2.Items.Clear();
            comboBox2.Items.Add("all");
            comboBox2.Items.Add("PC");
            comboBox2.Items.Add("Laptop");
```

```
            comboBox2.Items.Add("Monitor");
            comboBox2.Items.Add("Phone");
            comboBox2.Items.Add("Printer");
            comboBox2.Items.Add("Tablet");
            comboBox2.SelectedIndex = 0;
            button3.Visible = true;
            button4.Visible = true;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;


            //calculs


            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;



            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from supplier";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
```

```
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                dataGridView1.Columns[1].HeaderCell.Value = "Supplier name";
                dataGridView1.Columns[2].HeaderCell.Value = "Supplier adres";
                dataGridView1.Columns[3].HeaderCell.Value = "Production type";
                dataGridView1.Columns[4].HeaderCell.Value = "Telephones";
                dataGridView1.Columns[5].HeaderCell.Value = "Manufactirer id";
                if (user == "user")
                {
                    dataGridView1.Columns[0].Visible = false;
                    dataGridView1.Columns[5].Visible = false;
                }
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }


    public void updateMonitor()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible=true;
        label2.Visible = false;
        textBox1.Visible = false;
        button1.Visible = false;
        button2.Visible = false;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;

        //filter
        label3.Visible = true;
        label4.Visible = true;
        label4.Text = "Application:";
        comboBox2.Visible = true;
        comboBox2.Items.Clear();
        comboBox2.Items.Add("all");
        comboBox2.Items.Add("Gaming");
        comboBox2.Items.Add("Office");
        comboBox2.SelectedIndex = 0;
        button3.Visible = true;
        button4.Visible = true;
        label5.Visible = true;
        label5.Text = "Matrix type:";
        label6.Visible = true;
        label6.Text = "Screen type:";
        label7.Visible = false;
        comboBox3.Visible = true;
        comboBox3.Items.Clear();
        comboBox3.Items.Add("all");
        comboBox3.Items.Add("IPS");
        comboBox3.Items.Add("TN");
```

```
            comboBox3.SelectedIndex = 0;
            comboBox4.Visible = true;
            comboBox4.Items.Clear();
            comboBox4.Items.Add("all");
            comboBox4.Items.Add("mate");
            comboBox4.Items.Add("glossy");
            comboBox4.SelectedIndex = 0;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;
            table = "monitor";


            //calculs

            label8.Visible= true;
            label9.Visible= true;
            label10.Visible = false;
            label11.Visible = false;
            comboBox6.Visible = true;
            comboBox6.Items.Clear();
            comboBox6.Items.Add("diagonal");
            comboBox6.SelectedIndex= 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("diagonal");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = true;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Monitor";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
```

```
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                dataGridView1.Columns[5].HeaderCell.Value = "Application";
                if (user == "user") dataGridView1.Columns[0].Visible = false;
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void updatePhone()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible=true;
        label2.Visible = false;
        textBox1.Visible = false;
        button1.Visible = false;
        button2.Visible = false;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;

        //filter
        label3.Visible = true;
        label4.Visible = true;
        label4.Text = "CPU(min):";
        comboBox2.Visible = false;
        comboBox2.Items.Clear();
        button3.Visible = true;
        button4.Visible = true;
        label5.Visible = true;
        label5.Text = "RAM(min):";
        label6.Visible = true;
        label6.Text = "Intern.(min):";
        label7.Visible = true;
        label7.Text = "Application:";
        comboBox3.Visible = false;
        comboBox3.Items.Clear();
        comboBox4.Visible = false;
        comboBox4.Items.Clear();
        comboBox5.Visible = true;
        comboBox5.Items.Clear();
        comboBox5.Items.Add("all");
        comboBox5.Items.Add("Daily");
        comboBox5.Items.Add("Graphic design");
        comboBox5.SelectedIndex = 0;
        checkBox1.Visible = false;
        textBox5.Visible = true;
```

```
            textBox3.Visible = true;
            textBox4.Visible = true;
            table = "phone";

            //calculs

            label8.Visible = true;
            label9.Visible = true;
            label10.Visible = false;
            label11.Visible = false;
            comboBox6.Visible = true;
            comboBox6.Items.Clear();
            comboBox6.Items.Add("CPU");
            comboBox6.Items.Add("RAM");
            comboBox6.Items.Add("Internal memory");
            comboBox6.SelectedIndex = 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("CPU");
            comboBox7.Items.Add("RAM");
            comboBox7.Items.Add("Internal memory");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = true;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Phone";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 5;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
```

```
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                dataGridView1.Columns[3].HeaderCell.Value = "Internal memory";
                dataGridView1.Columns[4].HeaderCell.Value = "Application";
                if (user == "user") dataGridView1.Columns[0].Visible = false;
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void updateComputer()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible=true;
        label2.Visible = false;
        textBox1.Visible = false;
        button1.Visible = false;
        button2.Visible = false;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;

        //filter
        label3.Visible = true;
        label4.Visible = true;
        label4.Text = "CPU(min):";
        comboBox2.Visible = false;
        comboBox2.Items.Clear();
        button3.Visible = true;
        button4.Visible = true;
        label5.Visible = true;
        label5.Text = "RAM(min):";
        label6.Visible = true;
        label6.Text = "HDD(min):";
        label7.Visible = true;
        label7.Text = "Application:";
        comboBox3.Visible = false;
        comboBox3.Items.Clear();
        comboBox4.Visible = false;
        comboBox4.Items.Clear();
        comboBox5.Visible = true;
        comboBox5.Items.Clear();
        comboBox5.Items.Add("all");
        comboBox5.Items.Add("Gaming");
        comboBox5.Items.Add("Office");
        comboBox5.Items.Add("Home");
        comboBox5.Items.Add("Business");
        comboBox5.SelectedIndex = 0;
        checkBox1.Visible = true;
        checkBox1.Text = "CD included: ";
        textBox5.Visible = true;
        textBox3.Visible = true;
        textBox4.Visible = true;
        table = "computer";

        //calculs
```

```
            label8.Visible = true;
            label9.Visible = true;
            label10.Visible = false;
            label11.Visible = false;
            comboBox6.Visible = true;
            comboBox6.Items.Clear();
            comboBox6.Items.Add("CPU");
            comboBox6.Items.Add("RAM");
            comboBox6.Items.Add("HDD");
            comboBox6.SelectedIndex = 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("CPU");
            comboBox7.Items.Add("RAM");
            comboBox7.Items.Add("HDD");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = true;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Computer";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                    dataGridView1.Columns[2].HeaderCell.Value = "RAM";
```

```
                dataGridView1.Columns[3].HeaderCell.Value = "HDD";
                dataGridView1.Columns[4].HeaderCell.Value = "CD";
                dataGridView1.Columns[5].HeaderCell.Value = "Application";
                if (user == "user") dataGridView1.Columns[0].Visible = false;
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void updatePrinter()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible=true;
        label2.Visible = false;
        textBox1.Visible = false;
        button1.Visible = false;
        button2.Visible = false;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;

        //filter
        label3.Visible = true;
        label4.Visible = true;
        label4.Text = "Type:";
        comboBox2.Visible = true;
        comboBox2.Items.Clear();
        comboBox2.Items.Add("all");
        comboBox2.Items.Add("laser");
        comboBox2.Items.Add("color");
        comboBox2.SelectedIndex = 0;
        button3.Visible = true;
        button4.Visible = true;
        label5.Visible = true;
        label5.Text = "Application:";
        label6.Visible = false;
        label7.Visible = false;
        comboBox3.Visible = true;
        comboBox3.Items.Clear();
        comboBox3.Items.Add("all");
        comboBox3.Items.Add("Office");
        comboBox3.Items.Add("Photos");
        comboBox3.SelectedIndex = 0;
        comboBox4.Visible = false;
        comboBox4.Items.Clear();
        comboBox5.Visible = false;
        checkBox1.Visible = false;
        textBox5.Visible = false;
        textBox3.Visible = false;
        textBox4.Visible = false;
        table = "printer";


        //calculs

        label8.Visible = false;
        label9.Visible = false;
        label10.Visible = false;
        label11.Visible = false;
        label12.Visible = false;
```

```
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;



            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Printer";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 3;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Printer type";
                    dataGridView1.Columns[2].HeaderCell.Value = "Application";
                    if (user == "user") dataGridView1.Columns[0].Visible = false;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void updateProductType()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
```

```
            table = "product_type";
            //search
            label1.Visible=true;
            label2.Visible = false;
            textBox1.Visible = false;
            button1.Visible = false;
            button2.Visible = false;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;

            //filter
            label3.Visible=true;
            label4.Visible = false;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox2.Items.Clear();
            comboBox2.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            button3.Visible = false;
            button4.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;


            //calculs

            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;



            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Product_type";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();
```

```
                    dataGridView1.ColumnCount = 2;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Type id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Production type";
                    if (user == "user") dataGridView1.Columns[0].Visible = false;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void updatePayment()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
            table = "payment";
            //search
            label1.Visible=true;
            label2.Visible = false;
            textBox1.Visible = false;
            button1.Visible = false;
            button2.Visible = false;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;

            //filter
            label3.Visible=true;
            label4.Visible = false;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox2.Items.Clear();
            comboBox2.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            button3.Visible = false;
            button4.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;


            //calculs

            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
```

```
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;



            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Payment";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 2;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Payment code";
                    dataGridView1.Columns[1].HeaderCell.Value = "Payment type";
                    if (user == "user") dataGridView1.Columns[0].Visible = false;
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void updateChek()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
            table = "chek";
```

```
            //search
            label1.Visible=true;
            label2.Visible = false;
            textBox1.Visible = false;
            button1.Visible = false;
            button2.Visible = false;
            comboBox1.Visible = false;
            dateTimePicker1.Visible = false;

            //filter
            label3.Visible=true;
            label4.Visible = false;
            label5.Visible = false;
            label6.Visible = false;
            label7.Visible = false;
            comboBox2.Items.Clear();
            comboBox2.Visible = false;
            comboBox3.Visible = false;
            comboBox4.Visible = false;
            comboBox5.Visible = false;
            checkBox1.Visible = false;
            button3.Visible = false;
            button4.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = false;
            textBox4.Visible = false;

            //calculs


            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            label12.Visible = false;
            comboBox6.Visible = false;
            comboBox7.Visible = false;
            comboBox8.Visible = false;
            comboBox9.Visible = false;
            comboBox10.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;




            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Chek";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();
```

```
                    dataGridView1.ColumnCount = 3;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Client id";
                    dataGridView1.Columns[2].HeaderCell.Value = "Payment code";
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void updateConsignment()
        {
            panel1.Visible = true;
            panel2.Visible = true;
            panel3.Visible = true;
            label13.Visible = false;
            pictureBox1.Visible = false;
            //search
            label1.Visible = true;
            label2.Visible = false;
            //label2.Text = "Consign.№:";
            table = "consignment";
            textBox1.Visible = true;
            button1.Visible = true;
            button2.Visible = true;
            comboBox1.Visible = true;
            comboBox1.SelectedIndex = 0;
            dateTimePicker1.Visible = false;

            //filter
            label3.Visible = true;
            label4.Visible = true;
            label4.Text = "Prod. type:";
            comboBox2.Visible = true;
            comboBox2.Items.Clear();
            comboBox2.Items.Add("all");
            comboBox2.Items.Add("PC");
            comboBox2.Items.Add("Laptop");
            comboBox2.Items.Add("Monitor");
            comboBox2.Items.Add("Phone");
            comboBox2.Items.Add("Tablet PC");
            comboBox2.Items.Add("Printer");
            comboBox2.SelectedIndex = 0;
            button3.Visible = true;
            button4.Visible = true;
            label5.Visible = true;
            label5.Text = "Min Prod.cnt:";
            label6.Visible = false;
            label7.Visible = false;
            comboBox3.Visible = false;
            comboBox3.Items.Clear();
```

```
            comboBox4.Visible = false;
            comboBox4.Items.Clear();
            comboBox5.Visible = false;
            comboBox5.Items.Clear();
            checkBox1.Visible = false;
            textBox5.Visible = false;
            textBox3.Visible = true;
            textBox3.Text = "1";
            textBox4.Visible = false;
            table = "consignment";

            //calculs

            label8.Visible = true;
            label9.Visible = true;
            label10.Visible = true;
            label11.Visible = true;
            comboBox6.Visible = true;
            comboBox6.Items.Clear();
            comboBox6.Items.Add("Price");
            comboBox6.SelectedIndex = 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("Price");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = true;
            comboBox8.Items.Clear();
            comboBox8.Items.Add("Price");
            comboBox8.SelectedIndex = 0;
            comboBox9.Visible = true;
            comboBox9.Items.Clear();
            comboBox9.Items.Add("outcome");
            comboBox9.SelectedIndex = 0;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = true;
            button8.Visible = true;
            button9.Visible = true;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from consignment";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 7;
                    int i = 0;

                    while (reader.Read())
                    {
```

```
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                dataGridView1.Columns[2].HeaderCell.Value = "Product amount";
                dataGridView1.Columns[3].HeaderCell.Value = "Manufacturer id";
                dataGridView1.Columns[4].HeaderCell.Value = "Consignment number";
                dataGridView1.Columns[5].HeaderCell.Value = "Consignment date";
                dataGridView1.Columns[6].HeaderCell.Value = "Price";
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void updateChekInfoChek()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible = true;
        label2.Visible = true;
        label2.Text = "Purc. date:";
        table = "chekInfo";
        button1.Visible = true;
        button2.Visible = true;
        comboBox1.Visible = false;
        comboBox1.SelectedIndex = 0;
        textBox1.Visible = false;
        dateTimePicker1.Visible = true;

        //filter
        label3.Visible=true;
        label4.Visible = false;
        label5.Visible = false;
        label6.Visible = false;
        label7.Visible = false;
        comboBox2.Items.Clear();
        comboBox2.Visible = false;
        comboBox3.Visible = false;
        comboBox4.Visible = false;
        comboBox5.Visible = false;
        checkBox1.Visible = false;
        button3.Visible = false;
        button4.Visible = false;
        textBox5.Visible = false;
        textBox3.Visible = false;
        textBox4.Visible = false;


        //calculs
```

```
            label8.Visible = true;
            label9.Visible = true;
            label10.Visible = true;
            label11.Visible = true;
            comboBox6.Visible = true;
            comboBox6.Items.Clear();
            comboBox6.Items.Add("Price");
            comboBox6.SelectedIndex = 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("Price");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = true;
            comboBox8.Items.Clear();
            comboBox8.Items.Add("Price");
            comboBox8.SelectedIndex = 0;
            comboBox9.Visible = true;
            comboBox9.Items.Clear();
            comboBox9.Items.Add("income");
            comboBox9.SelectedIndex = 0;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = true;
            button8.Visible = true;
            button9.Visible = true;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Chek_infoChek";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 4;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Product id";
```

```
                dataGridView1.Columns[2].HeaderCell.Value = "Purchasing date";
                dataGridView1.Columns[3].HeaderCell.Value = "Total price";
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void updateProduct()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = true;
        //search
        label1.Visible = true;
        label2.Visible = true;
        label2.Text = "Product name:";
        textBox1.Visible = true;
        button1.Visible = true;
        button2.Visible = true;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;
        table = "product";

        //filter
        label3.Visible = true;
        label4.Visible = true;
        label4.Text = "Prod. type:";
        comboBox2.Visible = true;
        comboBox2.Items.Clear();
        comboBox2.Items.Add("all");
        comboBox2.Items.Add("PC");
        comboBox2.Items.Add("Laptop");
        comboBox2.Items.Add("Monitor");
        comboBox2.Items.Add("Phone");
        comboBox2.Items.Add("Tablet PC");
        comboBox2.Items.Add("Printer");
        comboBox2.SelectedIndex = 0;
        button3.Visible = true;
        button4.Visible = true;
        label5.Visible = true;
        label5.Text = "Min price:";
        label6.Visible = false;
        label7.Visible = false;
        comboBox3.Visible = false;
        comboBox3.Items.Clear();
        comboBox4.Visible = false;
        comboBox4.Items.Clear();
        comboBox5.Visible = false;
        comboBox5.Items.Clear();
        checkBox1.Visible = false;
        textBox5.Visible = false;
        textBox3.Visible = true;
        textBox3.Text = "100";
        textBox4.Visible = false;

        //calculs

        label8.Visible = true;
        label9.Visible = true;
        label10.Visible = false;
        label11.Visible = false;
        comboBox6.Visible = true;
```

```
            comboBox6.Items.Clear();
            comboBox6.Items.Add("Price");
            comboBox6.SelectedIndex = 0;
            comboBox7.Visible = true;
            comboBox7.Items.Clear();
            comboBox7.Items.Add("Price");
            comboBox7.SelectedIndex = 0;
            comboBox8.Visible = false;
            comboBox8.Items.Clear();
            comboBox9.Visible = false;
            comboBox9.Items.Clear();
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = true;
            comboBox10.Visible = true;
            comboBox10.Items.Clear();
            comboBox10.Items.Add("all");
            comboBox10.Items.Add("PC");
            comboBox10.Items.Add("Laptop");
            comboBox10.Items.Add("Monitor");
            comboBox10.Items.Add("Phone");
            comboBox10.Items.Add("Tablet PC");
            comboBox10.Items.Add("Printer");
            comboBox10.SelectedIndex = 0;
            label12.Visible = true;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Product";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 7;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                        pictureBox1.BackgroundImage =
Image.FromFile(reader.GetValue(6).ToString());
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
```

```
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "Production type id";
                dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer id";
                dataGridView1.Columns[3].HeaderCell.Value = "Product name";
                dataGridView1.Columns[4].HeaderCell.Value = "Product price";
                dataGridView1.Columns[5].HeaderCell.Value = "Waranty";
                dataGridView1.Columns[6].HeaderCell.Value = "Picture";
                if (user == "user")
                {
                    dataGridView1.Columns[0].Visible = false;
                    dataGridView1.Columns[1].Visible = false;
                    dataGridView1.Columns[2].Visible = false;
                    dataGridView1.Columns[6].Visible = false;
                }
            }

        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }


    private void updatePasportData()
    {
        panel1.Visible = true;
        panel2.Visible = true;
        panel3.Visible = true;
        label13.Visible = false;
        pictureBox1.Visible = false;
        //search
        label1.Visible = true;
        label2.Visible = true;
        label2.Text = "     IDNP:";
        textBox1.Visible = true;
        button1.Visible = true;
        button2.Visible = true;
        comboBox1.Visible = false;
        dateTimePicker1.Visible = false;
        table = "pasport";

        //filter
        label3.Visible=true;
        label4.Visible = false;
        label4.Text = "";
        comboBox2.Visible = false;
        comboBox2.Items.Clear();
        button3.Visible = false;
        button4.Visible = false;
        label5.Visible = false;
        label5.Text = "";
        label6.Visible = false;
        label7.Visible = false;
        comboBox3.Visible = false;
        comboBox3.Items.Clear();
        comboBox4.Visible = false;
        comboBox4.Items.Clear();
        comboBox5.Visible = false;
        comboBox5.Items.Clear();
        checkBox1.Visible = false;
        textBox5.Visible = false;
        textBox3.Visible = false;
        textBox3.Text = "";
        textBox4.Visible = false;

        //calculs
```

```
            label8.Visible = false;
            label9.Visible = false;
            label10.Visible = false;
            label11.Visible = false;
            comboBox6.Visible = false;
            comboBox6.Items.Clear();
            comboBox7.Visible = false;
            comboBox7.Items.Clear();
            comboBox8.Visible = false;
            comboBox8.Items.Clear();
            comboBox9.Visible = false;
            comboBox9.Items.Clear();
            button5.Visible = false;
            button6.Visible = false;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;
            comboBox10.Visible = false;
            comboBox10.Items.Clear();
            label12.Visible = false;


            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try
            {
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Pasport_data";

                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Client id";
                    dataGridView1.Columns[1].HeaderCell.Value = "IDNP";
                    dataGridView1.Columns[2].HeaderCell.Value = "Client adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Client surname";
                    dataGridView1.Columns[4].HeaderCell.Value = "Client name";
                    dataGridView1.Columns[5].HeaderCell.Value = "Client father name";
                }
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }


        private void manufacturerToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateManufacturer();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void suplToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateSupplier();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void monitorToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateMonitor();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void phoneToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updatePhone();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void computerToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateComputer();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void printerToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updatePrinter();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void productTypeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateProductType();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void printerToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            updatePayment();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void chekToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateChek();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void consignmentToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateConsignment();
            tablesToolStripMenuItem.HideDropDown();
        }
```

```
        private void chekInfoToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateChekInfoChek();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void aaaToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updateProduct();
            tablesToolStripMenuItem.HideDropDown();
        }

        private void pasportDataToolStripMenuItem_Click(object sender, EventArgs e)
        {
            updatePasportData();
        }




        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if(table == "supplier")
            {
                if(!Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\+(\\d+)?$") && (int)e.KeyChar !=
8)
                {
                    e.Handled = true;
                }
            } else if(table == "consignment")
            {
                if (!Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
                {
                    e.Handled = true;
                }
            }
        }




        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;
            if(table == "supplier" &&
!Regex.IsMatch(textBox1.Text,"^\\+373\\d\\d\\d\\d\\d\\d\\d\\d$"))
            {
                flag = false;
                error.SetError(textBox1, "Number should be in forat +373xxxxxxxx");
            }

            try {
                if (table == "supplier" && flag)
                {
                    string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    SqlConnection connection = new SqlConnection(connectionString);
                    using (connection)
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from supplier where
Telephone = @Telephone", connection);

                        command.Parameters.Add("@Telephone", textBox1.Text);
```

```
                    SqlDataReader reader = command.ExecuteReader();

                    if (reader.HasRows)
                    {
                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                            dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                    }
                }
            }



            if (table == "consignment" && flag)
            {
                string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                SqlConnection connection = new SqlConnection(connectionString);
                if (comboBox1.Text == "Date")
                {
                    //MessageBox.Show(dateTimePicker1.Text + " == " +
dateTimePicker1.Value.ToString());
                    using (connection)
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from consignment where
Consignment_date = @Consignment_date", connection);

                        command.Parameters.Add("@Consignment_date", dateTimePicker1.Value);

                        SqlDataReader reader = command.ExecuteReader();

                        //MessageBox.Show(reader.HasRows.ToString());

                        if (reader.HasRows)
                        {
                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                        }
                    }
                }
                else
```

```
                    {
                        using (connection)
                        {
                            connection.Open();

                            SqlCommand command = new SqlCommand("select * from consignment where
consignment_number = @consignment_number", connection);

                            command.Parameters.Add("@consignment_number", textBox1.Text);

                            SqlDataReader reader = command.ExecuteReader();

                            if (reader.HasRows)
                            {
                                dataGridView1.Rows.Clear();
                                int i = 0;

                                while (reader.Read())
                                {
                                    dataGridView1.RowCount++;
                                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                    dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                    i++;
                                }
                            }
                        }
                    }
                }

                if(table == "chekInfo" && flag)
                {
                    string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    SqlConnection connection = new SqlConnection(connectionString);
                    using (connection)
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("select * from Chek_infoChek where
Purc_date = @Purc_date", connection);

                        command.Parameters.Add("@Purc_date", dateTimePicker1.Value);

                        SqlDataReader reader = command.ExecuteReader();

                        //MessageBox.Show(reader.HasRows.ToString());

                        if (reader.HasRows)
                        {
                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                        }
```

```
                }
            }


            if (table == "product" && flag)
            {
                string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                SqlConnection connection = new SqlConnection(connectionString);
                using (connection)
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand("select * from Product where
Prod_name like @Prod_name", connection);

                    command.Parameters.Add("@Prod_name", "%" + textBox1.Text + "%");

                    SqlDataReader reader = command.ExecuteReader();

                    //MessageBox.Show(reader.HasRows.ToString());

                    if (reader.HasRows)
                    {
                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                            dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                            dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[6].Visible = false;
                    }
                }
            }
            if (table == "pasport" && flag)
            {
                string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                SqlConnection connection = new SqlConnection(connectionString);
                using (connection)
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand("select * from Pasport_data where
IDNP like @IDNP", connection);

                    command.Parameters.Add("@IDNP", "%" + textBox1.Text + "%");

                    SqlDataReader reader = command.ExecuteReader();

                    //MessageBox.Show(reader.HasRows.ToString());

                    if (reader.HasRows)
                    {
                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
```

```
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }



            textBox1.Text = "";
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if(table == "supplier")
            {
                updateSupplier();
            }
            if(table == "consignment")
            {
                updateConsignment();
            }
            if(table == "chekInfo")
            {
                updateChekInfoChek();
            }
            if(table == "product")
            {
                updateProduct();
            }
            if(table == "pasport")
            {
                updatePasportData();
            }
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            if(comboBox1.Text == "Date")
            {
                dateTimePicker1.Visible = true;
                textBox1.Visible = false;
            }
            else
            {
                dateTimePicker1.Visible = false;
                textBox1.Visible = true;
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if(table == "supplier")
            {
                updateSupplier();
            }
```

```
            if (table == "monitor")
            {
                updateMonitor();
            }
            if (table == "phone")
            {
                updatePhone();
            }
            if (table == "computer")
            {
                updateComputer();
            }
            if (table == "printer")
            {
                updatePrinter();
            }
            if (table == "consignment")
            {
                updateConsignment();
            }
            if (table == "chekInfo")
            {
                updateChekInfoChek();
            }
            if (table == "product")
            {
                updateProduct();
            }


        }

        private void button4_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);

            try {
                if(table == "supplier")
                {
                    using (connection)
                    {
                        connection.Open();

                        using (SqlCommand command = new SqlCommand("select * from supplier where
production_type like @production_type", connection))
                        {
                            command.Parameters.Add("@production_type", comboBox2.Text == "all" ?
"%": "%"+comboBox2.Text + "%");

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                                int i = 0;

                                while (reader.Read())
                                {
                                    dataGridView1.RowCount++;
                                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                    i++;
                                }
                                dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
```

```
                    dataGridView1.Columns[1].HeaderCell.Value = "Supplier name";
                    dataGridView1.Columns[2].HeaderCell.Value = "Supplier adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Production type";
                    dataGridView1.Columns[4].HeaderCell.Value = "Telephones";
                    dataGridView1.Columns[5].HeaderCell.Value = "Manufactirer id";
                }
            }
        }

        if(table == "monitor")
        {
            using (connection)
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand("select * from Monitor where
Tatrix_type like @Tatrix_type and Matrix_size like @Matrix_size and Applic like @Applic",
connection))
                {
                    command.Parameters.Add("@Applic", comboBox2.Text == "all" ? "%" : "%"
+ comboBox2.Text + "%");
                    command.Parameters.Add("@Tatrix_type", comboBox3.Text == "all" ? "%"
: "%" + comboBox3.Text + "%");
                    command.Parameters.Add("@Matrix_size", comboBox4.Text == "all" ? "%"
: "%" + comboBox4.Text + "%");

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Rows.Clear();
                    int i = 0;


                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                    dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                    dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                    dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                    dataGridView1.Columns[5].HeaderCell.Value = "Application";
                }
            }
        }

        if (table == "phone")
        {
            using (connection)
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand("select * from Phone where
CPU_freq >= @CPU_freq and RAM >= @RAM and Intern_mem >= @Intern_mem and Applic like @Applic",
connection))
                {
                    command.Parameters.Add("@CPU_freq", textBox5.Text);
                    command.Parameters.Add("@RAM", textBox3.Text);
```

```
                            command.Parameters.Add("@Intern_mem",
Convert.ToInt32(textBox4.Text));
                            command.Parameters.Add("@Applic", comboBox5.Text == "all" ? "%" :
comboBox5.Text);

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                            dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                            dataGridView1.Columns[3].HeaderCell.Value = "Internal memory";
                            dataGridView1.Columns[4].HeaderCell.Value = "Application";
                        }
                    }
                }

                if (table == "computer")
                {
                    using (connection)
                    {
                        connection.Open();

                        using (SqlCommand command = new SqlCommand("select * from Computer where
CPU_freq >= @CPU_freq and RAM >= @RAM and HDD >= @HDD and CD = @CD and Applic like @Applic",
connection))
                        {
                            command.Parameters.Add("@CPU_freq", textBox5.Text);
                            command.Parameters.Add("@RAM", textBox3.Text);
                            command.Parameters.Add("@HDD", Convert.ToInt32(textBox4.Text));
                            command.Parameters.Add("@CD", checkBox1.Checked);
                            command.Parameters.Add("@Applic", comboBox5.Text == "all" ? "%" :
comboBox5.Text);

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
```

```
                        dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                        dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                        dataGridView1.Columns[3].HeaderCell.Value = "HDD";
                        dataGridView1.Columns[4].HeaderCell.Value = "CD";
                        dataGridView1.Columns[5].HeaderCell.Value = "Application";
                    }
                }
            }

            if (table == "printer")
            {
                using (connection)
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand("select * from Printer where
Printer_type like @Printer_type and Applic like @Applic", connection))
                    {
                        command.Parameters.Add("@Printer_type", comboBox2.Text == "all" ? "%"
: "%" + comboBox2.Text + "%");
                        command.Parameters.Add("@Applic", comboBox3.Text == "all" ? "%" : "%"
+ comboBox3.Text + "%");

                        SqlDataReader reader = command.ExecuteReader();

                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
                            //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                        dataGridView1.Columns[1].HeaderCell.Value = "Printer type";
                        dataGridView1.Columns[2].HeaderCell.Value = "Application";
                    }
                }
            }

            if (table == "consignment")
            {
                using (connection)
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand("select * from consignment
where cod_prod in (select cod_prod from Product where Cod_type in (select Cod_type from
Product_type where Product_type like @Product_type)) and prod_number >= @prod_number",
connection))
                    {
                        //MessageBox.Show((comboBox2.Text == "all" ? "%" : comboBox2.Text) +
textBox3.Text);
                        command.Parameters.Add("@Product_type", comboBox2.Text == "all" ? "%"
: comboBox2.Text);
                        command.Parameters.Add("@prod_number",
Convert.ToInt32(textBox3.Text));

                        SqlDataReader reader = command.ExecuteReader();

                        dataGridView1.Rows.Clear();
                        int i = 0;
```

```
                        while (reader.Read())
                        {
                            //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                            dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                            dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                        dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                        dataGridView1.Columns[2].HeaderCell.Value = "Product amount";
                        dataGridView1.Columns[3].HeaderCell.Value = "Manufacturer id";
                        dataGridView1.Columns[4].HeaderCell.Value = "Consignment number";
                        dataGridView1.Columns[5].HeaderCell.Value = "Consignment date";
                        dataGridView1.Columns[6].HeaderCell.Value = "Price";
                    }
                }
            }

            if (table == "product")
            {
                using (connection)
                {
                    connection.Open();

                    using (SqlCommand command = new SqlCommand("select * from Product where
Cod_type in (select Cod_type from Product_type where Product_type like @Product_type) and Price
>= @Price", connection))
                    {
                        //MessageBox.Show((comboBox2.Text == "all" ? "%" : comboBox2.Text) +
textBox3.Text);
                        command.Parameters.Add("@Product_type", comboBox2.Text == "all" ? "%"
: comboBox2.Text);
                        command.Parameters.Add("@Price", Convert.ToInt32(textBox3.Text));

                        SqlDataReader reader = command.ExecuteReader();

                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
                            //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                            dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                            dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[6].Visible = false;
                        dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                        dataGridView1.Columns[1].HeaderCell.Value = "Production type id";
                        dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer id";
                        dataGridView1.Columns[3].HeaderCell.Value = "Product name";
                        dataGridView1.Columns[4].HeaderCell.Value = "Product price";
                        dataGridView1.Columns[5].HeaderCell.Value = "Waranty";
```

```
                    }
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    private void textBox5_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!Regex.IsMatch(textBox5.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
        {
            e.Handled = true;
        }
    }

    private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
        {
            e.Handled = true;
        }
    }

    private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (!Regex.IsMatch(textBox4.Text + e.KeyChar, "^\\d+$") && (int)e.KeyChar != 8)
        {
            e.Handled = true;
        }
    }


    private void button9_Click(object sender, EventArgs e)
    {
        if (table == "monitor")
        {
            updateMonitor();
        }
        if (table == "phone")
        {
            updatePhone();
        }
        if (table == "computer")
        {
            updateComputer();
        }
        if (table == "consignment")
        {
            updateConsignment();
        }
        if (table == "chekInfo")
        {
            updateChekInfoChek();
        }
        if (table == "product")
        {
            updateProduct();
        }
    }
```

```
        private void button5_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "monitor")
                    {
                        using (SqlCommand command = new SqlCommand("select * from Monitor where
Diagonal = (select max(Diagonal) from Monitor)", connection))
                        {
                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                            dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                            dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                            dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                            dataGridView1.Columns[5].HeaderCell.Value = "Application";
                        }
                    }

                    if (table == "phone")
                    {
                        using (SqlCommand command = new SqlCommand())
                        {
                            if (comboBox6.Text == "CPU")
                                command.CommandText = "select * from Phone where CPU_freq =
(select max(CPU_freq) from Phone)";
                            else if(comboBox6.Text == "RAM")
                            {
                                command.CommandText = $"select * from Phone where RAM = (select
max({comboBox6.Text}) from Phone)";
                            } else
                                command.CommandText = $"select * from Phone where Intern_mem =
(select max(Intern_mem) from Phone)";

                            command.Connection = connection;

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
```

```
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                dataGridView1.Columns[3].HeaderCell.Value = "Internal memory";
                dataGridView1.Columns[4].HeaderCell.Value = "Application";
            }
        }

        if (table == "computer")
        {
            using (SqlCommand command = new SqlCommand())
            {
                if (comboBox6.Text == "CPU")
                    command.CommandText = "select * from Computer where CPU_freq =
(select max(CPU_freq) from Computer)";
                else
                    command.CommandText = $"select * from Computer where
{comboBox6.Text} = (select max({comboBox6.Text}) from Computer)";


                command.Connection = connection;

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Rows.Clear();
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                dataGridView1.Columns[3].HeaderCell.Value = "HDD";
                dataGridView1.Columns[4].HeaderCell.Value = "CD";
                dataGridView1.Columns[5].HeaderCell.Value = "Application";
            }
        }

        if (table == "consignment")
        {
            using (SqlCommand command = new SqlCommand("select * from consignment
where Price = (select max(Price) from consignment)", connection))
            {
                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Rows.Clear();
```

```
                        int i = 0;

                        while (reader.Read())
                        {
                            //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                            dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                            dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                        dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                        dataGridView1.Columns[2].HeaderCell.Value = "Product amount";
                        dataGridView1.Columns[3].HeaderCell.Value = "Manufacturer id";
                        dataGridView1.Columns[4].HeaderCell.Value = "Consignment number";
                        dataGridView1.Columns[5].HeaderCell.Value = "Consignment date";
                        dataGridView1.Columns[6].HeaderCell.Value = "Price";
                    }
                }

                if (table == "chekInfo")
                {
                    using (SqlCommand command = new SqlCommand("select * from Chek_infoChek
where Gen_price = (select max(Gen_price) from Chek_infoChek)", connection))
                    {
                        SqlDataReader reader = command.ExecuteReader();

                        dataGridView1.Rows.Clear();
                        int i = 0;

                        while (reader.Read())
                        {
                            //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                            dataGridView1.RowCount++;
                            dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                            dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                            dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                            dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                            dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                            i++;
                        }
                        dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                        dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                        dataGridView1.Columns[2].HeaderCell.Value = "Purchasing date";
                        dataGridView1.Columns[3].HeaderCell.Value = "Total price";
                    }
                }


                if (table == "product")
                {
                    using (SqlCommand command = new SqlCommand("select * from Product where
Price = (select max(Price) from Product where Cod_type in (select Cod_type from Product_type
where Product_type like @Product_type)) and cod_prod in (select cod_prod from Product where
Cod_type in (select Cod_type from Product_type where Product_type like @Product_type))",
connection))
                    {
                        command.Parameters.Add("@Product_type", comboBox10.Text ==
"all"?"%":comboBox10.Text);

                        SqlDataReader reader = command.ExecuteReader();
```

```
                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Production type id";
                            dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer id";
                            dataGridView1.Columns[3].HeaderCell.Value = "Product name";
                            dataGridView1.Columns[4].HeaderCell.Value = "Product price";
                            dataGridView1.Columns[5].HeaderCell.Value = "Waranty";
                        }
                    }




                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void button6_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "monitor")
                    {
                        using (SqlCommand command = new SqlCommand("select * from Monitor where
Diagonal = (select min(Diagonal) from Monitor)", connection))
                        {
                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
```

```
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
                dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                dataGridView1.Columns[4].HeaderCell.Value = "Screen type";
                dataGridView1.Columns[5].HeaderCell.Value = "Application";
            }
        }

        if (table == "phone")
        {
            using (SqlCommand command = new SqlCommand())
            {
                if (comboBox7.Text == "CPU")
                    command.CommandText = "select * from Phone where CPU_freq =
(select min(CPU_freq) from Phone)";
                else if (comboBox7.Text == "RAM")
                {
                    command.CommandText = $"select * from Phone where RAM = (select
min({comboBox7.Text}) from Phone)";
                }
                else
                    command.CommandText = $"select * from Phone where Intern_mem =
(select min(Intern_mem) from Phone)";

                command.Connection = connection;

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Rows.Clear();
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                dataGridView1.Columns[3].HeaderCell.Value = "Internal memory";
                dataGridView1.Columns[4].HeaderCell.Value = "Application";
            }
        }

        if (table == "computer")
        {
            using (SqlCommand command = new SqlCommand())
            {
                if (comboBox7.Text == "CPU")
                    command.CommandText = "select * from Computer where CPU_freq =
(select min(CPU_freq) from Computer)";
                else
                    command.CommandText = $"select * from Computer where
{comboBox7.Text} = (select min({comboBox7.Text}) from Computer)";
```

```
                    command.Connection = connection;

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Rows.Clear();
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                    dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                    dataGridView1.Columns[3].HeaderCell.Value = "HDD";
                    dataGridView1.Columns[4].HeaderCell.Value = "CD";
                    dataGridView1.Columns[5].HeaderCell.Value = "Application";
                }
            }

            if (table == "consignment")
            {
                using (SqlCommand command = new SqlCommand("select * from consignment
where Price = (select min(Price) from consignment)", connection))
                {
                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Rows.Clear();
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[2].HeaderCell.Value = "Product amount";
                    dataGridView1.Columns[3].HeaderCell.Value = "Manufacturer id";
                    dataGridView1.Columns[4].HeaderCell.Value = "Consignment number";
                    dataGridView1.Columns[5].HeaderCell.Value = "Consignment date";
                    dataGridView1.Columns[6].HeaderCell.Value = "Price";
                }
            }

            if (table == "chekInfo")
            {
```

```
                        using (SqlCommand command = new SqlCommand("select * from Chek_infoChek
where Gen_price = (select min(Gen_price) from Chek_infoChek)", connection))
                        {
                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[2].HeaderCell.Value = "Purchasing date";
                            dataGridView1.Columns[3].HeaderCell.Value = "Total price";
                        }
                    }

                    if (table == "product")
                    {
                        using (SqlCommand command = new SqlCommand("select * from Product where
Price = (select min(Price) from Product where Cod_type in (select Cod_type from Product_type
where Product_type like @Product_type)) and cod_prod in (select cod_prod from Product where
Cod_type in (select Cod_type from Product_type where Product_type like @Product_type))",
connection))
                        {
                            command.Parameters.Add("@Product_type", comboBox10.Text == "all" ?
"%" : comboBox10.Text);

                            SqlDataReader reader = command.ExecuteReader();

                            dataGridView1.Rows.Clear();
                            int i = 0;

                            while (reader.Read())
                            {
                                //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                                dataGridView1.RowCount++;
                                dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                                dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                                dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                                dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                                dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                                dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                                dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                                i++;
                            }
                            dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                            dataGridView1.Columns[1].HeaderCell.Value = "Production type id";
                            dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer id";
                            dataGridView1.Columns[3].HeaderCell.Value = "Product name";
                            dataGridView1.Columns[4].HeaderCell.Value = "Product price";
                            dataGridView1.Columns[5].HeaderCell.Value = "Waranty";
                        }
                    }


                }
            }
```

```
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void button8_Click(object sender, EventArgs e)
        {

            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "consignment")
                    {
                        using (SqlCommand command = new SqlCommand("select AVG(Price) from
consignment", connection))
                        {
                            MessageBox.Show($"Consignments' Average Price =
{command.ExecuteScalar()}MDL", "Average price", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
                        }
                    }

                    if (table == "chekInfo")
                    {
                        using (SqlCommand command = new SqlCommand("select AVG(Gen_price) from
Chek_infoChek", connection))
                        {
                            MessageBox.Show($"Average buying price =
{command.ExecuteScalar()}MDL", "Average buying price", MessageBoxButtons.OK,
MessageBoxIcon.Asterisk);
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void button7_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            SqlConnection connection = new SqlConnection(connectionString);
            try
            {
                using (connection)
                {
                    connection.Open();
                    if (table == "consignment")
                    {
                        using (SqlCommand command = new SqlCommand("select Sum(Price) from
consignment", connection))
                        {
                            MessageBox.Show($"Total consignments' outcome =
{command.ExecuteScalar()}MDL", "Total outcome", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                        }
                    }
```

```
                    if (table == "chekInfo")
                    {
                        using (SqlCommand command = new SqlCommand("select Sum(Gen_price) from
Chek_infoChek", connection))
                        {
                            MessageBox.Show($"Total income = {command.ExecuteScalar()}MDL",
"Total income", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void consignmentToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            ConsignmentReportForm consignmentRF = new ConsignmentReportForm(ds);
            consignmentRF.Show();
        }

        private void checksToolStripMenuItem_Click(object sender, EventArgs e)
        {
            CheckReportForm crf = new CheckReportForm();
            crf.Show();
        }

        private void informationToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            ProductReportForm prf = new ProductReportForm();
            prf.Show();
        }

        private void diagramToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ProductDiagramReportForm pdrf = new ProductDiagramReportForm();
            pdrf.Show();
        }

        private void addToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            PasportDataEditForm pasportData = new PasportDataEditForm(this, "add");
            pasportData.ShowDialog();
            updatePasportData();
        }

        private void editToolStripMenuItem12_Click(object sender, EventArgs e)
        {
            if (user == "manager")
            {
                PasportDataEditForm pasportData = new PasportDataEditForm(this, "edit-only");
                pasportData.ShowDialog();
                updatePasportData();
            }
            else
            {
                PasportDataEditForm pasportData = new PasportDataEditForm(this, "edit");
                pasportData.ShowDialog();
                updatePasportData();
            }
        }

        private void addNewAdminToolStripMenuItem_Click(object sender, EventArgs e)
        {
            RolesEditForm roles = new RolesEditForm(this);
            roles.ShowDialog();
```

```
        }

        private void backupDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DateTime now = DateTime.Now;
            String backupName = now.Year.ToString() + "." + now.Month.ToString() + "." +
now.Day.ToString() + "_Backup.bak";
            saveFileDialog1.FileName= backupName;
            saveFileDialog1.DefaultExt = ".bak";

            if(saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(saveFileDialog1.FileName, ".+\\.bak")){

                    //MessageBox.Show(saveFileDialog1.FileName);


                    try
                    {
                        using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                        {
                            connection.Open();

                            String query = "backup database Computer_magazine to disk = @Path";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.Add("@Path", saveFileDialog1.FileName);

                            command.ExecuteNonQuery();
                        }

                        MessageBox.Show("Successfully saved!", "Success!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect backup extension!", "Invalid extension",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void restoreDataBaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                if(Regex.IsMatch(openFileDialog1.FileName, ".+\\.bak"))
                {
                    try
                    {
                        using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                        {
                            connection.Open();

                            String query = "use master";
                            SqlCommand command = new SqlCommand(query, connection);
                            command.ExecuteNonQuery();

                            command.CommandText = "restore database Computer_magazine from disk =
@Path";
```

```
                            command.Parameters.Add("@Path", openFileDialog1.FileName);
                            command.ExecuteNonQuery();

                            command.CommandText = "use Computer_magazine";
                            command.ExecuteNonQuery();

                            MessageBox.Show("Successfully restored!", "Success!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }
                    } catch(Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Incorrect backup extension!", "Invalid extension",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            if (table == "product")
            {
                //MessageBox.Show(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
                pictureBox1.BackgroundImage =
Image.FromFile(dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString());
            }
        }
    }
}
```

RolesEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Computer_magazine
{
    public partial class RolesEditForm : Form
    {
        private Form caller;
        string mode = "edit";
        int your_row = -1;
        int admins_count = 0;
        public RolesEditForm(Form caller)
        {
            InitializeComponent();
            this.caller = caller;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
```

```
            label2.ForeColor = Color.Black;
            label3.ForeColor = Color.Black;
            label4.ForeColor = Color.Black;
            checkBox1.ForeColor = Color.Black;
            dataGridView1.BackgroundColor = Color.WhiteSmoke;
        }
        else
        {
            this.BackColor = Color.FromArgb(35, 35, 35);
            label1.ForeColor = Color.White;
            label2.ForeColor = Color.White;
            label3.ForeColor = Color.White;
            label4.ForeColor = Color.White;
            checkBox1.ForeColor = Color.White;
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
        }

        comboBox1.Items.Add("user");
        comboBox1.Items.Add("seller");
        comboBox1.Items.Add("manager");
        comboBox1.Items.Add("admin");

        updateDG();
    }




    private void updateDG() {
        try
        {
            using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
            {
                connection.Open();

                SqlCommand command = new SqlCommand("select * from Users", connection);

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.ColumnCount = 3;
                int i = 0;
                your_row = -1;
                admins_count = 0;
                dataGridView1.Rows.Clear();

                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        //MessageBox.Show(reader.GetValue(0) + " == " +
((Form1)caller).getUserLogin() + " - " +
(reader.GetValue(0).Equals(((Form1)caller).getUserLogin())));
                        if (reader.GetValue(0).Equals(((Form1)caller).getUserLogin()))
                            your_row = i;

                        if (reader.GetValue(2).Equals("admin"))
                            admins_count++;

                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].HeaderCell.Value = (i+1).ToString();
                        i++;
                    }
```

```
                    dataGridView1.Columns[0].HeaderCell.Value = "User login";
                    dataGridView1.Columns[1].HeaderCell.Value = "User password";
                    dataGridView1.Columns[2].HeaderCell.Value = "User role";
                }
            }
        }
        catch(Exception ex)
        {
            MessageBox.Show("An error has been occured: "+ex.Message, "Error!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        if(mode == "edit")
        {
            //MessageBox.Show($"e.RowIndex({e.RowIndex}) == your_row({your_row}) - " +
(e.RowIndex == your_row).ToString());
            if(e.RowIndex == your_row)
                comboBox1.Enabled= false;
            else
                comboBox1.Enabled = true;


            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
        }

        if(mode == "delete")
        {
            if (!dataGridView1.Rows[e.RowIndex].Cells[2].Value.Equals("admin"))
            {
                try
                {
                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand("delete from Users where
user_login = @user_login", connection);
                        command.Parameters.Add("@user_login",
dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

                        if (MessageBox.Show("Are you sure?", "Confirm!",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                        {
                            command.ExecuteNonQuery();
                        }

                        updateDG();
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("An error has been occured: " + ex.Message, "Error!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }

            }
            else if (admins_count > 1)
            {
                try
                {
                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                    {
```

```
                            connection.Open();

                            SqlCommand command = new SqlCommand("delete from Users where
user_login = @user_login", connection);
                            command.Parameters.Add("@user_login",
dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

                            if (MessageBox.Show("Are you sure?", "Confirm!",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                            {
                                command.ExecuteNonQuery();

                                if (e.RowIndex == your_row)
                                {
                                    caller.Close();
                                }
                            }

                            updateDG();
                        }
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("An error has been occured: " + ex.Message, "Error!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
                else
                {
                        MessageBox.Show("There are only one admin, you can't delete him!",
"Denied!", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }



            }

        }

        private void button1_Click(object sender, EventArgs e)
        {
            if(textBox3.Text!="" && textBox1.Text != "")
            {
                try
                {
                    using (SqlConnection connection = new SqlConnection("Data
Source=.\\SQLEXPRESS;Initial Catalog=Computer_magazine;Integrated Security=True"))
                    {
                        connection.Open();

                        SqlCommand command = new SqlCommand($"update Users set user_login =
@user_login, user_password = @user_password, user_role = @user_role where user_login =
'{textBox3.Text}'", connection);

                        command.Parameters.Add("@user_login", textBox3.Text);
                        command.Parameters.Add("@user_password", textBox1.Text);
                        command.Parameters.Add("@user_role", comboBox1.Text);

                        int result = command.ExecuteNonQuery();

                        if(result > 0)
                        {
                            MessageBox.Show("Success! " + result.ToString() + " rows affected!",
"Success!", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            updateDG();
                        }


                    }
```

```
                }
                catch(Exception ex)
                {
                    MessageBox.Show("An error has been occured: " + ex.Message, "Error!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if(mode == "edit")
            {
                mode = "delete";
                button1.Visible= false;


                textBox3.Text = "";
                textBox1.Text = "";
                comboBox1.Text = "";

                textBox3.Enabled = false;
                textBox1.Enabled = false;
                comboBox1.Enabled = false;
            }
            else
            {
                mode = "edit";
                button1.Visible= true;

                textBox3.Enabled = true;
                textBox1.Enabled = true;
                comboBox1.Enabled = true;
            }
        }
    }
}
```

SupEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Computer_magazine
{
    public partial class SupEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int max_id;
        private int old_id;
```

```
        public SupEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                label5.ForeColor = Color.White;
                label6.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 288;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select Man_cod from supplier", connection);

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    comboBox1.Items.Add(reader.GetValue(0).ToString());
                }

                comboBox1.Sorted= true;
                comboBox1.SelectedIndex= 0;

            }

        }
```

```
        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
            {
                e.Handled = true;
            }
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                mode = "delete";
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                textBox3.Enabled = false;
                textBox4.Enabled = false;
                textBox6.Enabled = false;
                comboBox1.Enabled = false;
                button1.Visible= false;
            }
            else
            {
                mode = "edit";
                textBox1.Enabled = true;
                textBox2.Enabled = true;
                textBox3.Enabled = true;
                textBox4.Enabled = true;
                textBox6.Enabled = true;
                comboBox1.Enabled = true;
                button1.Visible= true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit")
            {
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
                textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
                textBox6.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
                comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
                old_id = Convert.ToInt32(textBox1.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from supplier where Sup_code =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
```

```
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }

            if(mode != "delete" && !Regex.IsMatch(textBox6.Text,
"^\\+373\\d\\d\\d\\d\\d\\d\\d$")){
                error.SetError(textBox6, "Enter phone in format +373xxxxxxxx");
                flag = false;
            }

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into supplier
values({Convert.ToInt32(textBox1.Text)}, '{textBox2.Text}', '{textBox3.Text}', '{textBox4.Text}',
'{textBox6.Text}', {Convert.ToInt32(comboBox1.Text)})";
                            query = "insert into supplier values(@Sup_code, @Sup_name,
@Sup_Adres, @production_type, @Telephone, @Man_cod)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Sup_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Sup_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Sup_Adres", textBox3.Text);
                            command.Parameters.AddWithValue("@production_type", textBox4.Text);
                            command.Parameters.AddWithValue("@Telephone", textBox6.Text);
                            command.Parameters.AddWithValue("@Man_cod", comboBox1.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit")
                        {
                            query = $"update supplier set Sup_code = @Sup_code, Sup_name =
@Sup_name, Sup_Adres = @Sup_Adres, production_type = @production_type, Telephone = @Telephone,
Man_cod = @Man_cod where Sup_code = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Sup_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Sup_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Sup_Adres", textBox3.Text);
```

```
                        command.Parameters.AddWithValue("@production_type", textBox4.Text);
                        command.Parameters.AddWithValue("@Telephone", textBox6.Text);
                        command.Parameters.AddWithValue("@Man_cod", comboBox1.Text);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                        //Close();
                    }
                    else if (mode == "delete")
                    {
                        //Close();
                    }
                }
                if (mode != "add")
                    updateDG();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }


    private void updateDG()
    {
        try
        {
            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            using (connection)
            {
                connection.Open();

                string query = "select max(Sup_code) from supplier";
                SqlCommand command = new SqlCommand(query, connection);

                max_id = (int)command.ExecuteScalar();


                query = "select * from supplier";
                command.CommandText = query;

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Visible = true;
                dataGridView1.ColumnCount = 6;
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
```

```
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Supplier id";
                dataGridView1.Columns[1].HeaderCell.Value = "Supplier name";
                dataGridView1.Columns[2].HeaderCell.Value = "Supplier adres";
                dataGridView1.Columns[3].HeaderCell.Value = "Production type";
                dataGridView1.Columns[4].HeaderCell.Value = "Telephones";
                dataGridView1.Columns[5].HeaderCell.Value = "Manufactirer id";
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
}
}
```

ProductTypeEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class ProductTypeEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
        private int max_id;

        public ProductTypeEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
```

```
            }

            if (mode == "add")
            {
                this.Width = 300;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            try
            {

                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

                using (connection)
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand("select max(Cod_type) from Product_type",
connection);

                    max_id = Convert.ToInt32(command.ExecuteScalar());
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }


        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
            {
                e.Handled = true;
            }
        }


        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                mode = "delete";
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                button1.Visible= false;
            }
            else
            {
                mode = "edit";
                textBox1.Enabled = true;
                textBox2.Enabled = true;
                button1.Visible = true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit")
            {
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                old_id = Convert.ToInt32(textBox1.Text);
            }
            else if (mode == "delete")
            {
```

```
            if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query = $"delete from Product_type where Cod_type =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                        SqlCommand command = new SqlCommand(query, connection);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
                    }

                    updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        error.Clear();
        bool flag = true;

        if (mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
        {
            error.SetError(textBox1, $"Id shoud be more than {max_id}");
            flag = false;
        }

        if (flag)
        {
            try
            {
                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();
                    string query;

                    if (mode == "add")
                    {
                        query = $"insert into Product_type values(@Cod_type, @Product_type)";

                        SqlCommand command = new SqlCommand(query, connection);

                        command.Parameters.AddWithValue("@Cod_type", textBox1.Text);
                        command.Parameters.AddWithValue("@Product_type", textBox2.Text);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
                        Close();
                    }
                    else if (mode == "edit")
                    {
                        //query = $"update Phone set cod_prod = {Convert.ToInt32(comboBox1.Text)},
CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM = {Convert.ToInt32(textBox2.Text)}, Intern_mem =
{Convert.ToInt32(textBox3.Text)}, Applic = '{textBox4.Text}' where cod_prod = {old_id}";
                        query = $"update Product_type set Cod_type = @Cod_type, Product_type =
@Product_type where Cod_type = {old_id}";

                        SqlCommand command = new SqlCommand(query, connection);

                        command.Parameters.AddWithValue("@Cod_type", textBox1.Text);
                        command.Parameters.AddWithValue("@Product_type", textBox2.Text);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
```

```
                        //Close();
                    }
                    else if (mode == "delete")
                    {
                        //Close();
                    }
                }
                if (mode != "add")
                    updateDG();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }


    private void updateDG()
    {
        try
        {
            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            using (connection)
            {
                connection.Open();

                string query = "select max(Cod_type) from Product_type";
                SqlCommand command = new SqlCommand(query, connection);

                max_id = Convert.ToInt32(command.ExecuteScalar());



                query = "select * from Product_type";
                command = new SqlCommand(query, connection);

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Visible = true;
                dataGridView1.ColumnCount = 2;
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Type id";
                dataGridView1.Columns[1].HeaderCell.Value = "Production type";
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

ProductReportForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class ProductReportForm : Form
    {
        public ProductReportForm()
        {
            InitializeComponent();
        }

        private void ProductReportForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'computer_magazineDataSet.Product_view' table. You can move, or remove it, as needed.
            this.product_viewTableAdapter1.Fill(this.computer_magazineDataSet.Product_view);
            //// TODO: данная строка кода позволяет загрузить данные в таблицу
"computer_magazineDataSet.Product_view". При необходимости она может быть перемещена или удалена.
            //this.product_viewTableAdapter.Fill(this.computer_magazineDataSet.Product_view);

            this.reportViewer1.RefreshReport();
        }
    }
}
```

ProductEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class ProductEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
        private int max_id;

        public ProductEditForm(Form caller, string mode)
        {
            InitializeComponent();
```

```
            //openFileDialog1.FileName = "";

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                label7.ForeColor = Color.Black;
                label8.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.MidnightBlue;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                label5.ForeColor = Color.White;
                label6.ForeColor = Color.White;
                label7.ForeColor = Color.White;
                label8.ForeColor = Color.DeepSkyBlue;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }


            if (mode == "add")
            {
                this.Width = 280;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";


            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select Cod_type from Product_type",
connection);

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    comboBox1.Items.Add(reader.GetValue(0).ToString());
                }

                comboBox1.Sorted = true;
                comboBox1.SelectedIndex = 0;
```

```
            command = new SqlCommand("select Man_code from manufacturer", connection);

            reader.Close();

            reader = command.ExecuteReader();

            while (reader.Read())
            {
                comboBox2.Items.Add(reader.GetValue(0).ToString());
            }

            comboBox2.Sorted = true;
            comboBox2.SelectedIndex = 0;
        }
    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox4.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            comboBox1.Enabled = false;
            comboBox2.Enabled = false;
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            textBox4.Enabled = false;
            button1.Visible= false;
            label8.Enabled=false;
        }
        else
        {
            mode = "edit";
            comboBox1.Enabled = true;
            comboBox2.Enabled = true;
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox3.Enabled = true;
            textBox4.Enabled = true;
            button1.Visible = true;
            label8.Enabled = true;
        }
    }
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
```

```
        {
            if (mode == "edit")
            {
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                comboBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
                textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
                textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
                textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();

                openFileDialog1.FileName =
dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();

                String[] filePath = openFileDialog1.FileName.Replace("/", "\\").Split('\\');
                label8.Text = filePath[filePath.Length - 1];

                old_id = Convert.ToInt32(textBox1.Text);

            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Product where cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }

            //MessageBox.Show(openFileDialog1.FileName);

            if(openFileDialog1.FileName == "")
            {
                error.SetError(label8, "Choose a picture");
                flag = false;
            }
```

```
            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into Monitor
values({Convert.ToInt32(comboBox1.Text)}, '{textBox2.Text}', {Convert.ToInt32(textBox3.Text)},
'{textBox4.Text}', '{textBox5.Text}', '{textBox6.Text}')";
                            query = $"insert into Product values(@cod_prod, @Cod_type, @Man_cod,
@Prod_name, @Price, @Waranty, @Picture)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", textBox1.Text);
                            command.Parameters.AddWithValue("@Cod_type", comboBox1.Text);
                            command.Parameters.AddWithValue("@Man_cod", comboBox2.Text);
                            command.Parameters.AddWithValue("@Prod_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Price", textBox3.Text);
                            command.Parameters.AddWithValue("@Waranty", textBox4.Text);
                            command.Parameters.AddWithValue("@Picture",
openFileDialog1.FileName);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Monitor set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, Tatrix_type = '{textBox2.Text}', Diagonal =
{Convert.ToInt32(textBox3.Text)}, Monitor_type = '{textBox4.Text}', Matrix_size =
'{textBox5.Text}', Applic = '{textBox6.Text}' where cod_prod = {old_id}";
                            query = $"update Product set cod_prod = @cod_prod, Cod_type =
@Cod_type,  Man_cod = @Man_cod, Prod_name = @Prod_name, Price = @Price, Waranty = @Waranty,
Picture = @Picture where cod_prod = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", textBox1.Text);
                            command.Parameters.AddWithValue("@Cod_type", comboBox1.Text);
                            command.Parameters.AddWithValue("@Man_cod", comboBox2.Text);
                            command.Parameters.AddWithValue("@Prod_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Price", textBox3.Text);
                            command.Parameters.AddWithValue("@Waranty", textBox4.Text);
                            MessageBox.Show(openFileDialog1.FileName);
                            command.Parameters.AddWithValue("@Picture",
openFileDialog1.FileName);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
```

```
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();


                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();


                    string query = "select max(cod_prod) from Product";
                    SqlCommand command = new SqlCommand(query, connection);

                    max_id = (int)command.ExecuteScalar();


                    query = "select * from Product";
                    command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 7;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].Cells[6].Value = reader.GetValue(6);
                        dataGridView1.Rows[i].HeaderCell.Value = i.ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Production type id";
                    dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer id";
                    dataGridView1.Columns[3].HeaderCell.Value = "Product name";
                    dataGridView1.Columns[4].HeaderCell.Value = "Product price";
                    dataGridView1.Columns[5].HeaderCell.Value = "Waranty";
                    dataGridView1.Columns[6].HeaderCell.Value = "Picture";
                }
            }
            catch (Exception ex)
            {
```

```
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void label8_Click(object sender, EventArgs e)
        {

            if(openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                String[] filePath = openFileDialog1.FileName.Replace("/", "\\").Split('\\');
                label8.Text = filePath[filePath.Length-1];
            }
        }
    }
}
```

ProductDiagramReportForm.cs

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class ProductDiagramReportForm : Form
    {
        public ProductDiagramReportForm()
        {
            InitializeComponent();
        }

        private void ProductDiagramReportForm_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"computer_magazineDataSet.Product_view". При необходимости она может быть перемещена или удалена.
            this.product_viewTableAdapter.Fill(this.computer_magazineDataSet.Product_view);

            this.reportViewer1.RefreshReport();
        }
    }
}
```

PrinterEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class PrinterEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;

        public PrinterEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 250;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select cod_prod from Product where Cod_type
= 2050", connection); //2050 - Printers id;

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    comboBox1.Items.Add(reader.GetValue(0).ToString());
```

```
            }

            comboBox1.Sorted = true;
            comboBox1.SelectedIndex = 0;

        }
    }


    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            comboBox1.Enabled = false;
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            button1.Visible= false;
        }
        else
        {
            mode = "edit";
            comboBox1.Enabled = true;
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            button1.Visible= true;
        }
    }
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if (mode == "edit")
        {
            comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            old_id = Convert.ToInt32(comboBox1.Text);
        }
        else if (mode == "delete")
        {
            if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query = $"delete from Printer where cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                        SqlCommand command = new SqlCommand(query, connection);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                    }

                    updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }
    }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (mode!= "delete" && (!textBox1.Text.Equals("color")) &&
(!textBox1.Text.Equals("laser")))
            {
                flag = !flag;
                error.SetError(textBox1, "Printer type should be only color or laser");
            }

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            query = $"insert into Printer values(@cod_prod, @Printer_type,
@Applic)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@Printer_type", textBox1.Text);
                            command.Parameters.AddWithValue("@Applic", textBox2.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Phone set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM =
{Convert.ToInt32(textBox2.Text)}, Intern_mem = {Convert.ToInt32(textBox3.Text)}, Applic =
'{textBox4.Text}' where cod_prod = {old_id}";
                            query = $"update Printer set cod_prod = @cod_prod, Printer_type =
@Printer_type, Applic = @Applic where cod_prod = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@Printer_type", textBox1.Text);
                            command.Parameters.AddWithValue("@Applic", textBox2.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
```

```
            }
        }
    }


    private void updateDG()
    {
        try
        {
            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            using (connection)
            {
                connection.Open();

                string query = "select * from Printer";
                SqlCommand command = new SqlCommand(query, connection);

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Visible = true;
                dataGridView1.ColumnCount = 3;
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "Printer type";
                dataGridView1.Columns[2].HeaderCell.Value = "Application";
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
}
```

PhoneEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class PhoneEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;

        public PhoneEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                label5.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 290;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";


            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select cod_prod from Product where Cod_type
= 2030 or Cod_type = 2040", connection); //2030 - Phones id; 2040 - Tablet pc-es id
```

```
            SqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                comboBox1.Items.Add(reader.GetValue(0).ToString());
            }

            comboBox1.Sorted = true;
            comboBox1.SelectedIndex = 0;

        }

    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox2.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            comboBox1.Enabled = false;
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            textBox4.Enabled = false;
            button1.Visible= false;
        }
        else
        {
            mode = "edit";
            comboBox1.Enabled = true;
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox3.Enabled = true;
            textBox4.Enabled = true;
            button1.Visible = true;
        }
    }
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if (mode == "edit")
        {
            comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
```

```
                textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
                textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
                old_id = Convert.ToInt32(comboBox1.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Phone where cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into Phone
values({Convert.ToInt32(comboBox1.Text)}, {Convert.ToInt32(textBox1.Text)},
{Convert.ToInt32(textBox2.Text)}, {Convert.ToInt32(textBox3.Text)}, '{textBox4.Text}')";
                            query = $"insert into Phone values(@cod_prod, @CPU_freq, @RAM,
@Intern_mem, @Applic)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@CPU_freq", textBox1.Text);
                            command.Parameters.AddWithValue("@RAM", textBox2.Text);
                            command.Parameters.AddWithValue("@Intern_mem", textBox3.Text);
                            command.Parameters.AddWithValue("@Applic", textBox4.Text);
```

```
                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Phone set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM =
{Convert.ToInt32(textBox2.Text)}, Intern_mem = {Convert.ToInt32(textBox3.Text)}, Applic =
'{textBox4.Text}' where cod_prod = {old_id}";
                            query = $"update Phone set cod_prod = @cod_prod, CPU_freq =
@CPU_freq, RAM = @RAM, Intern_mem = @Intern_mem, Applic = @Applic where cod_prod = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@CPU_freq", textBox1.Text);
                            command.Parameters.AddWithValue("@RAM", textBox2.Text);
                            command.Parameters.AddWithValue("@Intern_mem", textBox3.Text);
                            command.Parameters.AddWithValue("@Applic", textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();


                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Phone";
                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 5;
                    int i = 0;
```

```
                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                    dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                    dataGridView1.Columns[3].HeaderCell.Value = "Internal memory";
                    dataGridView1.Columns[4].HeaderCell.Value = "Application";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

PaymentEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class PaymentEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
        private int max_id;

        public PaymentEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
```

```
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 270;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            try
            {

                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

                using (connection)
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand("select max(Pay_code) from Payment",
connection);

                    max_id = Convert.ToInt32(command.ExecuteScalar());
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
            {
                e.Handled = true;
            }
        }


        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                mode = "delete";
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                button1.Visible= false;
```

```
            }
            else
            {
                mode = "edit";
                textBox1.Enabled = true;
                textBox2.Enabled = true;
                button1.Visible = true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit")
            {
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                old_id = Convert.ToInt32(textBox1.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Payment where Pay_code =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }

            /*if (mode != "delete" && !textBox2.Text.Equals("cash") &&
!textBox2.Text.Equals("credit") && !textBox2.Text.Equals("credit card"))
            {
                error.SetError(textBox2, $"Payment mode should be cash, credit or credit catd
only!");
                flag = false;
            }*/

            if (flag)
```

```
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            query = $"insert into Payment values(@Pay_code, @Pay_type)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Pay_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Pay_type", textBox2.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Phone set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM =
{Convert.ToInt32(textBox2.Text)}, Intern_mem = {Convert.ToInt32(textBox3.Text)}, Applic =
'{textBox4.Text}' where cod_prod = {old_id}";
                            query = $"update Payment set Pay_code = @Pay_code, Pay_type =
@Pay_type where Pay_code = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Pay_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Pay_type", textBox2.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();
```

```
                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();

                    string query = "select max(Pay_code) from Payment";
                    SqlCommand command = new SqlCommand(query, connection);

                    max_id = (int)command.ExecuteScalar();


                    query = "select * from Payment";
                    command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 2;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Payment code";
                    dataGridView1.Columns[1].HeaderCell.Value = "Payment type";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

PasportDataEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class PasportDataEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
```

```
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
        private int max_id;

        public PasportDataEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                label5.ForeColor = Color.White;
                label6.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            try
            {
                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();


                    string query = "select max(Client_id) from Pasport_data";
                    SqlCommand command = new SqlCommand(query, connection);

                    max_id = (int)command.ExecuteScalar();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }


            if (mode == "add")
            {
                this.Width = 290;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
```

```
            checkBox1.Visible = true;
            updateDG();

            //fill the fields with the first line
        }

        if (mode == "edit-only")
        {
            checkBox1.Visible = false;
            updateDG();
        }
    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox6_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox6.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            textBox6.Enabled = false;
            textBox5.Enabled = false;
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            textBox4.Enabled = false;
            button1.Visible = false;
        }
        else
        {
            mode = "edit";
            textBox6.Enabled = true;
            textBox5.Enabled = true;
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox3.Enabled = true;
            textBox4.Enabled = true;
            button1.Visible = true;
        }
    }
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if (mode == "edit" || mode == "edit-only")
        {
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox6.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox5.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
            textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
            old_id = Convert.ToInt32(textBox1.Text);

        }
        else if (mode == "delete")
        {
```

```
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Pasport_data where Client_id =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            //MessageBox.Show(max_id.ToString());

            if (mode == "add" && textBox1.Text != "" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }


            if (mode == "add" && textBox6.Text.Length < 13)
            {
                error.SetError(textBox6, "IDNP shoud contain 13 numbers");
                flag = false;
            }

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into Monitor
values({Convert.ToInt32(comboBox1.Text)}, '{textBox2.Text}', {Convert.ToInt32(textBox3.Text)},
'{textBox4.Text}', '{textBox5.Text}', '{textBox6.Text}')";
                            query = $"insert into Pasport_data values(@Client_id, @IDNP,
@C_Adres, @C_Surname, @C_Name, @C_FatherName)";
```

```
                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Client_id", textBox1.Text);
                            command.Parameters.AddWithValue("@IDNP", textBox6.Text);
                            command.Parameters.AddWithValue("@C_Adres", textBox5.Text);
                            command.Parameters.AddWithValue("@C_Surname", textBox2.Text);
                            command.Parameters.AddWithValue("@C_Name", textBox3.Text);
                            command.Parameters.AddWithValue("@C_FatherName", textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit" || mode == "edit-only")
                        {
                            //query = $"update Monitor set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, Tatrix_type = '{textBox2.Text}', Diagonal =
{Convert.ToInt32(textBox3.Text)}, Monitor_type = '{textBox4.Text}', Matrix_size =
'{textBox5.Text}', Applic = '{textBox6.Text}' where cod_prod = {old_id}";
                            query = $"update Pasport_data set Client_id = @Client_id, IDNP =
@IDNP,  C_Adres = @C_Adres, C_Name = @C_Name, C_Surname = @C_Surname, C_FatherName =
@C_FatherName where Client_id = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Client_id", textBox1.Text);
                            command.Parameters.AddWithValue("@IDNP", textBox6.Text);
                            command.Parameters.AddWithValue("@C_Adres", textBox5.Text);
                            command.Parameters.AddWithValue("@C_Surname", textBox2.Text);
                            command.Parameters.AddWithValue("@C_Name", textBox3.Text);
                            command.Parameters.AddWithValue("@C_FatherName", textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();


                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
```

```
                    connection.Open();


                    string query = "select max(Client_id) from Pasport_data";
                    SqlCommand command = new SqlCommand(query, connection);

                    max_id = (int)command.ExecuteScalar();


                    query = "select * from Pasport_data";
                    command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = i.ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Client id";
                    dataGridView1.Columns[1].HeaderCell.Value = "IDNP";
                    dataGridView1.Columns[2].HeaderCell.Value = "Client adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Client surname";
                    dataGridView1.Columns[4].HeaderCell.Value = "Client name";
                    dataGridView1.Columns[5].HeaderCell.Value = "Client father name";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

MonitorEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
```

```
    public partial class MonitorEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;

        public MonitorEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                label5.ForeColor = Color.White;
                label6.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 260;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select cod_prod from Product where Cod_type
= 2020", connection); //2020- Monitors id

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
```

```
                comboBox1.Items.Add(reader.GetValue(0).ToString());
            }

            comboBox1.Sorted = true;
            comboBox1.SelectedIndex = 0;

        }
    }

    private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            comboBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            textBox4.Enabled = false;
            textBox5.Enabled = false;
            textBox6.Enabled = false;
            button1.Visible= false;
        }
        else
        {
            mode = "edit";
            comboBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox3.Enabled = true;
            textBox4.Enabled = true;
            textBox5.Enabled = true;
            textBox6.Enabled = true;
            button1.Visible = true;
        }
    }
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if (mode == "edit")
        {
            comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
            textBox5.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
            textBox6.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
            old_id = Convert.ToInt32(comboBox1.Text);
        }
        else if (mode == "delete")
        {
            if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query = $"delete from Monitor where cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";
```

```
                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into Monitor
values({Convert.ToInt32(comboBox1.Text)}, '{textBox2.Text}', {Convert.ToInt32(textBox3.Text)},
'{textBox4.Text}', '{textBox5.Text}', '{textBox6.Text}')";
                            query = $"insert into Monitor values(@cod_prod, @Matrix_type,
@Diagonal, @Monitor_type, @Matrix_cover, @Applic)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@Matrix_type", textBox2.Text);
                            command.Parameters.AddWithValue("@Diagonal", textBox3.Text);
                            command.Parameters.AddWithValue("@Monitor_type", textBox4.Text);
                            command.Parameters.AddWithValue("@Matrix_cover", textBox5.Text);
                            command.Parameters.AddWithValue("@Applic", textBox6.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Monitor set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, Tatrix_type = '{textBox2.Text}', Diagonal =
{Convert.ToInt32(textBox3.Text)}, Monitor_type = '{textBox4.Text}', Matrix_size =
'{textBox5.Text}', Applic = '{textBox6.Text}' where cod_prod = {old_id}";
                            query = $"update Monitor set cod_prod = @cod_prod, Tatrix_type =
@Matrix_type, Diagonal = @Diagonal, Monitor_type = @Monitor_type, Matrix_size = @Matrix_cover,
Applic = @Applic where cod_prod = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@Matrix_type", textBox2.Text);
```

```
                        command.Parameters.AddWithValue("@Diagonal", textBox3.Text);
                        command.Parameters.AddWithValue("@Monitor_type", textBox4.Text);
                        command.Parameters.AddWithValue("@Matrix_cover", textBox5.Text);
                        command.Parameters.AddWithValue("@Applic", textBox6.Text);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                        //Close();
                    }
                    else if (mode == "delete")
                    {
                        //Close();
                    }
                }
                if (mode != "add")
                    updateDG();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }


    private void updateDG()
    {
        try
        {
            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            using (connection)
            {
                connection.Open();

                string query = "select * from Monitor";
                SqlCommand command = new SqlCommand(query, connection);

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Visible = true;
                dataGridView1.ColumnCount = 6;
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                    dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                dataGridView1.Columns[1].HeaderCell.Value = "Matrix type";
                dataGridView1.Columns[2].HeaderCell.Value = "Diagonal";
```

```
                    dataGridView1.Columns[3].HeaderCell.Value = "Monitor type";
                    dataGridView1.Columns[4].HeaderCell.Value = "Screen size";
                    dataGridView1.Columns[5].HeaderCell.Value = "Application";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

ManEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class ManEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int max_id;
        private int old_id;

        public ManEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
```

```
            checkBox1.ForeColor = Color.White;
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
        }

        if(mode == "add")
        {
            this.Width = 365;
            checkBox1.Visible = false;
        } else if(mode == "edit")
        {
            button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
            checkBox1.Visible = true;
            updateDG();

            //fill the fields with the first line
        }

    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if(((int)e.KeyChar)!=8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox4.Text + e.KeyChar, "^\\-
?\\d+\\.?\\d*$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            mode = "delete";
            textBox1.Enabled= false;
            textBox2.Enabled= false;
            textBox3.Enabled= false;
            textBox4.Enabled= false;
            button1.Visible = false;
        }
        else
        {
            mode = "edit";
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox3.Enabled = true;
            textBox4.Enabled = true;
            button1.Visible = true;
        }
    }

    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if(mode == "edit")
        {
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
            old_id = Convert.ToInt32(textBox1.Text);
        } else if( mode == "delete")
        {
```

```
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from manufacturer where Man_code =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if(mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }

            if(mode!="delete" && Convert.ToInt32(textBox4.Text) < 0)
            {
                error.SetError(textBox4, "Price shoud be more than 0!");
                flag = false;
            }

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if(mode == "add")
                        {
                            //query = $"insert into manufacturer
values({Convert.ToInt32(textBox1.Text)}, '{textBox2.Text}', '{textBox3.Text}',
{Convert.ToInt32(textBox4.Text)})";
                            query = "insert into manufacturer values(@Man_code, @Man_name,
@Man_Adres, @Man_Production_price)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Man_code", textBox1.Text);
```

```
                            command.Parameters.AddWithValue("@Man_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Man_Adres", textBox3.Text);
                            command.Parameters.AddWithValue("@Man_Production_price",
textBox4.Text);


                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if(mode == "edit")
                        {
                            //query = $"update manufacturer set Man_code =
{Convert.ToInt32(textBox1.Text)}, Man_name = '{textBox2.Text}', Man_Adres = '{textBox3.Text}',
Man_Production_price = {Convert.ToInt32(textBox4.Text)} where Man_code = {old_id}";
                            query = $"update manufacturer set Man_code = @Man_code, Man_name =
@Man_name, Man_Adres = @Man_Adres, Man_Production_price = @Man_Production_price where Man_code =
{old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Man_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Man_name", textBox2.Text);
                            command.Parameters.AddWithValue("@Man_Adres", textBox3.Text);
                            command.Parameters.AddWithValue("@Man_Production_price",
textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                            //Close();
                        }
                        else if(mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if(mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();


                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();

                    string query = "select max(Man_code) from manufacturer";
                    SqlCommand command = new SqlCommand(query, connection);
```

```
                    max_id = (int)command.ExecuteScalar();


                    query = "select * from manufacturer";
                    command.CommandText = query;

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 4;
                    int i = 0;

                    while (reader.Read())
                    {
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }

                    dataGridView1.Columns[0].HeaderCell.Value = "Manufacturer id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Manufacturer name";
                    dataGridView1.Columns[2].HeaderCell.Value = "Manufacturer adres";
                    dataGridView1.Columns[3].HeaderCell.Value = "Production price";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

ConsignmentReportForm.cs

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class ConsignmentReportForm : Form
    {
        DataSet ds;
        public ConsignmentReportForm(DataSet ds)
        {
            InitializeComponent();
            this.ds = ds;
        }

        private void ConsignmentReportForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'computer_magazineDataSet.consignment_view' table. You can move, or remove it, as needed.
```

```
            this.consignment_viewTableAdapter.Fill(this.computer_magazineDataSet.consignment_view
);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"computer_magazineDataSet.consignment_view". При необходимости она может быть перемещена или
удалена.
            this.consignment_viewTableAdapter.Fill(this.computer_magazineDataSet.consignment_view
);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"computer_magazineDataSet.consignment_view". При необходимости она может быть перемещена или
удалена.
            this.consignment_viewTableAdapter.Fill(this.computer_magazineDataSet.consignment_view
);

            this.reportViewer1.RefreshReport();

            //ReportDataSource reportDataSource = new ReportDataSource("consignmentDataSource",
ds.Tables[0]);

            //reportViewer1.LocalReport.DataSources.Clear();
            //reportViewer1.LocalReport.DataSources.Add(reportDataSource);
        }
    }
}
```

ComputerEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class ComputerEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;

        public ComputerEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                label5.ForeColor = Color.Black;
                label6.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
```

```
            this.BackColor = Color.FromArgb(35, 35, 35);
            label1.ForeColor = Color.White;
            label2.ForeColor = Color.White;
            label3.ForeColor = Color.White;
            label4.ForeColor = Color.White;
            label5.ForeColor = Color.White;
            label6.ForeColor = Color.White;
            checkBox1.ForeColor = Color.White;
            dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
        }

        if (mode == "add")
        {
            this.Width = 265;
            checkBox1.Visible = false;
        }
        else if (mode == "edit")
        {
            button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
            checkBox1.Visible = true;
            updateDG();

            //fill the fields with the first line
        }

        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";


        using (connection)
        {
            connection.Open();
            SqlCommand command = new SqlCommand("select cod_prod from Product where Cod_type = 2000 or
Cod_type = 2010", connection); //2000 - PC-es id; 2010 - Laptops id

            SqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                comboBox1.Items.Add(reader.GetValue(0).ToString());
            }

            comboBox1.Sorted = true;
            comboBox1.SelectedIndex = 0;

        }

    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox2.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox3.Text + e.KeyChar, "^\\d+$"))
        {
            e.Handled = true;
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
```

```
                mode = "delete";
                comboBox1.Enabled = false;
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                textBox3.Enabled = false;
                textBox4.Enabled = false;
                checkBox2.Enabled = false;
                button1.Visible = false;
            }
            else
            {
                mode = "edit";
                comboBox1.Enabled = true;
                textBox1.Enabled = true;
                textBox2.Enabled = true;
                textBox3.Enabled = true;
                textBox4.Enabled = true;
                checkBox2.Enabled = true;
                button1.Visible = true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit")
            {
                comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
                textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
                if(dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString().ToLower() == "true")
                {
                    checkBox2.Checked = true;
                }
                else
                {
                    checkBox2.Checked = false;
                }

                textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
                old_id = Convert.ToInt32(comboBox1.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Computer where cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;
```

```
            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            query = $"insert into Computer values(@cod_prod, @CPU_freq, @RAM, @HDD, @CD,
@Applic)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@CPU_freq", textBox1.Text);
                            command.Parameters.AddWithValue("@RAM", textBox2.Text);
                            command.Parameters.AddWithValue("@HDD", textBox3.Text);
                            command.Parameters.AddWithValue("@CD",
checkBox2.Checked.ToString().ToLowerInvariant());
                            command.Parameters.AddWithValue("@Applic", textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
                            Close();
                        }
                        else if (mode == "edit")
                        {
                            //query = $"update Phone set cod_prod = {Convert.ToInt32(comboBox1.Text)},
CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM = {Convert.ToInt32(textBox2.Text)}, Intern_mem =
{Convert.ToInt32(textBox3.Text)}, Applic = '{textBox4.Text}' where cod_prod = {old_id}";
                            query = $"update Computer set cod_prod = @cod_prod, CPU_freq = @CPU_freq, RAM =
@RAM, HDD = @HDD, CD = @CD, Applic = @Applic where cod_prod = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@cod_prod", comboBox1.Text);
                            command.Parameters.AddWithValue("@CPU_freq", textBox1.Text);
                            command.Parameters.AddWithValue("@RAM", textBox2.Text);
                            command.Parameters.AddWithValue("@HDD", textBox3.Text);
                            command.Parameters.AddWithValue("@CD",
checkBox2.Checked.ToString().ToLowerInvariant());
                            command.Parameters.AddWithValue("@Applic", textBox4.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows affected!");
                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }


        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();
```

```
                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();

                    string query = "select * from Computer";
                    SqlCommand command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 6;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                        dataGridView1.Rows[i].Cells[4].Value = reader.GetValue(4);
                        dataGridView1.Rows[i].Cells[5].Value = reader.GetValue(5);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Product id";
                    dataGridView1.Columns[1].HeaderCell.Value = "CPU frequency";
                    dataGridView1.Columns[2].HeaderCell.Value = "RAM";
                    dataGridView1.Columns[3].HeaderCell.Value = "HDD";
                    dataGridView1.Columns[4].HeaderCell.Value = "CD";
                    dataGridView1.Columns[5].HeaderCell.Value = "Application";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

ChekInfoChekEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class ChekInfoChekEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
```

```
        private int old_prod_id;

        public ChekInfoChekEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                label4.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                label4.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }

            if (mode == "add")
            {
                this.Width = 305;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";


            using (connection)
            {
                connection.Open();
                SqlCommand command = new SqlCommand("select Check_code from Chek", connection);

                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    comboBox1.Items.Add(reader.GetValue(0).ToString());
                }

                comboBox1.Sorted = true;
                comboBox1.SelectedIndex = 0;

                command = new SqlCommand("select cod_prod from Product", connection);

                reader.Close();

                reader = command.ExecuteReader();
```

```
                while (reader.Read())
                {
                    comboBox2.Items.Add(reader.GetValue(0).ToString());
                }

                comboBox2.Sorted = true;
                comboBox2.SelectedIndex = 0;

            }

            if (mode == "edit-only")
            {
                checkBox1.Visible = false;
                updateDG();
            }

        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
            {
                e.Handled = true;
            }
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                mode = "delete";
                comboBox1.Enabled = false;
                comboBox2.Enabled = false;
                textBox1.Enabled = false;
                dateTimePicker1.Enabled = false;
                button1.Visible = false;
            }
            else
            {
                mode = "edit";
                comboBox1.Enabled = true;
                comboBox2.Enabled = true;
                textBox1.Enabled = true;
                dateTimePicker1.Enabled = true;
                button1.Visible = true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit" || mode == "edit-only")
            {
                comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                comboBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                dateTimePicker1.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
                old_id = Convert.ToInt32(comboBox1.Text);
                old_prod_id = Convert.ToInt32(comboBox2.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
```

```
                        {
                            connection.Open();
                            string query = $"delete from Chek_infoChek where Check_code =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value} and cod_prod =
{dataGridView1.Rows[e.RowIndex].Cells[1].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            //query = $"insert into Monitor
values({Convert.ToInt32(comboBox1.Text)}, '{textBox2.Text}', {Convert.ToInt32(textBox3.Text)},
'{textBox4.Text}', '{textBox5.Text}', '{textBox6.Text}')";
                            query = $"insert into Chek_infoChek values(@Check_code, @cod_prod,
@Purc_date, @Gen_price)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Check_code", comboBox1.Text);
                            command.Parameters.AddWithValue("@cod_prod", comboBox2.Text);
                            command.Parameters.AddWithValue("@Purc_date", dateTimePicker1.Value);
                            command.Parameters.AddWithValue("@Gen_price", textBox1.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit" || mode == "edit-only")
                        {
                            //query = $"update Monitor set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, Tatrix_type = '{textBox2.Text}', Diagonal =
{Convert.ToInt32(textBox3.Text)}, Monitor_type = '{textBox4.Text}', Matrix_size =
'{textBox5.Text}', Applic = '{textBox6.Text}' where cod_prod = {old_id}";
                            query = $"update Chek_infoChek set Check_code = @Check_code, cod_prod
= @cod_prod,  Purc_date = @Purc_date, Gen_price = @Gen_price where Check_code = {old_id} and
cod_prod = {old_prod_id}";
```

```
                        SqlCommand command = new SqlCommand(query, connection);

                        command.Parameters.AddWithValue("@Check_code", comboBox1.Text);
                        command.Parameters.AddWithValue("@cod_prod", comboBox2.Text);
                        command.Parameters.AddWithValue("@Purc_date", dateTimePicker1.Value);
                        command.Parameters.AddWithValue("@Gen_price", textBox1.Text);

                        MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");


                        //Close();
                    }
                    else if (mode == "delete")
                    {
                        //Close();
                    }
                }
                if (mode != "add")
                    updateDG();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }


    private void updateDG()
    {
        try
        {
            dataGridView1.Visible = true;
            dataGridView1.Rows.Clear();
            dataGridView1.Columns.Clear();


            connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
            using (connection)
            {
                connection.Open();

                string query = "select * from Chek_infoChek";
                SqlCommand command = new SqlCommand(query, connection);

                SqlDataReader reader = command.ExecuteReader();

                dataGridView1.Visible = true;
                dataGridView1.ColumnCount = 4;
                int i = 0;

                while (reader.Read())
                {
                    //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                    dataGridView1.RowCount++;
                    dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                    dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                    dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                    dataGridView1.Rows[i].Cells[3].Value = reader.GetValue(3);
                    dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                    i++;
                }
                dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                dataGridView1.Columns[1].HeaderCell.Value = "Product id";
                dataGridView1.Columns[2].HeaderCell.Value = "Purchasing date";
```

```
                    dataGridView1.Columns[3].HeaderCell.Value = "Total price";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

ChekEditForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Computer_magazine
{
    public partial class ChekEditForm : Form
    {
        private SqlConnection connection = new SqlConnection("Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True");
        private ErrorProvider error;
        private Form caller;
        private string mode;
        private int old_id;
        private int max_id;

        public ChekEditForm(Form caller, string mode)
        {
            InitializeComponent();

            error = new ErrorProvider();
            this.caller = caller;
            this.mode = mode;

            if (((Form1)caller).getStyle().Equals("white"))
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                label2.ForeColor = Color.Black;
                label3.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                dataGridView1.BackgroundColor = Color.WhiteSmoke;
            }
            else
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
                label3.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                dataGridView1.BackgroundColor = Color.FromArgb(35, 35, 35);
            }
```

```
            if (mode == "add")
            {
                this.Width = 245;
                checkBox1.Visible = false;
            }
            else if (mode == "edit")
            {
                button1.BackgroundImage = Image.FromFile("../../resources/pencil.png");
                checkBox1.Visible = true;
                updateDG();

                //fill the fields with the first line
            }

            try
            {

                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";

                using (connection)
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand("select max(Check_code) from Chek",
connection);

                    max_id = Convert.ToInt32(command.ExecuteScalar());



                    command = new SqlCommand("select Pay_code from Payment", connection);

                    SqlDataReader reader = command.ExecuteReader();

                    while (reader.Read())
                    {
                        comboBox1.Items.Add(reader.GetValue(0).ToString());
                    }

                    reader.Close();


                    command = new SqlCommand("select Client_id from Pasport_data", connection);

                    reader = command.ExecuteReader();

                    while (reader.Read())
                    {
                        comboBox2.Items.Add(reader.GetValue(0));
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

            if (mode == "edit-only")
            {
                checkBox1.Visible = false;
                updateDG();
            }

        }

        private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
        {
```

```
            if (((int)e.KeyChar) != 8 && !Regex.IsMatch(textBox1.Text + e.KeyChar, "^\\d+$"))
            {
                e.Handled = true;
            }
        }


        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                mode = "delete";
                textBox1.Enabled = false;
                comboBox1.Enabled = false;
                button1.Visible= false;
            }
            else
            {
                mode = "edit";
                textBox1.Enabled = true;
                comboBox1.Enabled = true;
                button1.Visible = true;
            }
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            if (mode == "edit" || mode == "edit-only")
            {
                textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();
                comboBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
                comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
                old_id = Convert.ToInt32(textBox1.Text);
            }
            else if (mode == "delete")
            {
                if (MessageBox.Show("Are you sure?", "Confirm", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    try
                    {
                        connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                        using (connection)
                        {
                            connection.Open();
                            string query = $"delete from Chek where Check_code =
{dataGridView1.Rows[e.RowIndex].Cells[0].Value}";

                            SqlCommand command = new SqlCommand(query, connection);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");
                        }

                        updateDG();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            error.Clear();
            bool flag = true;
```

```
            if (mode == "add" && Convert.ToInt32(textBox1.Text) <= max_id)
            {
                error.SetError(textBox1, $"Id shoud be more than {max_id}");
                flag = false;
            }

            if (flag)
            {
                try
                {
                    connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                    using (connection)
                    {
                        connection.Open();
                        string query;

                        if (mode == "add")
                        {
                            query = $"insert into Chek values(@Check_code, @Client_id,
@Pay_code)";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Check_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Client_id", comboBox2.Text);
                            command.Parameters.AddWithValue("@Pay_code", comboBox1.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            Close();
                        }
                        else if (mode == "edit" || mode == "edit-only")
                        {
                            //query = $"update Phone set cod_prod =
{Convert.ToInt32(comboBox1.Text)}, CPU_freq = {Convert.ToInt32(textBox1.Text)}, RAM =
{Convert.ToInt32(textBox2.Text)}, Intern_mem = {Convert.ToInt32(textBox3.Text)}, Applic =
'{textBox4.Text}' where cod_prod = {old_id}";
                            query = $"update Chek set Check_code = @Check_code,
Client_id=@Client_id,  Pay_code = @Pay_code where Check_code = {old_id}";

                            SqlCommand command = new SqlCommand(query, connection);

                            command.Parameters.AddWithValue("@Check_code", textBox1.Text);
                            command.Parameters.AddWithValue("@Client_id", comboBox2.Text);
                            command.Parameters.AddWithValue("@Pay_code", comboBox1.Text);

                            MessageBox.Show($"Success! {command.ExecuteNonQuery()} rows
affected!");

                            //Close();
                        }
                        else if (mode == "delete")
                        {
                            //Close();
                        }
                    }
                    if (mode != "add")
                        updateDG();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
            }
        }
```

```
        private void updateDG()
        {
            try
            {
                dataGridView1.Visible = true;
                dataGridView1.Rows.Clear();
                dataGridView1.Columns.Clear();


                connection.ConnectionString = "Data Source=.\\SQLEXPRESS;Initial
Catalog=Computer_magazine;Integrated Security=True";
                using (connection)
                {
                    connection.Open();


                    string query = "select max(Check_code) from Chek";
                    SqlCommand command = new SqlCommand(query, connection);

                    max_id = (int)command.ExecuteScalar();

                    query = "select * from Chek";
                    command = new SqlCommand(query, connection);

                    SqlDataReader reader = command.ExecuteReader();

                    dataGridView1.Visible = true;
                    dataGridView1.ColumnCount = 3;
                    int i = 0;

                    while (reader.Read())
                    {
                        //MessageBox.Show($"{reader.GetValue(0)}  {reader.GetValue(1)}
{reader.GetValue(2)}    {reader.GetValue(3)}");
                        dataGridView1.RowCount++;
                        dataGridView1.Rows[i].Cells[0].Value = reader.GetValue(0);
                        dataGridView1.Rows[i].Cells[1].Value = reader.GetValue(1);
                        dataGridView1.Rows[i].Cells[2].Value = reader.GetValue(2);
                        dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();
                        i++;
                    }
                    dataGridView1.Columns[0].HeaderCell.Value = "Check id";
                    dataGridView1.Columns[1].HeaderCell.Value = "Client id";
                    dataGridView1.Columns[2].HeaderCell.Value = "Payment code";
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

ChekReportForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class CheckReportForm : Form
    {
        public CheckReportForm()
        {
            InitializeComponent();
        }

        private void CheckReportForm_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'computer_magazineDataSet.Chek_view'
table. You can move, or remove it, as needed.
            this.chek_viewTableAdapter1.Fill(this.computer_magazineDataSet.Chek_view);
            //// TODO: данная строка кода позволяет загрузить данные в таблицу
"computer_magazineDataSet.Chek_view". При необходимости она может быть перемещена или удалена.
            //this.chek_viewTableAdapter.Fill(this.computer_magazineDataSet.Chek_view);

            this.reportViewer1.RefreshReport();
        }
    }
}
```

StyleForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Computer_magazine
{
    public partial class StyleForm : Form
    {
        Form caller;
        public StyleForm(Form caller)
        {
            InitializeComponent();
            this.caller = caller;

            if (((Form1)this.caller).getStyle() == "white")
            {
                checkBox1.Checked = false;
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
            }
            else
            {
                checkBox1.Checked = true;
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
            };
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
```

```
            if (checkBox1.Checked)
            {
                this.BackColor = Color.FromArgb(35, 35, 35);
                label1.ForeColor = Color.White;
                checkBox1.ForeColor = Color.White;
                ((Form1)this.caller).setStyle("dark");
            } else
            {
                this.BackColor = Color.WhiteSmoke;
                label1.ForeColor = Color.Black;
                checkBox1.ForeColor = Color.Black;
                ((Form1)this.caller).setStyle("white");
            }
        }
    }
}
```