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;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace bd

{

public partial class Form1 : Form

{

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\Привет\\Desktop\\разраб\\bd\\bd\\Database1.mdf;Integrated Security=True";

public Form1()

{

InitializeComponent();

}

private void LoadData()

{

// Замените эти параметры на соответствующие значения вашей базы данных

string query = "SELECT \* FROM [Client]";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

try

{

connection.Open();

dataAdapter.Fill(dataTable);

dataGridView1.DataSource = dataTable; // Здесь dataGridView1 — имя вашего элемента DataGridView

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

}

private void button1\_click(object sender, EventArgs e)

{

// Получаем данные из TextBox

string clientFio = textBox2.Text;

string phoneNumber = textBox3.Text;

string address = textBox4.Text;

string email = textBox5.Text;

string orderNumber = textBox6.Text;

// Строка подключения к базе данных (замените на вашу)

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = "INSERT INTO client (client\_fio, phone\_number, adress, email, order\_number) VALUES (@Fio, @PhoneNumber, @Address, @Email, @OrderNumber)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@Fio", clientFio);

command.Parameters.AddWithValue("@PhoneNumber", phoneNumber);

command.Parameters.AddWithValue("@Address", address);

command.Parameters.AddWithValue("@Email", email);

command.Parameters.AddWithValue("@OrderNumber", orderNumber);

command.ExecuteNonQuery();

}

}

// Очистка текстовых полей после добавления

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

LoadData();

}

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

{

LoadData();

}

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

{

Order orderForm = new Order();

orderForm.ShowDialog();

}

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

{

Material materialForm = new Material();

materialForm.ShowDialog();

}

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

{

Product productForm = new Product();

productForm.ShowDialog();

}

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

{

Supplier supplierForm = new Supplier();

supplierForm.ShowDialog();

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_click(object sender, EventArgs e)

{

string clientFio = textBox2.Text;

string phoneNumber = textBox3.Text;

string address = textBox4.Text;

string email = textBox5.Text;

string orderNumber = textBox6.Text;

// SQL запрос для обновления данных

string query = "Update client SET phone\_number = @phone\_number, adress = @adress, email = @email, order\_number = @order\_number WHERE client\_fio = @client\_fio";

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@client\_Fio", clientFio);

command.Parameters.AddWithValue("@phone\_number", phoneNumber);

command.Parameters.AddWithValue("@adress", address);

command.Parameters.AddWithValue("@email", email);

command.Parameters.AddWithValue("@order\_number", orderNumber);

try

{

connection.Open();

command.ExecuteNonQuery(); // Выполняем команду

MessageBox.Show("Данные успешно обновлены!");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при обновлении данных: " + ex.Message);

}

}

}

// Перезагружаем данные в DataGridView

LoadData();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

// Получаем выбранную строку

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

// Пример: заполняем текстовые поля с данными клиента

textBox2.Text = row.Cells["client\_fio"].Value.ToString(); // Убедитесь, что вы используете правильные имена столбцов

textBox3.Text = row.Cells["phone\_number"].Value.ToString();

textBox4.Text = row.Cells["adress"].Value.ToString();

textBox5.Text = row.Cells["email"].Value.ToString();

textBox6.Text = row.Cells["order\_number"].Value.ToString();

}

}

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

{

if (string.IsNullOrEmpty(textBox2.Text))

{

MessageBox.Show("Выберите клиента для удаления.");

return;

}

string clientFio = textBox2.Text;

// SQL запрос для удаления данных

string query = "DELETE FROM client WHERE client\_fio = @client\_fio";

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@client\_fio", clientFio);

try

{

connection.Open();

int rowsAffected = command.ExecuteNonQuery(); // Выполняем команду удаления

if (rowsAffected > 0)

{

MessageBox.Show("Клиент успешно удалён.");

}

else

{

MessageBox.Show("Клиент не найден.");

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при удалении клиента: " + ex.Message);

}

}

}

// Перезагружаем данные в DataGridView

LoadData();

}

}

}