**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по учебной практике УП.02.01 (по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/2 группы

Тихонова Валерия Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «23» сентября 2024 г. по «05» октября 2024 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на учебную практику (по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/2 группы

Тихонова Валерия Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «23» сентября 2024 г. по «05» октября 2024 г.

**Виды работ, обязательные для выполнения** *(переносится из программы, соответствующего ПМ):*

Участие в выработке требований к программному обеспечению;

Стадии проектирования программного обеспечения;

Разработка модулей программного обеспечения;

Тестирование программных модулей и их интеграции;

Разработка программной документации и стандарты кодирования.

**Индивидуальное задание: ВАРИАНТ 19**

Задание выдал «23» сентября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Хисамутдинова А.С.

(подпись) (Ф.И.О.)

Задание получил «23» сентября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_ Тихонова В.А. (подпись) (Ф.И.О.)

**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения учебной практики УП.02.01**

**(по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/2 группы

Тихонова Валерия Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «23» сентября 2024 г. по «05» октября 2024 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике

Санкт-Петербург

2024

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| **Дата** | **Виды выполненных работ и заданий по программе практики** | **Подпись руководителя практики** |
| **1** | **2** | **3** |
| 23.09.2024 | Анализ предметной области. Выявление требований к программе. |  |
| 24.09.2024 | Разработка технического задания. UML. Проектирование диаграммы вариантов использования. |  |
| 25.09.2024 | UML. Проектирование диаграммы последовательности. UML. Проектирование диаграммы активности. |  |
| 26.09.2024 | Моделирование структуры ПО. Проектирование инфологической и даталогической модели данных. |  |
| 27.09.2024 | Проектирование интерфейса пользователя. Создание Wireframe эскизов. Разработка дизайна программы в соответствии с руководством по стилю. |  |
| 28.09.2024 | Разработка базы данных. Разработка словаря данных. |  |
| 30.09.2024 | Создание приложения. Форма авторизации. Создание приложения. Форма заказов. |  |
| 01.10.2024 | Создание приложения. Основные формы приложения. Разработка библиотеки классов. |  |
| 02.10.2024 | Подготовка отчетов и выгрузка документов для печати. Отладка программных модулей. |  |
| 03.10.2024 | Модульное тестирование. Создание тестовых случаев. |  |
| 04.10.2024 | Интеграционное тестирование. Разработка самодокументирующегося кода. |  |
| 05.10.2024 | Инспекция кода на соответствие стандартам кодирования. Подготовка отчета и размещение результатов в репозитории контроля версий. |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc178841222)

[Задание № 1 “Разработка технического задания” 5](#_Toc178841223)

[Задание № 2 “Разработка алгоритмов и диаграмм” 5](#_Toc178841224)

[Задание 3. Разработка макетов программы 5](#_Toc178841225)

[Задание 4. Разработка программы 5](#_Toc178841226)

[Задание 5. Тестирование 5](#_Toc178841227)

[ЗАКЛЮЧЕНИЕ 5](#_Toc178841228)

ВВЕДЕНИЕ

В современном мире информационных технологий программирование и разработка баз данных играют ключевую роль в создании эффективных и надежных систем. Учебная практика, проведенная в рамках нашего курса, была направлена на приобретение практических навыков в области разработки программного обеспечения, работы с базами данных, тестирования и создания диаграмм и макетов.

Целью данной практики было не только углубить теоретические знания, полученные в ходе учебного процесса, но и применить их на практике, создав полноценное программное решение. В рамках практики я разработала программу, интегрировала ее с базой данных, провела тестирование, создала диаграммы и макеты, что позволило мне лучше понять все этапы разработки программного обеспечения.

Задание № 1 “Разработка технического задания”

1. Предметная область, которую Анастасия составила для меня по теме “Поставка товаров”:

Основной целью данной информационной системы является автоматизация процесса управления поставками товаров (цветов). Она позволит компании эффективно управлять заказами, отслеживать статус поставок и оптимизировать логистику.

Разработка программы управления поставками товаров направлена на создание удобного и эффективного инструмента, который помогает минимизировать задержки и обеспечивает точность учета.

Задача разработки такой программы состоит в создании системы, которая позволяет компаниям эффективно отслеживать и управлять поставками товаров, что способствует бесперебойной работе и оптимизации использования ресурсов.

Система позволяет компании сократить время обработки заказов, оптимизировать использование ресурсов и повысить удовлетворенность клиентов.

Основные функции и возможности модуля включают:

1. Создание заказа:

Это информация, предоставленная клиентом о товарах, которые требуется поставить. Клиент оставляет новый заказ. Заказ может содержать данные о наименовании товара, количестве, сроках поставки, личной информации (ФИО клиента и контактные данные). Также должны содержаться сведения об идентификаторе заказа

1. Обработка заказа:

Процесс, включающий анализ заказа и назначение исполнителя (менеджера по поставкам) для выполнения. В процессе обработки может потребоваться дополнительная информация или уточнение деталей заказа у клиента.

1. Исполнение заказа:

Фактическое выполнение поставки товаров. На этом этапе назначенный менеджер по поставкам организует доставку товаров, вносит необходимые изменения или координирует работу с другими специалистами.

1. Отчётность:

Фиксация и отчёт о выполненной работе. После завершения поставки менеджер по поставкам должен предоставить отчёт о проделанной работе, включая информацию о затраченных ресурсах (время, транспортные расходы) и оказанной помощи.

1. Мониторинг и анализ:

Этот этап предполагает контроль и анализ процесса управления поставками. Важно отслеживать и анализировать время обработки заказов, качество выполненных работ и расходы, которые могут помочь в оптимизации процесса.

1. Техническое задание для моей предметной области

1. Общие сведения

1.1. Наименование проекта: Разработка программного модуля для управления поставками товаров.

1.2. Заказчик: Компания «SPORTX».

1.3. Исполнитель: ИТ-компания «Торговый дом “Цветочек”».

2. Функциональные требования

2.1 Возможность входа под разными ролями (администратор, менеджер, клиент)

2.2. Возможность добавления заказов в базу данных с указанием следующих параметров:

* Номер заказа;
* Дата добавления;
* Наименование товара;
* Количество товара;
* Сроки поставки;
* ФИО клиента;
* Контактные данные клиента;
* Статус заказа (новый заказ, в обработке, в процессе поставки, завершен).

2.3. Возможность редактирования заказов:

* Изменение этапа выполнения (в обработке, в процессе поставки, завершен);
* Изменение сроков поставки;
* Изменение ответственного за выполнение работ.

2.4. Возможность отслеживания статуса заказа:

* Отображение списка заказов;
* Поиск заказа по номеру.

2.5. Возможность добавления менеджера по поставкам к заказу;

2.6. Расчет статистики работы отдела поставок:

* Количество выполненных заказов;

3. Нефункциональные требования

3.1. Поддержка работы на ОС семейства Windows.

3.2. Безопасность:

* Логин и пароль для доступа к приложению;
* Доступ к данным должен быть ограничен в зависимости от роли пользователя.

3.3. Удобство использования:

* Простой интерфейс;

3.4. Производительность:

* Приложение должно иметь быстрый доступ к данным;
* Минимальное время отклика на запросы пользователя.

4. Требования к реализации

4.1. Язык программирования: C#.

4.2. СУБД: SQL Server.

5. Требования к документации

5.1. Техническое задание на разработку программного модуля.

6. Руководство по стилю

6.1. Шрифт: Comic Sans 11 и 10;

6.2. Цветовая схема: цветовая гамма приложения должна быть в следующих оттенках:

* DarkViolet #9400D3
* MediumPurple #9370DB
* BlueViolet #8A2BE2
* Green #008000
* DarkGreen #006400
* YellowGreen #9ACD32

6.3 В свободном пространстве должны располагаться картинки цветов и бабочек;

Задание № 2 “Разработка алгоритмов и диаграмм”

По составленному техническому заданию необходимо разработать алгоритмы в виде блок-схем и UML-диаграммы:

* диаграммы вариантов использования;

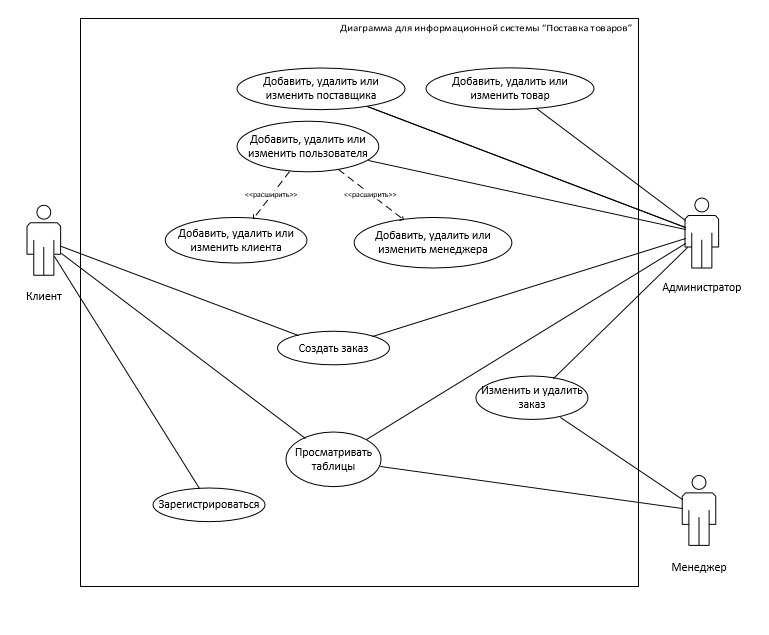


Рисунок 1 – Диаграмма вариантов использования

* диаграммы последовательности;

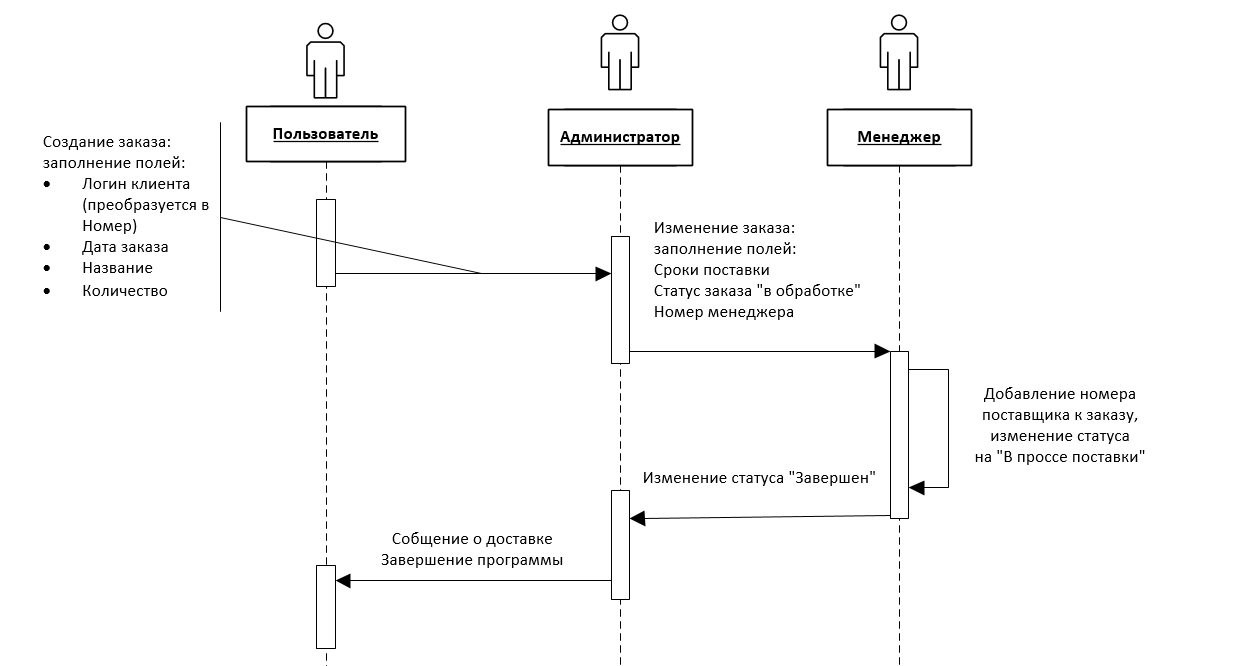


Рисунок 2 – Диаграмма последовательностей

* диаграммы активности.



Рисунок 3 – Диаграмма активностей

ER-диаграмма для БД + словарь данных.

Рисунок 5 – ER-диаграмма

Таблицы:

Заказы

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Номер\_заказа | int | Первичный ключ |
| Номер\_клиента | Int |  |
| Дата\_добавления | nvarchar(20) | Дата создания заказа |
| Название\_товара | nvarchar(20) |  |
| Количество\_товара | nvarchar(20) |  |
| Сроки\_поставки | nvarchar(20) |  |
| Статус\_заказа | nvarchar(20) | (в обработке, в процессе поставки, завершен) |
| Номер\_менеджера | int |  |
| Номер\_поставщика | nvarchar(20) |  |

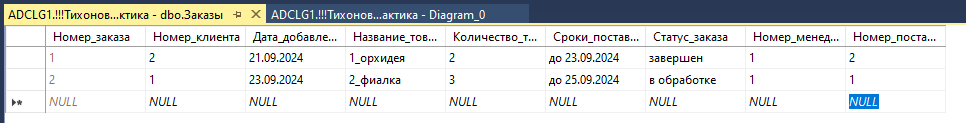


Рисунок 5 – Таблица Заказы

Клиенты

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Номер\_клиента | int | Первичный ключ |
| Логин\_клиента | nvarchar(20) |  |
| Телефон | nvarchar(20) | Для возможной связи |

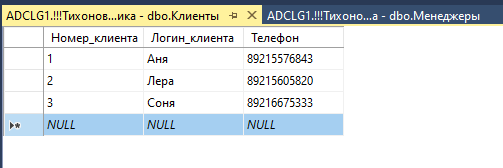


Рисунок 6 – Таблица Клиенты

Менеджеры

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Номер\_менеджер | int | Первичный ключ |
| Логин\_менеджера | nvarchar(20) |  |
| Имя\_менеджера | nvarchar(50) |  |
| Оклад | nvarchar(20) | Зарплата |
| Дата\_приема\_на\_работу | nvarchar(20) |  |

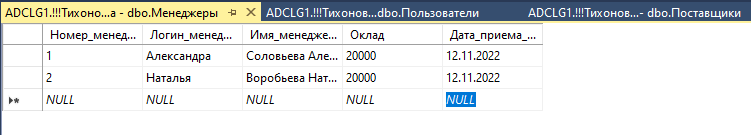


Рисунок 7 – Таблица Менеджеры

Пользователи

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Логин | nvarchar(20) | Первичный ключ |
| Пароль | nvarchar(20) |  |
| Роль | nvarchar(20) | Администратор, пользователь или менеджер |

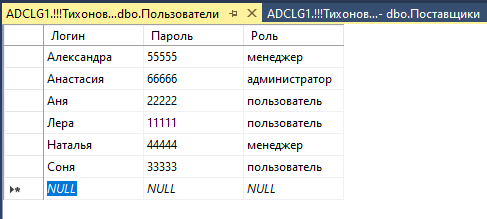


Рисунок 8 – Таблица Пользователи

Поставщики

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Номер\_поставщика | nvarchar(20) | Первичный ключ |
| Имя\_поставщика | nvarchar(50) | Обычно это ООО |
| Номер\_автомобиля | nvarchar(20) |  |

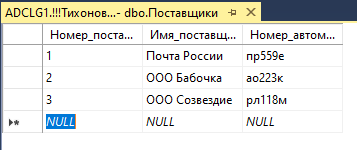


Рисунок 9 – Таблица Поставщики

Товары

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Дополнительное описание |
| Название\_товара | nvarchar(20) | Так как видов одного растения может быть много, нумерация проходит в формате номер\_название. Первичный ключ |
| Полное\_название | nvarchar(50) | Наприме, Фиалка домашняя |
| Условия\_содержания | nvarchar(50) | Частота полива, свет и т.д. |

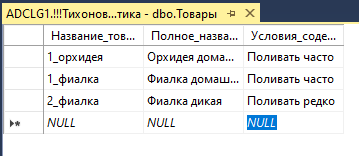


Рисунок 10 – Таблица Товары

Задание 3. Разработка макетов программы

Wireframe макеты

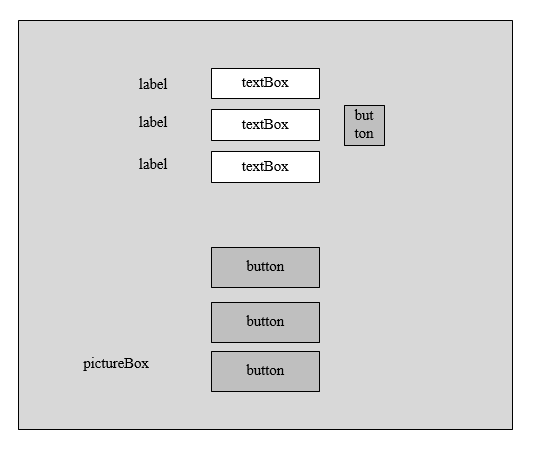


Рисунок 11 - Wireframe макет вкладки Авторизация

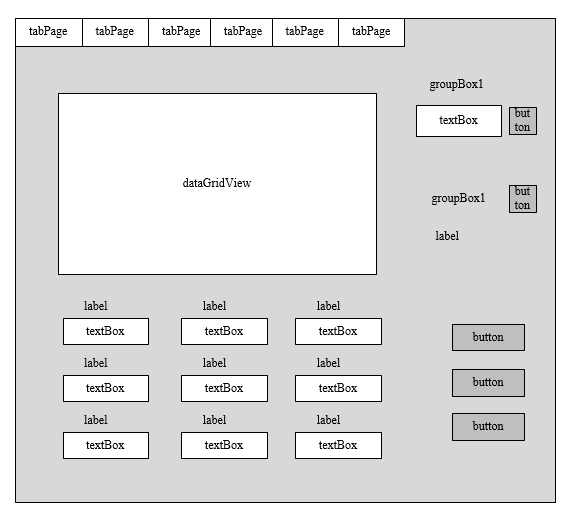


Рисунок 12 - Wireframe макет вкладки Заказы

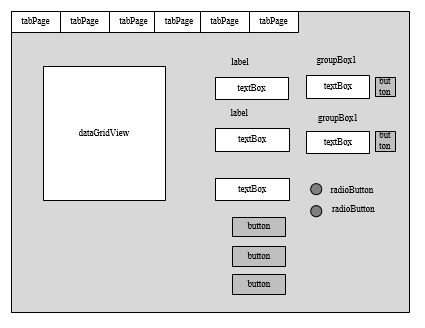


Рисунок 13 - Wireframe макет вкладки Клиенты

Mockup макеты

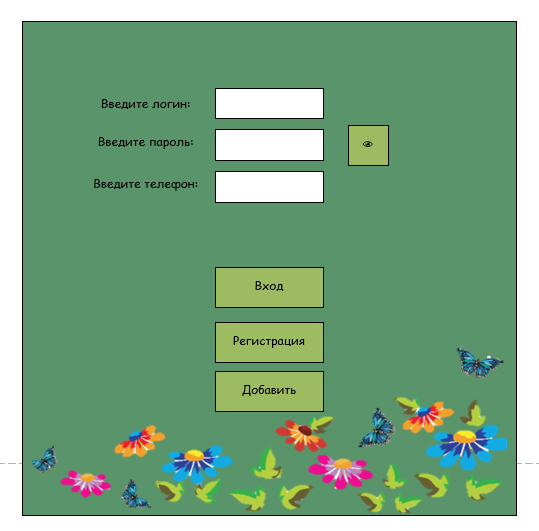


Рисунок 14 - Mockup макет вкладки Авторизация

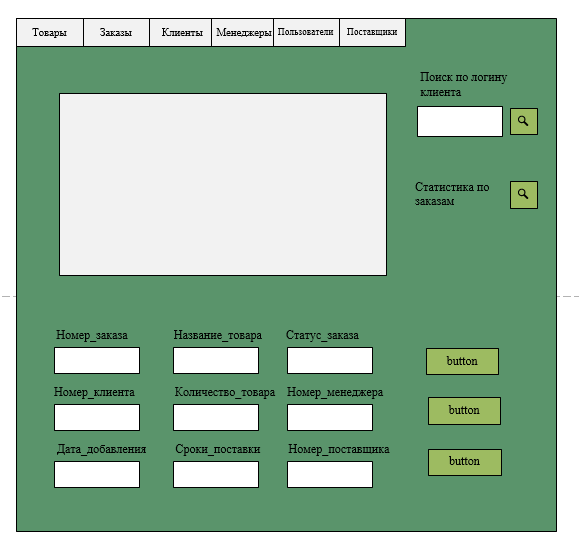


Рисунок 15 - Mockup макет вкладки Заказы

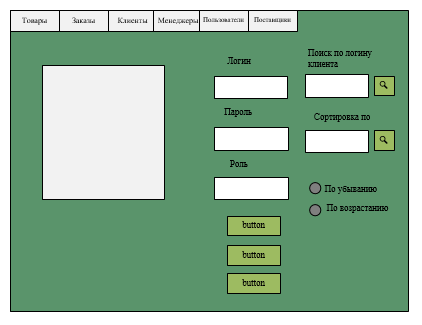


Рисунок 16 - Mockup макет вкладки Клиенты

Карта навигации приложения



Рисунок 17 – Карта навигации приложения

Задание 4. Разработка программы

Код Form1:

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.Threading.Tasks;

using System.Windows.Forms;

namespace \_17

{

public partial class Form1 : Form

{

Form2 form2;

public int a = 2;

string str = "";

//public bool Admin = form2.ADMIN;

public Form1()

{

InitializeComponent();

}

//bool admf = true;

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter2.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры". При необходимости она может быть перемещена или удалена.

this.менеджерыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики". При необходимости она может быть перемещена или удалена.

this.поставщикиTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы". При необходимости она может быть перемещена или удалена.

this.заказыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары". При необходимости она может быть перемещена или удалена.

this.товарыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары);

form2 = (Form2)this.Owner;

a = form2.ADMIN;

if (a==0)

{

panel1.Enabled = true;

panel2.Enabled = true;

panel3.Enabled = true;

panel4.Enabled = true;

panel5.Enabled = true;

panel6.Enabled = true;

}

else if (a == 1)

{

panel1.Enabled = true;

panel2.Enabled = true;

panel3.Enabled = false;

panel4.Enabled = true;

panel5.Enabled = false;

panel6.Enabled = false;

tabPage5.Parent = null;

tabPage6.Parent = null;

}

else

{

panel1.Enabled = false;

panel2.Enabled = true;

textBox28.Enabled = false;

textBox23.Enabled = false;

textBox22.Enabled = false;

textBox21.Enabled = false;

panel3.Enabled = false;

panel4.Enabled = false;

panel5.Enabled = false;

panel6.Enabled = false;

tabPage3.Parent = null;

tabPage4.Parent = null;

tabPage5.Parent = null;

tabPage6.Parent = null;

}

}

private void button4\_Click(object sender, EventArgs e) //добавить 1 стр

{

string name = Convert.ToString(textBox2.Text);

string fullname = Convert.ToString(textBox3.Text);

string yslovie = Convert.ToString(textBox5.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Товары([Название\_товара],[Полное\_название],[Условия\_содержания]) VALUES(@name,@fullname,@yslovie);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@fullname", fullname);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@yslovie", yslovie);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары". При необходимости она может быть перемещена или удалена.

this.товарыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары);

textBox2.Clear();

textBox3.Clear();

textBox5.Clear();

}

private void button1\_Click(object sender, EventArgs e) // изменить 1 стр

{

string name = Convert.ToString(textBox2.Text);

string fullname = Convert.ToString(textBox3.Text);

string yslovie = Convert.ToString(textBox5.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Товары set [Полное\_название]=@fullname, [Условия\_содержания]=@yslovie where [Название\_товара]=@name;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@fullname", fullname);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@yslovie", yslovie);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары". При необходимости она может быть перемещена или удалена.

this.товарыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары);

textBox2.Clear();

textBox3.Clear();

textBox5.Clear();

}

private void button2\_Click(object sender, EventArgs e) //удаление для первой таблицы

{

string yslovie = Convert.ToString(textBox2.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Delete from Товары where [Название\_товара]=@yslovie";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@yslovie", yslovie);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары". При необходимости она может быть перемещена или удалена.

this.товарыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet1.Товары);

textBox2.Clear();

textBox3.Clear();

textBox5.Clear();

}

private void button6\_Click(object sender, EventArgs e) //добавить для второй таблицы

{

string numberClient = textBox1.Text;

string date = textBox9.Text;

string name = textBox8.Text;

string count = textBox7.Text;

string time = textBox28.Text;

string status = textBox23.Text;

string numberMeneger = textBox22.Text;

string numberPost = textBox21.Text;

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Заказы([Номер\_клиента],[Дата\_добавления],[Название\_товара],[Количество\_товара],[Сроки\_поставки],[Статус\_заказа],[Номер\_менеджера],[Номер\_поставщика]) VALUES(@numberClient,@date,@name,@count,@time,@status,@numberMeneger,@numberPost);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@numberClient", numberClient);

cmd1.Parameters.AddWithValue("@date", date);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@count", count);

cmd1.Parameters.AddWithValue("@time", time);

cmd1.Parameters.AddWithValue("@status", status);

cmd1.Parameters.AddWithValue("@numberMeneger", numberMeneger);

cmd1.Parameters.AddWithValue("@numberPost", numberPost);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы". При необходимости она может быть перемещена или удалена.

this.заказыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы);

textBox4.Clear();

textBox1.Clear();

textBox9.Clear();

textBox8.Clear();

textBox7.Clear();

textBox28.Clear();

textBox23.Clear();

textBox22.Clear();

textBox21.Clear();

}

private void button5\_Click(object sender, EventArgs e) //изменить для второй таблицы

{

string numberZakaz = textBox4.Text;

string numberClient = textBox1.Text;

string date = textBox9.Text;

string name = textBox8.Text;

string count = textBox7.Text;

string time = textBox28.Text;

string status = textBox23.Text;

string numberMeneger = textBox22.Text;

string numberPost = textBox21.Text;

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Заказы set [Номер\_клиента]=@numberClient,[Дата\_добавления]=@date,[Название\_товара]=@name,[Количество\_товара]=@count,[Сроки\_поставки]=@time,[Статус\_заказа]=@status,[Номер\_менеджера]=@numberMeneger,[Номер\_поставщика]=@numberPost where [Номер\_заказа]=@numberZakaz;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@numberZakaz", numberZakaz);

cmd1.Parameters.AddWithValue("@numberClient", numberClient);

cmd1.Parameters.AddWithValue("@date", date);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@count", count);

cmd1.Parameters.AddWithValue("@time", time);

cmd1.Parameters.AddWithValue("@status", status);

cmd1.Parameters.AddWithValue("@numberMeneger", numberMeneger);

cmd1.Parameters.AddWithValue("@numberPost", numberPost);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы". При необходимости она может быть перемещена или удалена.

this.заказыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы);

textBox4.Clear();

textBox1.Clear();

textBox9.Clear();

textBox8.Clear();

textBox7.Clear();

textBox28.Clear();

textBox23.Clear();

textBox22.Clear();

textBox21.Clear();

}

private void button3\_Click(object sender, EventArgs e) //удалить для второй таблицы

{

int a = Convert.ToInt32(textBox4.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = " Delete from Заказы where [Номер\_заказа]=@a";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@a", a);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы". При необходимости она может быть перемещена или удалена.

this.заказыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet2.Заказы);

textBox4.Clear();

textBox1.Clear();

textBox9.Clear();

textBox8.Clear();

textBox7.Clear();

textBox28.Clear();

textBox23.Clear();

textBox22.Clear();

textBox21.Clear();

}

private void button7\_Click(object sender, EventArgs e) // поиск для первой таблицы

{

dataGridView1.ClearSelection();

string searchValue = textBox10.Text;

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView1.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView1.Refresh();

textBox10.Clear();

}

private void button9\_Click(object sender, EventArgs e) //поиск для второй

{

dataGridView3.ClearSelection();

string searchValue = textBox11.Text;

dataGridView3.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView3.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView3.Refresh();

textBox11.Clear();

}

private void button8\_Click(object sender, EventArgs e)

{

label10.Text = dataGridView3.RowCount.ToString();

}

//////////////////////// ТРЕТЬЯ ТАБЛИЦА

private void button20\_Click(object sender, EventArgs e) // добавить

{

string number = Convert.ToString(textBox14.Text);

string name = Convert.ToString(textBox15.Text);

string car = Convert.ToString(textBox6.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Поставщики([Номер\_поставщика],[Имя\_поставщика],[Номер\_автомобиля]) VALUES(@number,@name,@car);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@number", number);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@car", car);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики". При необходимости она может быть перемещена или удалена.

this.поставщикиTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики);

textBox14.Clear();

textBox15.Clear();

textBox6.Clear();

}

private void button19\_Click(object sender, EventArgs e) //изменить

{

string number = Convert.ToString(textBox14.Text);

string name = Convert.ToString(textBox15.Text);

string car = Convert.ToString(textBox6.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Поставщики set [Имя\_поставщика]=@name,[Номер\_автомобиля]=@car where [Номер\_поставщика]=@number;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@number", number);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@car", car);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики". При необходимости она может быть перемещена или удалена.

this.поставщикиTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики);

textBox14.Clear();

textBox15.Clear();

textBox6.Clear();

}

private void button18\_Click(object sender, EventArgs e) //удалить

{

string number = Convert.ToString(textBox14.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Delete from Поставщики where [Номер\_поставщика]=@number";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@number", number);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики". При необходимости она может быть перемещена или удалена.

this.поставщикиTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet3.Поставщики);

textBox14.Clear();

textBox15.Clear();

textBox6.Clear();

}

private void button17\_Click(object sender, EventArgs e) //поиск

{

dataGridView4.ClearSelection();

string searchValue = textBox13.Text;

dataGridView4.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView4.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView4.Refresh();

textBox13.Clear();

}

//////////////////////////////// ТАБЛИЦА 4

private void button15\_Click(object sender, EventArgs e) //добавить

{

//string num = Convert.ToString(textBox19.Text);

string login = Convert.ToString(textBox18.Text);

string name = Convert.ToString(textBox17.Text);

string ocl = Convert.ToString(textBox16.Text);

string time = Convert.ToString(textBox29.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Менеджеры([Логин\_менеджера],[Имя\_менеджера],[Оклад],[Дата\_приема\_на\_работу]) VALUES(@login,@name,@ocl,@time);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

//cmd1.Parameters.AddWithValue("@num", num);

cmd1.Parameters.AddWithValue("@login", login);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@ocl", ocl);

cmd1.Parameters.AddWithValue("@time", time);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры". При необходимости она может быть перемещена или удалена.

this.менеджерыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры);

textBox19.Clear();

textBox18.Clear();

textBox17.Clear();

textBox16.Clear();

textBox29.Clear();

}

private void button14\_Click(object sender, EventArgs e) //изменить

{

string num = Convert.ToString(textBox19.Text);

string login = Convert.ToString(textBox18.Text);

string name = Convert.ToString(textBox17.Text);

string ocl = Convert.ToString(textBox16.Text);

string time = Convert.ToString(textBox29.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Менеджеры set [Логин\_менеджера]=@login,[Имя\_менеджера]=@name,[Оклад]=@ocl,[Дата\_приема\_на\_работу]=@time where [Номер\_менеджера]=@num;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@num", num);

cmd1.Parameters.AddWithValue("@login", login);

cmd1.Parameters.AddWithValue("@name", name);

cmd1.Parameters.AddWithValue("@ocl", ocl);

cmd1.Parameters.AddWithValue("@time", time);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры". При необходимости она может быть перемещена или удалена.

this.менеджерыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры);

textBox19.Clear();

textBox18.Clear();

textBox17.Clear();

textBox16.Clear();

textBox29.Clear();

}

private void button13\_Click(object sender, EventArgs e) //удалить

{

string num = Convert.ToString(textBox19.Text);

string connectionString = @" Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Delete from Менеджеры where [Номер\_менеджера]=@num";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@num", num);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры". При необходимости она может быть перемещена или удалена.

this.менеджерыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet4.Менеджеры);

textBox19.Clear();

textBox18.Clear();

textBox17.Clear();

textBox16.Clear();

textBox29.Clear();

}

private void button12\_Click(object sender, EventArgs e) //поиск

{

dataGridView2.ClearSelection();

string searchValue = textBox12.Text;

dataGridView2.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView2.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView2.Refresh();

textBox12.Clear();

}

////////////////////////// ТАБЛИЦА 5

private void button25\_Click(object sender, EventArgs e) //добавить

{

string log = Convert.ToString(textBox26.Text);

string tel = Convert.ToString(textBox25.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Клиенты([Логин\_клиента],[Телефон]) VALUES(@log,@tel);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@log", log);

cmd1.Parameters.AddWithValue("@tel", tel);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты);

textBox27.Clear();

textBox26.Clear();

textBox25.Clear();

}

private void button24\_Click(object sender, EventArgs e) //изменить

{

string num = Convert.ToString(textBox27.Text);

string log = Convert.ToString(textBox26.Text);

string tel = Convert.ToString(textBox25.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Клиенты set [Логин\_клиента]=@log,[Телефон]=@tel where [Номер\_клиента]=@num;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@num", num);

cmd1.Parameters.AddWithValue("@log", log);

cmd1.Parameters.AddWithValue("@tel", tel);

MyConnection.Open();

cmd1.ExecuteNonQuery();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты);

textBox27.Clear();

textBox26.Clear();

textBox25.Clear();

}

private void button23\_Click(object sender, EventArgs e) //удалить

{

string num = Convert.ToString(textBox27.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Delete from Клиенты where [Номер\_клиента]=@num";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@num", num);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты". При необходимости она может быть перемещена или удалена.

this.клиентыTableAdapter.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet5.Клиенты);

textBox27.Clear();

textBox26.Clear();

textBox25.Clear();

}

private void button22\_Click(object sender, EventArgs e) //поиск

{

dataGridView5.ClearSelection();

string searchValue = textBox20.Text;

dataGridView5.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView5.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView5.Refresh();

textBox20.Clear();

}

private void button21\_Click(object sender, EventArgs e) //сортировка

{

if (radioButton10.Checked)

{

if (comboBox5.Text == "Номер")

{

dataGridView5.Sort(dataGridView5.Columns[0], ListSortDirection.Descending);

}

else if (comboBox5.Text == "Логин")

{

dataGridView5.Sort(dataGridView5.Columns[1], ListSortDirection.Descending);

}

else if (comboBox5.Text == "Телефон")

{

dataGridView5.Sort(dataGridView5.Columns[2], ListSortDirection.Descending);

}

}

else if (radioButton9.Checked)

{

if (comboBox5.Text == "Номер")

{

dataGridView5.Sort(dataGridView5.Columns[0], ListSortDirection.Ascending);

}

else if (comboBox5.Text == "Логин")

{

dataGridView5.Sort(dataGridView5.Columns[1], ListSortDirection.Ascending);

}

else if (comboBox5.Text == "Телефон")

{

dataGridView5.Sort(dataGridView5.Columns[2], ListSortDirection.Ascending);

}

}

}

////////////////////////// ТАБЛИЦА 6

private void button16\_Click(object sender, EventArgs e) //добавить

{

string login = Convert.ToString(textBox33.Text);

string parol = Convert.ToString(textBox32.Text);

string rol = Convert.ToString(textBox30.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Insert into Пользователи([Логин],[Пароль],[Роль]) VALUES(@login,@parol,@rol);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@login", login);

cmd1.Parameters.AddWithValue("@parol", parol);

cmd1.Parameters.AddWithValue("@rol", rol);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter2.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи);

textBox33.Clear();

textBox32.Clear();

textBox30.Clear();

}

private void button11\_Click(object sender, EventArgs e) //изменить

{

string login = Convert.ToString(textBox33.Text);

string parol = Convert.ToString(textBox32.Text);

string rol = Convert.ToString(textBox30.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = $"Update Пользователи set [Пароль]=@parol,[Роль]=@rol where [Логин]=@login;";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@login", login);

cmd1.Parameters.AddWithValue("@parol", parol);

cmd1.Parameters.AddWithValue("@rol", rol);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter2.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи);

textBox33.Clear();

textBox32.Clear();

textBox30.Clear();

}

private void button10\_Click(object sender, EventArgs e) //удалить

{

string log = Convert.ToString(textBox33.Text);

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Delete from Пользователи where [Логин]=@log";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@log", log);

MyConnection.Open();

cmd1.ExecuteNonQuery();

MyConnection.Close();

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter2.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet7.Пользователи);

textBox33.Clear();

textBox32.Clear();

textBox30.Clear();

}

private void button26\_Click(object sender, EventArgs e) //поиск

{

dataGridView6.ClearSelection();

string searchValue = textBox34.Text;

dataGridView6.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView6.Rows)

{

if (row.Cells[0].Value.ToString().TrimEnd() == searchValue)

{

row.Selected = true;

break;

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

dataGridView6.Refresh();

textBox34.Clear();

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

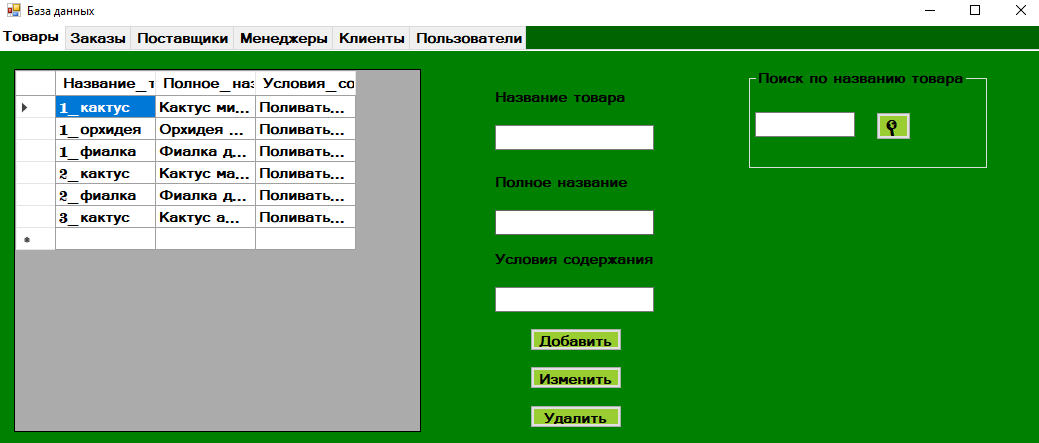


Рисунок 18 – Вкладка Товары

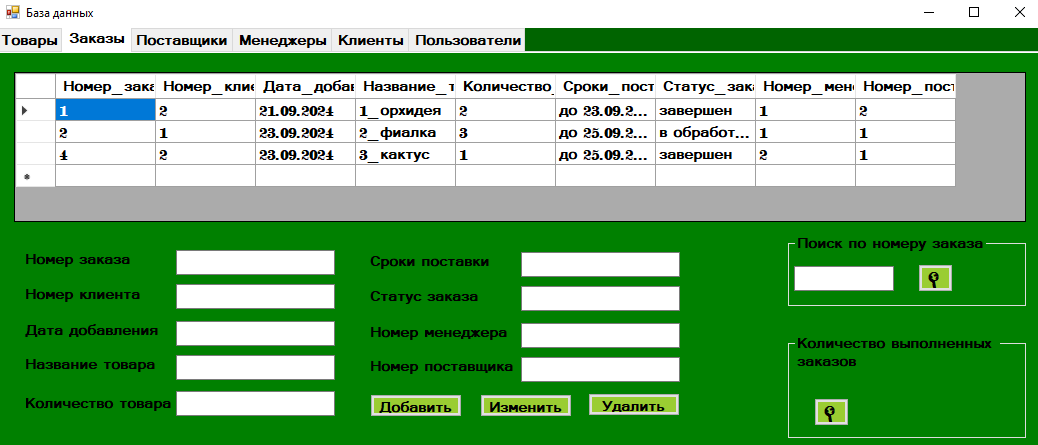


Рисунок 19 – Вкладка Заказы

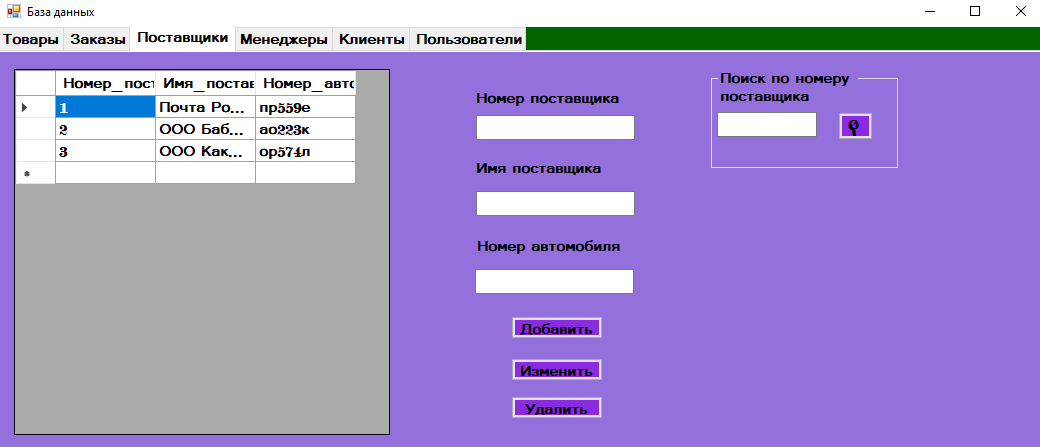


Рисунок 20 – Вкладка Поставщики

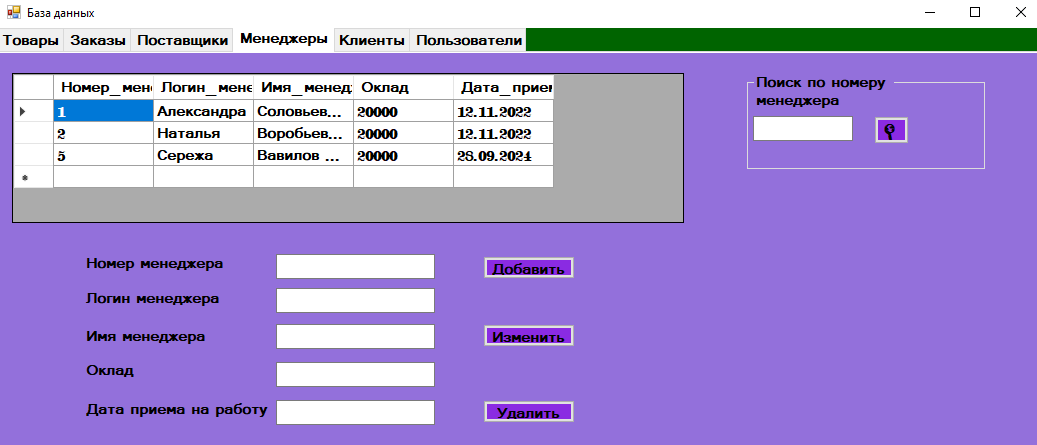


Рисунок 21 – Вкладка Менеджеры

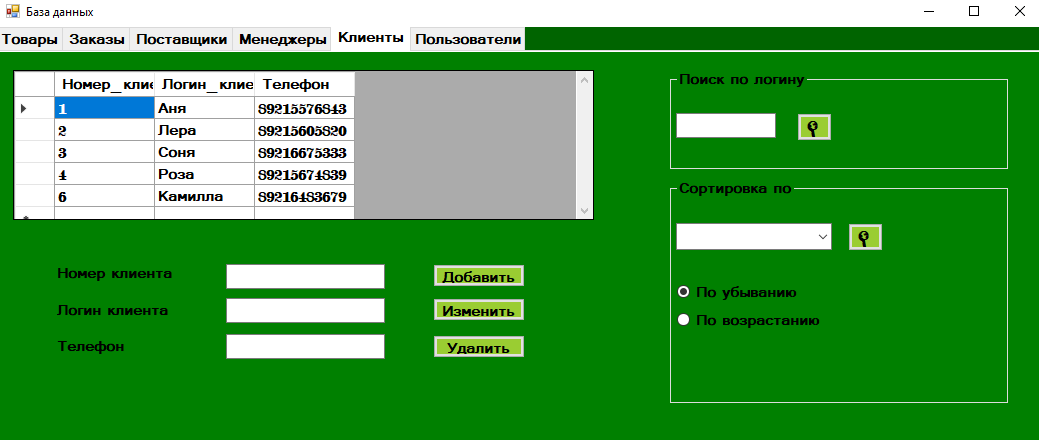


Рисунок 22 – Вкладка Клиенты

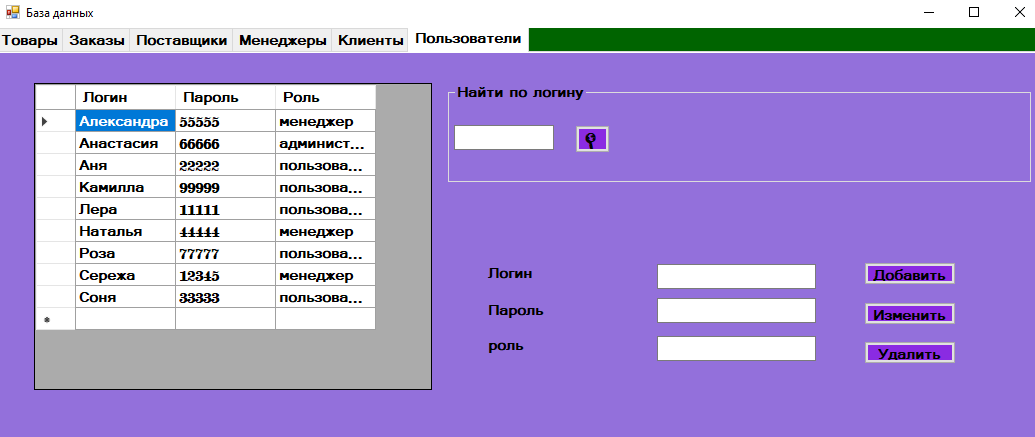


Рисунок 23 – Вкладка Пользователи

Код Form2:

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;

namespace \_17

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public int Admin;

public int ADMIN

{

get { return Admin; }

set { value = Admin; }

}

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

{

button3.Visible = false;

textBox3.Visible = false;

button3.Visible = false;

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter4.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи);

try

{

string log = textBox1.Text;

string pass =textBox2.Text;

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDel = "Select Логин from Пользователи where Логин='" + log + "'";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

string ComDell = "Select Пароль from Пользователи where Пароль=" + pass+ " and Логин='" + log + "'";

SqlCommand cmd2 = new SqlCommand(ComDell, MyConnection);

string ComDelll = "Select Роль from Пользователи where Пароль=" + pass + " and Логин='" + log + "'";

SqlCommand cmd3 = new SqlCommand(ComDelll, MyConnection);

MyConnection.Open();

string a = cmd1.ExecuteScalar().ToString().TrimEnd();

string b = cmd2.ExecuteScalar().ToString().TrimEnd();

string c = cmd3.ExecuteScalar().ToString().TrimEnd();

if (a == log && b == pass)

{

Form1 form1 = new Form1(); // создание экземпляра первой формы

if (c == "администратор")

Admin = 0;

else if (c == "менеджер")

Admin = 1;

else Admin = 2;

form1.Owner = this; // родительской для формы 2 будет текущая форма

form1.ShowDialog(); // показать окно второй формы в немодальном режиме

this.Visible = false;

this.Enabled = false;

}

else { MessageBox.Show("Неправильный пароль!"); }

MyConnection.Close();

}

catch (Exception ex)

{

MessageBox.Show("Такого пользователя не существует. Пройдите регистрацию!");

}

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter4.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи);

}

private void button2\_Click(object sender, EventArgs e) //кнопка регистрации

{

button3.Visible = true;

button1.Visible = false;

textBox3.Visible = true;

button3.Visible = true;

label3.Visible = true;

}

private void button3\_Click(object sender, EventArgs e) //кнопка добавить

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter4.Fill(this.\_\_\_\_Тихонова\_4\_курс\_практикаDataSet.Пользователи);

try

{

string log = textBox1.Text;

string pass = textBox2.Text;

string tel = textBox3.Text;

string role = "пользователь";

string connectionString = @"Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(connectionString);

string ComDell = "SELECT COUNT(\*) FROM [Пользователи] where [Логин]=@log";

SqlCommand cmd2 = new SqlCommand(ComDell, MyConnection);

cmd2.Parameters.AddWithValue("@log", log);

MyConnection.Open();

int result = 0;

result = ((int)cmd2.ExecuteScalar());

//String b = cmd2.ExecuteScalar().ToString();

if (result == 0 && log != "")

{

string ComDel = $"Insert into Пользователи([Логин],[Пароль],[Роль]) VALUES(@log,@pass,@role);";

SqlCommand cmd1 = new SqlCommand(ComDel, MyConnection);

cmd1.Parameters.AddWithValue("@log", log);

cmd1.Parameters.AddWithValue("@pass", pass);

cmd1.Parameters.AddWithValue("@role", role);

cmd1.ExecuteNonQuery();

string ComDel2 = $"Insert into Клиенты([Логин\_клиента],[Телефон]) VALUES(@log,@tel);";

SqlCommand cmd3 = new SqlCommand(ComDel2, MyConnection);

cmd3.Parameters.AddWithValue("@log", log);

cmd3.Parameters.AddWithValue("@tel", tel);

cmd3.ExecuteNonQuery();

}

else

{

MessageBox.Show("Такой логин уже занят, выберите другое имя пользователя");

}

MyConnection.Close();

}

catch

{

MessageBox.Show("Введите данные в поля");

}

label3.Visible = false;

textBox3.Visible = false;

button3.Visible = false;

label3.Visible = false;

button1.Visible = true;

textBox1.Clear();

textBox2.Clear();

}

private void button4\_MouseUp(object sender, MouseEventArgs e)

{

button4.ForeColor = Color.Black;

textBox2.PasswordChar = '\*';

}

private void button4\_MouseDown(object sender, MouseEventArgs e)

{

button4.ForeColor = Color.Red;

textBox2.PasswordChar = '\0';

}

}

}

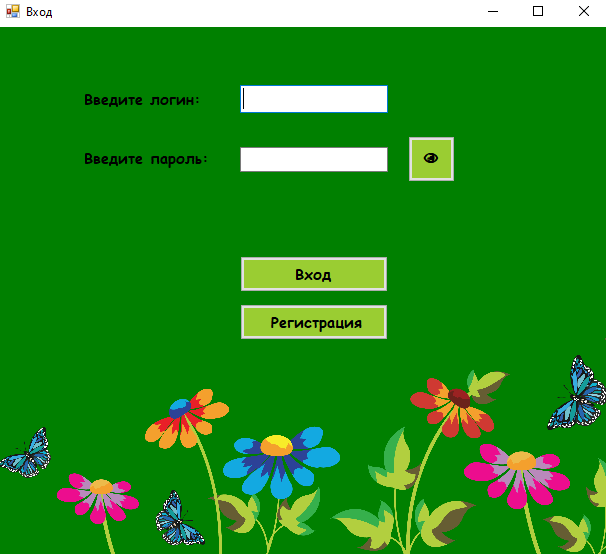


Рисунок 24 – Форма входа

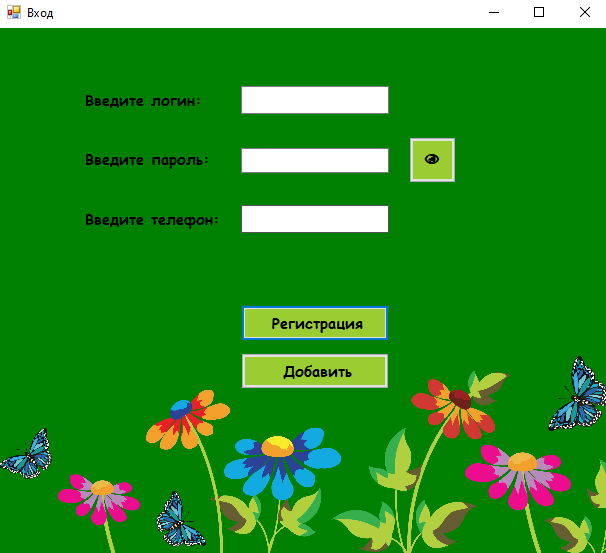


Рисунок 25 – Форма регистрации

Работа с отладчиком:

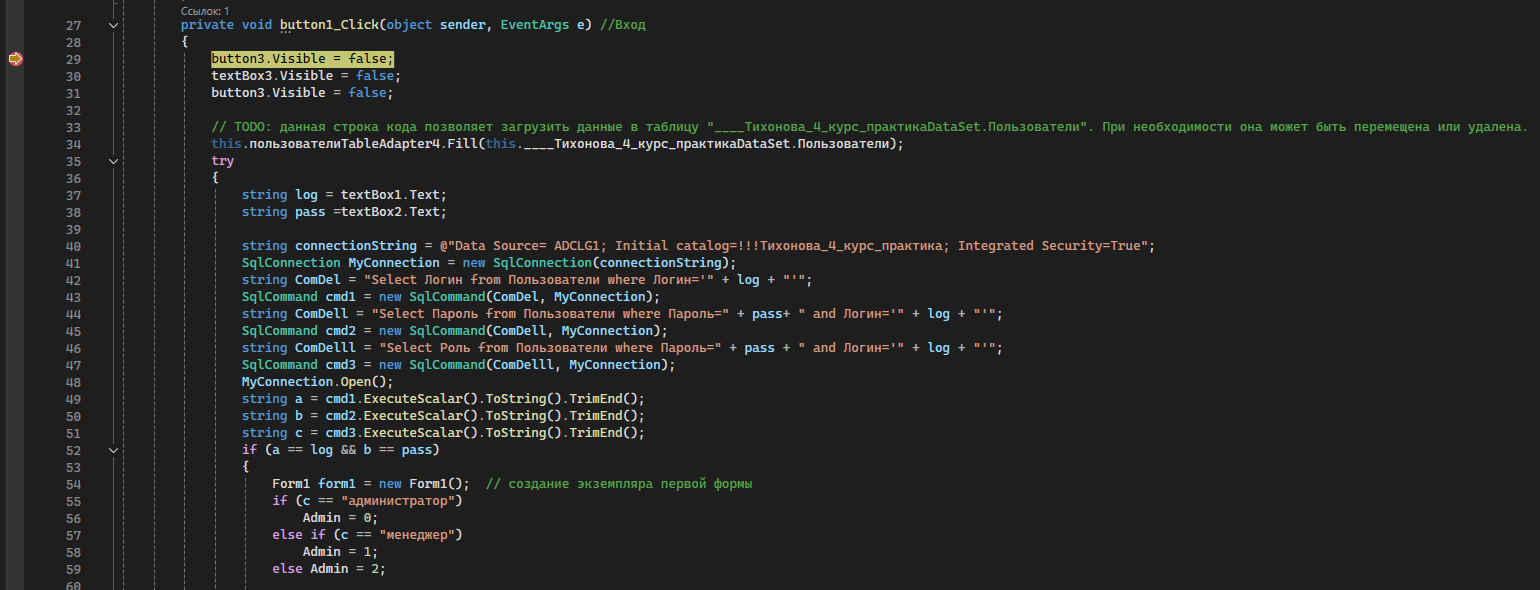


Рисунок 25 – Работа с отладчиком

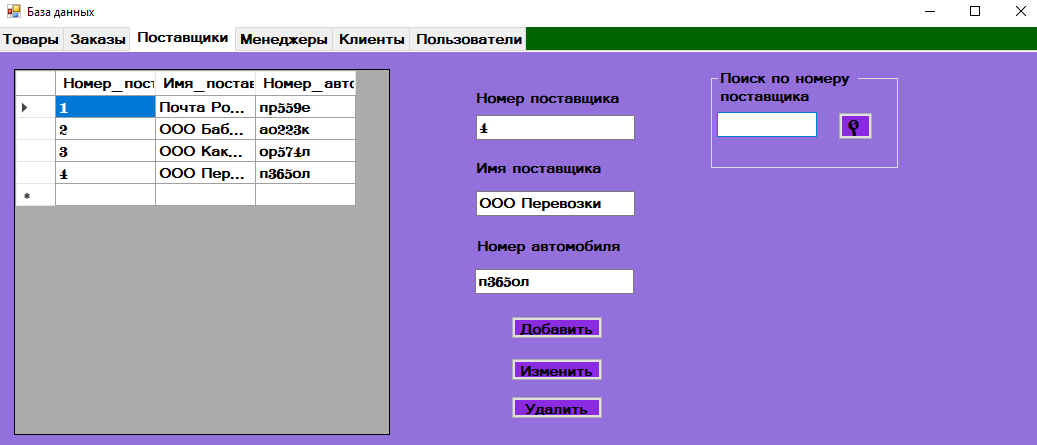


Рисунок 26 – Функция добавления

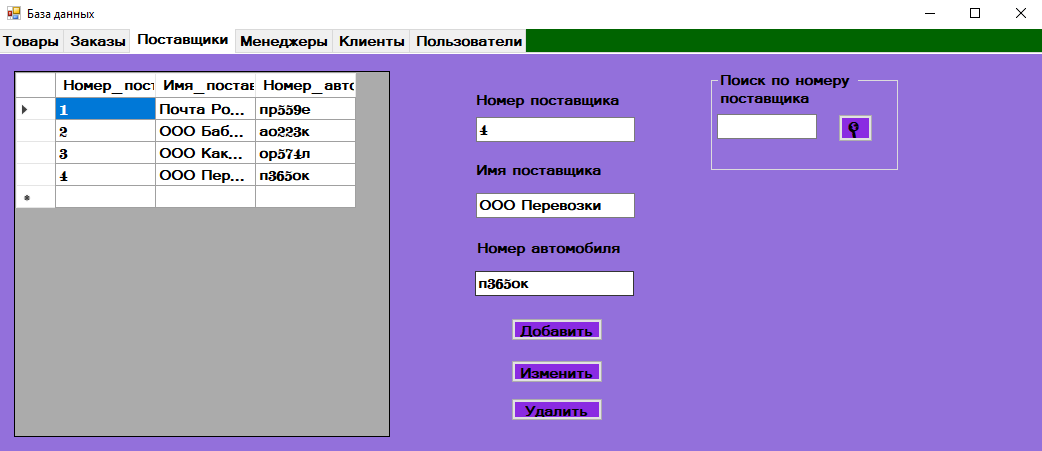


Рисунок 27 – Функция изменения

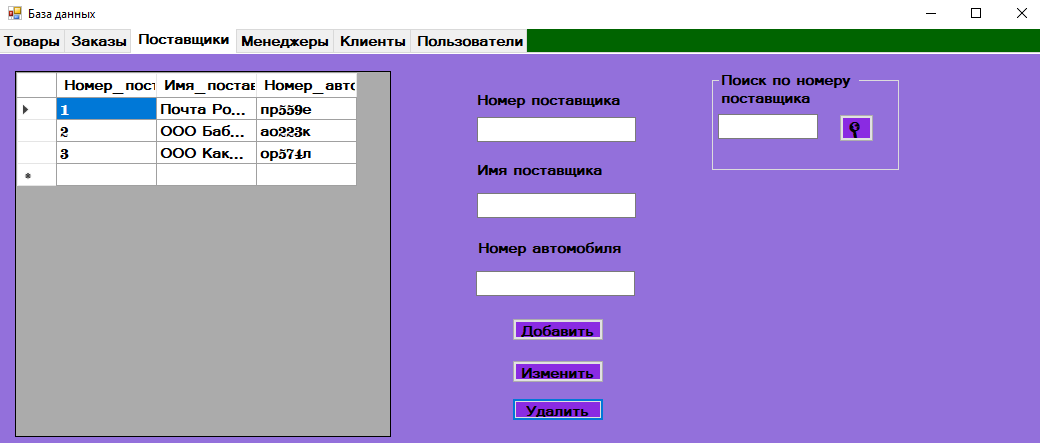


Рисунок 28 – Функция удаления

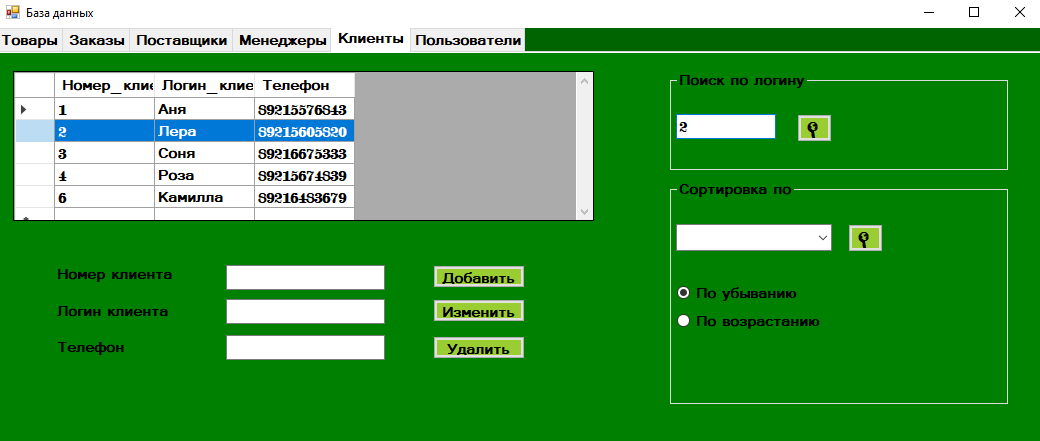


Рисунок 29 – Функция поиска

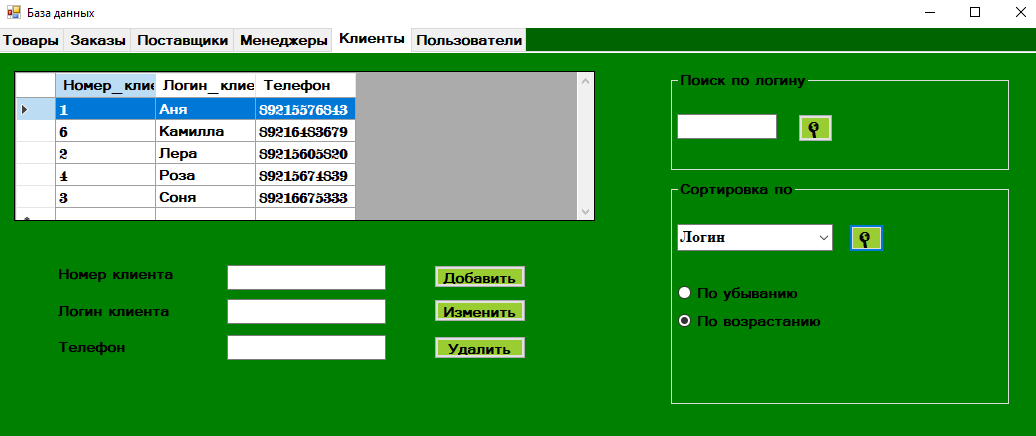


Рисунок 30 – Функция сортировки

Задание 5. Тестирование

Тест-кейсы

Тестовые сценарии:

1. Test-сase 1

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Приложение для работы с БД “Поставка товаров” |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Валерия |
| **Дата(ы) теста** | 02.10.2024 |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Подтверждение страницы входа с действительным именем пользователя и, паролем. |
| **Краткое изложение теста** | Тест проверяет функцию входа в программу под учетной записью пользователя. |
| **Этапы теста** | * Запустить программу * Вписать логин * Вписать пароль * Нажать кнопку войти * Проверить количество открытых вкладок |
| **Тестовые данные** | Логин – Аня  Пароль – 22222 |
| **Ожидаемый результат** | Ошибок не должно возникнуть, программа продолжает работать правильно. |
| **Фактический результат** | В результате теста программа должна авторизовать пользователя и позволить ему видеть две вкладки программы. |
| **Предварительное условие** | Логин и пароль должны существовать в базе данных. |
| **Постусловие** | Два поля из шести должны быть доступны для просмотра. |
| **Статус** *(Зачет/Незачет)* | Зачет |

1. Test-сase 2

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Приложение для работы с БД “Поставка товаров” |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Валерия |
| **Дата(ы) теста** | 02.10.2024 |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Подтверждение страницы входа с действительным именем пользователя и, паролем под учетной записью администратора. |
| **Краткое изложение теста** | Тест проверяет функцию входа в программу под учетной записью администратора. |
| **Этапы теста** | * Запустить программу * Вписать логин * Вписать пароль * Нажать кнопку войти * Проверить количество открытых вкладок |
| **Тестовые данные** | Логин – Анастасия  Пароль – 66666 |
| **Ожидаемый результат** | Ошибок не должно возникнуть, программа продолжает работать правильно. |
| **Фактический результат** | В результате теста программа должна авторизовать пользователя и позволить ему видеть все вкладки программы. |
| **Предварительное условие** | Логин и пароль должны существовать в базе данных. |
| **Постусловие** | Все шесть полей из шести должны быть доступны для просмотра. |
| **Статус** *(Зачет/Незачет)* | Зачет |

1. Test-сase 3

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Приложение для работы с БД “Поставка товаров” |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Валерия |
| **Дата(ы) теста** | 02.10.2024 |
| **Тестовый пример #** | TC\_UI\_3 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Подтверждение страницы создания профиля с действительным именем пользователя, паролем и телефоном. |
| **Краткое изложение теста** | Тест проверяет функцию создания профиля. |
| **Этапы теста** | * Запустить программу * Нажать кнопку “регистрация” * Вписать логин * Вписать пароль * Вписать телефон * Нажать кнопку “добавить” |
| **Тестовые данные** | Логин – Виктор  Пароль – 12345  Телефон – 89326473628 |
| **Ожидаемый результат** | Ошибок не должно возникнуть, программа продолжает работать правильно. |
| **Фактический результат** | В результате теста в БД в таблице “пользователи” должен появиться введенный пользователь. |
| **Предварительное условие** | Логин не должен существовать в базе данных до проведения теста. |
| **Постусловие** | БД в таблице “пользователи” должен появиться введенный пользователь. |
| **Статус** *(Зачет/Незачет)* | Зачет |

1. Test-сase 4

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Приложение для работы с БД “Поставка товаров” |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Валерия |
| **Дата(ы) теста** | 02.10.2024 |
| **Тестовый пример #** | TC\_UI\_4 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Проверка функции добавления информации в таблицу. |
| **Краткое изложение теста** | Тест проверяет функцию добавления информации в таблицу. |
| **Этапы теста** | * Запустить программу * Войти под учетной записью администратора * Зайти на страницу “товары” * Заполнить поле “Название товара” * Заполнить поле “Полное название” * Заполнить поле “Условия содержания” * Нажать кнопку “Добавить” |
| **Тестовые данные** | Логин – Анастасия  Пароль – 66666  Название товара – 1\_пальма  Полное название – Пальма африканская  Условия содержания – Поливать редко |
| **Ожидаемый результат** | Ошибок не должно возникнуть, программа продолжает работать правильно. |
| **Фактический результат** | В результате теста в таблице “Товары” должна появиться новая запись с введенными данными. |
| **Предварительное условие** | Логин и пароль должны существовать в БД. Запись о товаре не должна существовать в БД. |
| **Постусловие** | Данные в появившейся строке должны совпадать с введенными пользователем. |
| **Статус** *(Зачет/Незачет)* | Зачет |

1. Test-сase 5

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Приложение для работы с БД “Поставка товаров” |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Валерия |
| **Дата(ы) теста** | 02.10.2024 |
| **Тестовый пример #** | TC\_UI\_5 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Проверка функции поиска. |
| **Краткое изложение теста** | Тест проверяет функцию поиска информации в таблице. |
| **Этапы теста** | * Запустить программу * Войти под учетной записью администратора * Зайти на страницу “товары” * Заполнить поле “Найти по названию товара” * Нажать кнопку с картинкой лупы |
| **Тестовые данные** | Логин – Анастасия  Пароль – 66666  Название товара – 1\_кактус |
| **Ожидаемый результат** | Ошибок не должно возникнуть, программа продолжает работать правильно. |
| **Фактический результат** | В результате теста в таблице “Товары” должна выделиться синим цветом запись с названием “1\_кактус”. |
| **Предварительное условие** | Логин и пароль должны существовать в БД. Запись о товаре должна существовать в БД. |
| **Постусловие** | Данные о названии в появившейся строке должны совпадать с введенным пользователем названием. |
| **Статус** *(Зачет/Незачет)* | Зачет |

Модульные тесты:

Код:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

string connectionString = "Data Source= ADCLG1; Initial catalog=!!!Тихонова\_4\_курс\_практика; Integrated Security=True";

[TestMethod]

public void Test\_Users\_Count()

{

// Подготовка теста: создание запроса для получения всех логинов

string querystring = $"select Пользователи.Логин from Пользователи";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

// Проверка результата: ожидается, что таблица будет содержать 9 строк

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(9, table.Rows.Count);

}

[TestMethod]

public void Test\_Tovar\_Count()

{

// Подготовка теста: создание запроса для получения всех логинов

string querystring = $"select Название\_товара from Товары";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

// Проверка результата: ожидается, что таблица будет содержать 6 строк

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(6, table.Rows.Count);

}

[TestMethod]

public void Test\_Zakaz\_Count()

{

// Подготовка теста: создание запроса для получения всех логинов

string querystring = $"select Номер\_заказа from Заказы";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

// Проверка результата: ожидается, что таблица будет содержать 3 строки

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(3, table.Rows.Count);

}

[TestMethod]

public void GetQuantityForProduct\_NoLogin()

{

// Подготовка теста: создание запроса с пустым логино

string querystring = $"select Логин, Пароль, Роль from Пользователи where Логин =''";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

// Проверка результата: ожидается, что таблица будет пустой (0 строк)

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(0, table.Rows.Count);

}

[TestMethod]

public void GetQuantityForProduct\_CorrectLogin()

{

// Подготовка теста: создание запроса с корректными логином и паролем

string querystring = $"select Логин, Пароль, Роль from Пользователи where Логин ='Анастасия' and Пароль= '66666' ";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

// Проверка результата: ожидается, что таблица будет содержать одну строку (1 пользователь)

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(1, table.Rows.Count);

}

[TestMethod]

public void LookFor\_Zakaz()

{

string querystring = $"select Название\_товара from Заказы where Номер\_заказа = 1";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual("1\_орхидея", table.Rows[0]["Название\_товара"].ToString());

}

[TestMethod]

public void LookFor\_Client()

{

string querystring = $"select Телефон from Клиенты where Номер\_клиента = 1";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual("89215576843", table.Rows[0]["телефон"].ToString());

}

[TestMethod]

public void LookFor\_Meneger()

{

string querystring = $"select Дата\_приема\_на\_работу from Менеджеры where Номер\_менеджера = 1";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual("12.11.2022", table.Rows[0]["Дата\_приема\_на\_работу"].ToString());

}

[TestMethod]

public void Check\_Meneger\_not\_exists()

{

// Подготовка теста: создание запроса для получения всех логинов менеджеров

string querystring = $"select Логин from Пользователи where Роль = 'менеджер'";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreNotEqual("Александр", table.Rows[0]["Логин"].ToString());

}

[TestMethod]

public void Check\_Meneger\_exists()

{

// Подготовка теста: создание запроса для получения всех логинов менеджеров

string querystring = $"select Логин from Пользователи where Роль = 'менеджер'";

SqlDataAdapter adapter = new SqlDataAdapter(querystring, connectionString);

// Выполнение теста: заполнение DataTable данными из базы данных

DataTable table = new DataTable();

adapter.Fill(table);

Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual("Александра", table.Rows[0]["Логин"].ToString());

}

}

}

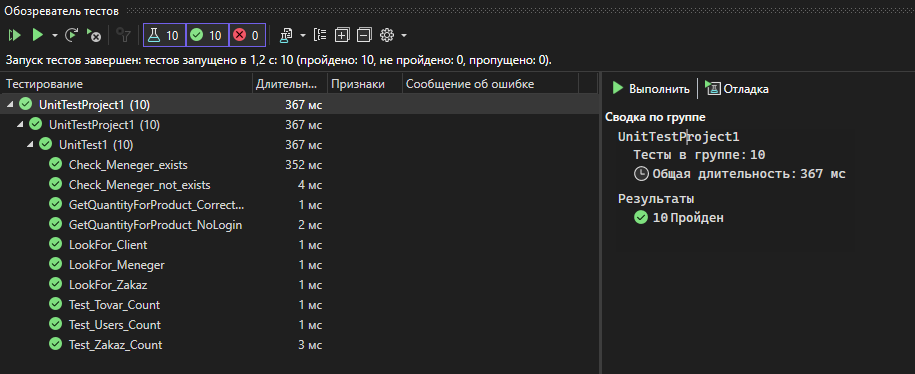


Рисунок 31 – Результат выполнения модульных тестов

Задание 6. GitHub. Оценка проекта

В репозиторий GIT выгрузите получившийся проект.

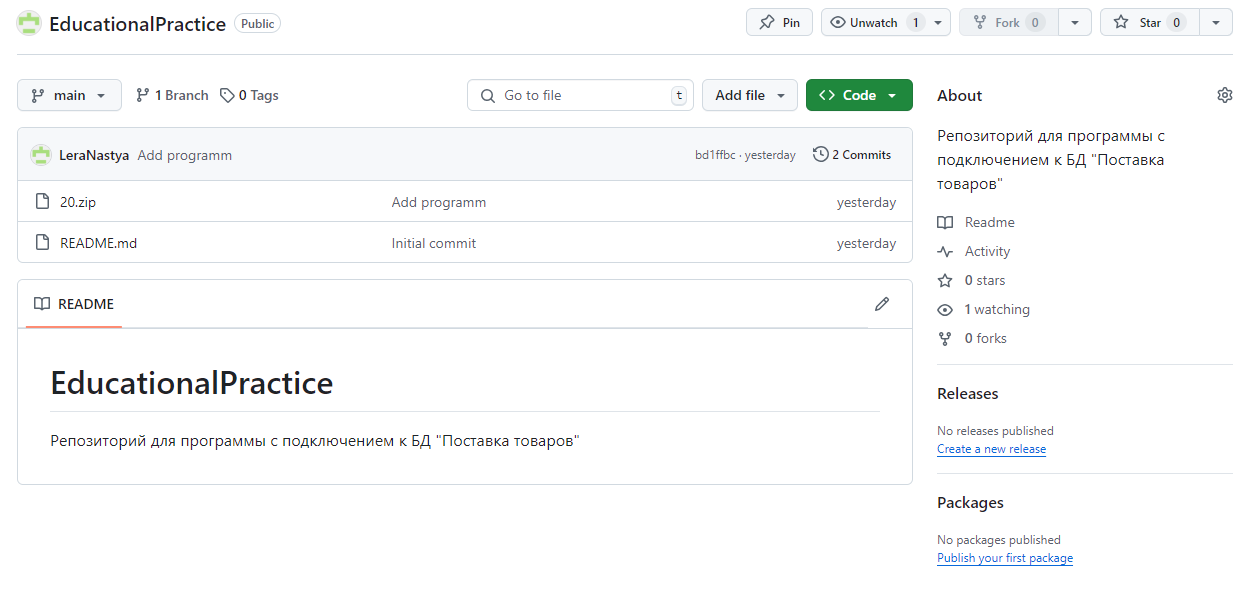


Рисунок 32 – выгрузка на git

ЗАКЛЮЧЕНИЕ

В ходе учебной практики я успешно разработала программу, интегрировала ее с базой данных, провела тестирование и создали диаграммы и макеты. Этот опыт позволил мне не только закрепить теоретические знания, но и приобрести ценные практические навыки, которые будут полезны в дальнейшей профессиональной деятельности.

Работа над проектом показала важность каждого этапа разработки программного обеспечения, от планирования и проектирования до тестирования и документирования.

В заключение хотелось бы отметить, что учебная практика стала важным шагом на пути к профессиональному росту. Полученные знания и навыки будут полезны в дальнейшем обучении и работе, а также помогут мне более эффективно решать задачи в области информационных технологий.

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ УП.02.01 (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/2 группы

Тихонова Валерия Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «23» сентября 2024 г. по «05» октября 2024 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Виды выполненных работ обучающимся**  **во время практики** | **Объем работ, час.** | **Качество выполнения работ (оценка по пятибалльной системе)** |
| Участие в выработке требований к программному обеспечению | 9 |  |
| Стадии проектирования программного обеспечения | 21 |  |
| Разработка модулей программного обеспечения | 24 |  |
| Тестирование программных модулей и их интеграции | 9 |  |
| Разработка программной документации и стандарты кодирования | 9 |  |

**Характеристика учебной/профессиональной деятельности обучающегося во время учебной практики (по профилю специальности):**

Общие и профессиональные компетенции, предусмотренные программой практики, освоены **/** не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Хисамутдинова А.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)

Дата «05» октября 2024 г.