CREATE DATABASE HotelManagement;

CREATE TABLE Rooms (

Number INT PRIMARY KEY,

Rooms INT,

RoomPhoto IMAGE,

Storey INT,

TV BIT,

Fridge BIT,

Bed INT,

Balcony BIT

);

INSERT INTO Rooms (Number, Rooms, Storey, TV, Fridge, Bed, Balcony, RoomPhoto)

VALUES

(101, 2, 1, 1, 1, 1, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\101.png', Single\_Blob) as RoomImage)),

(102, 1, 2, 0, 1, 2, 1, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\102.png', Single\_Blob) as RoomImage)),

(103, 3, 1, 1, 0, 2, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\103.png', Single\_Blob) as RoomImage)),

(104, 2, 2, 1, 1, 1, 1, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\104.png', Single\_Blob) as RoomImage)),

(105, 1, 1, 0, 0, 2, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\105.png', Single\_Blob) as RoomImage)),

(106, 3, 3, 1, 1, 1, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\106.png', Single\_Blob) as RoomImage)),

(107, 2, 2, 0, 0, 2, 1, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\107.png', Single\_Blob) as RoomImage)),

(108, 1, 1, 1, 1, 2, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\108.png', Single\_Blob) as RoomImage)),

(109, 3, 2, 0, 0, 1, 1, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\109.png', Single\_Blob) as RoomImage)),

(110, 2, 3, 1, 1, 2, 0, (SELECT BulkColumn FROM Openrowset( Bulk 'C:\Users\Tays\Desktop\Практика\Деятельность гостинцы\Data\110.png', Single\_Blob) as RoomImage));

CREATE TABLE Admins (

AdminID INT IDENTITY(1,1) PRIMARY KEY,

Username VARCHAR(20),

Password VARCHAR(20)

);

INSERT INTO Admins VALUES

(1, 'admin', 'admin123');

CREATE TABLE Managers (

ManagerID INT IDENTITY(1,1) PRIMARY KEY,

Username VARCHAR(20),

Password VARCHAR(20)

);

INSERT INTO Managers VALUES

(1, 'manager', 'manager123');

CREATE TABLE Guests (

GuestID INT IDENTITY(1,1) PRIMARY KEY,

LastName VARCHAR(40),

FirstName VARCHAR(40),

MiddleName VARCHAR(40),

Address VARCHAR(60),

Town VARCHAR(20),

Aim VARCHAR(30),

Passport VARCHAR(20),

PassportDate DATE,

Region VARCHAR(40),

Work VARCHAR(20),

BirthYear DATE,

Picture IMAGE,

PhoneNumber VARCHAR(20)

);

CREATE TABLE Reservations (

ReservationID INT IDENTITY(1,1) PRIMARY KEY,

GuestID INT,

RoomNumber INT,

LastName VARCHAR(40),

FirstName VARCHAR(40),

MiddleName VARCHAR(40),

Come DATE,

Leave DATE,

Money MONEY,

Cash BIT,

Receipt INT,

Comment VARCHAR(MAX),

Registrar VARCHAR(15),

FOREIGN KEY (RoomNumber) REFERENCES Rooms(Number),

FOREIGN KEY (GuestID) REFERENCES Guests(GuestID)

);

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;

using System.IO;

using static Деятельность\_гостинцы.GuestForm;

using System.Data.SqlClient;

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

using System.Globalization;

using QRCoder;

using System.Runtime.Remoting.Messaging;

namespace Деятельность\_гостинцы

{

public partial class ManagerForm : Form

{

private static GuestForm guestFormInstance;

private byte[] imageData;

private int currentReceiptNumber = 1;

public ManagerForm(GuestForm guestForm)

{

InitializeComponent();

guestFormInstance = guestForm;

textBox14.Text = currentReceiptNumber.ToString("D4");

maskedTextBox1.Mask = "9999-00-00";

maskedTextBox2.Mask = "9999-00-00";

maskedTextBox3.Mask = "9999-00-00";

maskedTextBox4.Mask = "9999-00-00";

}

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

{

timer1.Interval = 1000;

timer1.Tick += timer1\_Tick;

timer1.Start();

GuestForm.ReservationRequested += GuestForm\_ReservationRequested;

}

private void GuestForm\_ReservationRequested(object sender, GuestForm.ReservationEventArgs e)

{

HandleReservationData(e.RoomNumber, e.PhoneNumber);

}

private void HandleReservationData(int roomNumber, string phoneNumber)

{

textBox1.Text = roomNumber.ToString();

textBox2.Text = phoneNumber;

}

private void timer1\_Tick(object sender, EventArgs e)

{

string filePath = "ReservationData.txt";

if (File.Exists(filePath))

{

string[] lines = File.ReadAllLines(filePath);

if (lines.Length > 0)

{

string[] data = lines[0].Split(',');

int roomNumber = int.Parse(data[0]);

string phoneNumber = data[1];

HandleReservationData(roomNumber, phoneNumber);

File.Delete(filePath);

}

}

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Image Files |\*.jpg;\*.jpeg;\*.png;\*.gif;\*.bmp";

pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

string imagePath = openFileDialog.FileName;

pictureBox1.Image = Image.FromFile(imagePath);

imageData = File.ReadAllBytes(imagePath);

}

}

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

{

if (ValidateInputs())

{

SaveGuestToDatabase();

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

}

else

{

MessageBox.Show("Пожалуйста, заполните все обязательные поля.");

}

}

private bool ValidateInputs()

{

return !string.IsNullOrEmpty(textBox3.Text)

&& !string.IsNullOrEmpty(textBox2.Text)

&& !string.IsNullOrEmpty(textBox4.Text)

&& !string.IsNullOrEmpty(textBox5.Text)

&& !string.IsNullOrEmpty(maskedTextBox2.Text)

&& !string.IsNullOrEmpty(textBox7.Text)

&& !string.IsNullOrEmpty(textBox8.Text)

&& !string.IsNullOrEmpty(textBox9.Text)

&& !string.IsNullOrEmpty(textBox10.Text)

&& !string.IsNullOrEmpty(maskedTextBox1.Text)

&& !string.IsNullOrEmpty(textBox12.Text)

&& !string.IsNullOrEmpty(textBox13.Text);

}

private void SaveGuestToDatabase()

{

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=HotelManagement;Integrated Security=True";

string query = @"

INSERT INTO Guests

(LastName, FirstName, MiddleName, Address, Town, Aim, Passport, PassportDate, Region, Work, BirthYear, Picture, PhoneNumber)

VALUES

(@LastName, @FirstName, @MiddleName, @Address, @Town, @Aim, @Passport, @PassportDate, @Region, @Work, @BirthYear, @Picture, @PhoneNumber)";

using (SqlConnection connection = new SqlConnection(connectionString))

{

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

{

connection.Open();

command.Parameters.AddWithValue("@LastName", textBox3.Text);

command.Parameters.AddWithValue("@FirstName", textBox4.Text);

command.Parameters.AddWithValue("@MiddleName", textBox5.Text);

command.Parameters.AddWithValue("@Address", textBox8.Text);

command.Parameters.AddWithValue("@Town", textBox7.Text);

command.Parameters.AddWithValue("@Aim", textBox9.Text);

command.Parameters.AddWithValue("@Passport", textBox10.Text);

DateTime passportDate;

if (DateTime.TryParseExact(maskedTextBox1.Text, "yyyy-MM-dd", CultureInfo.InvariantCulture, DateTimeStyles.None, out passportDate))

{

command.Parameters.AddWithValue("@PassportDate", passportDate);

}

else

{

MessageBox.Show("Ошибка ввода даты паспорта.");

return;

}

command.Parameters.AddWithValue("@Region", textBox12.Text);

command.Parameters.AddWithValue("@Work", textBox13.Text);

DateTime BirthYear;

if (DateTime.TryParseExact(maskedTextBox2.Text, "yyyy-MM-dd", CultureInfo.InvariantCulture, DateTimeStyles.None, out BirthYear))

{

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

}

else

{

MessageBox.Show("Ошибка ввода даты рождения.");

return;

}

SqlParameter pictureParameter = new SqlParameter("@Picture", SqlDbType.VarBinary);

if (imageData != null && imageData.Length > 0)

{

pictureParameter.Value = imageData;

}

else

{

pictureParameter.Value = DBNull.Value;

}

command.Parameters.Add(pictureParameter);

command.Parameters.AddWithValue("@PhoneNumber", textBox2.Text);

command.ExecuteNonQuery();

}

}

}

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

{

if (ValidateReservationInputs())

{

MakeReservation();

MessageBox.Show("Бронирование успешно завершено!");

}

else

{

MessageBox.Show("Пожалуйста, заполните все обязательные поля.");

}

}

private bool ValidateReservationInputs()

{

return !string.IsNullOrEmpty(textBox3.Text)

&& !string.IsNullOrEmpty(textBox4.Text)

&& !string.IsNullOrEmpty(textBox5.Text)

&& !string.IsNullOrEmpty(maskedTextBox3.Text)

&& !string.IsNullOrEmpty(maskedTextBox4.Text)

&& !string.IsNullOrEmpty(textBox15.Text)

&& !string.IsNullOrEmpty(textBox14.Text)

&& !string.IsNullOrEmpty(textBox16.Text)

&& !string.IsNullOrEmpty(textBox17.Text);

}

private void MakeReservation()

{

int selectedRoomNumber = int.Parse(textBox1.Text);

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=HotelManagement;Integrated Security=True";

string query = @"

INSERT INTO Reservations

(RoomNumber, LastName, FirstName, MiddleName, Come, Leave, Money, Cash, Receipt, Comment, Registrar)

VALUES

(@RoomNumber, @LastName, @FirstName, @MiddleName, @Come, @Leave, @Money, @Cash, @Receipt, @Comment, @Registrar)";

using (SqlConnection connection = new SqlConnection(connectionString))

{

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

{

connection.Open();

command.Parameters.AddWithValue("@RoomNumber", selectedRoomNumber);

command.Parameters.AddWithValue("@LastName", textBox3.Text);

command.Parameters.AddWithValue("@FirstName", textBox4.Text);

command.Parameters.AddWithValue("@MiddleName", textBox5.Text);

DateTime ComeDate;

if (DateTime.TryParseExact(maskedTextBox3.Text, "yyyy-MM-dd", CultureInfo.InvariantCulture, DateTimeStyles.None, out ComeDate))

{

command.Parameters.AddWithValue("@Come", ComeDate);

}

else

{

MessageBox.Show("Ошибка ввода даты.");

return;

}

DateTime LeaveDate;

if (DateTime.TryParseExact(maskedTextBox4.Text, "yyyy-MM-dd", CultureInfo.InvariantCulture, DateTimeStyles.None, out LeaveDate))

{

command.Parameters.AddWithValue("@Leave", LeaveDate);

}

else

{

MessageBox.Show("Ошибка ввода даты.");

return;

}

command.Parameters.AddWithValue("@Money", textBox15.Text);

command.Parameters.AddWithValue("@Cash", checkBox1.Checked);

command.Parameters.AddWithValue("@Receipt", textBox14.Text);

command.Parameters.AddWithValue("@Comment", textBox16.Text);

command.Parameters.AddWithValue("@Registrar", textBox17.Text);

command.ExecuteNonQuery();

guestFormInstance.RemoveReservedRoom(selectedRoomNumber);

currentReceiptNumber++;

textBox14.Text = currentReceiptNumber.ToString("D4");

}

}

}

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

{

}

}

}  
  
  
  
  
  
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;

using System.Data.SqlClient;

using System.Collections;

using System.IO;

namespace Деятельность\_гостинцы

{

public partial class GuestForm : Form

{

public static event EventHandler<ReservationEventArgs> ReservationRequested;

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=HotelManagement;Integrated Security=True;";

private string query = "SELECT Number, Rooms, Storey, TV, Fridge, Bed, Type, Balcony FROM Rooms WHERE 1=1 ";

public GuestForm()

{

InitializeComponent();

}

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

{

dataGridView1.ReadOnly = true;

this.roomsTableAdapter.Fill(this.dataSet2.Rooms);

foreach (DataGridViewRow row in dataGridView1.Rows)

{

int roomNumber = Convert.ToInt32(row.Cells[0].Value.ToString());

comboBox1.Items.Add(roomNumber);

maskedTextBox1.Mask = "+7 (999) 000-0000";

}

}

public void RemoveReservedRoom(int roomNumberToRemove)

{

if (dataGridView1.Columns.Contains("Number"))

{

DataGridViewColumn numberColumn = dataGridView1.Columns["Number"];

int columnIndex = numberColumn.Index;

foreach (DataGridViewRow row in dataGridView1.Rows)

{

if (row.Cells[columnIndex].Value != null && Convert.ToInt32(row.Cells[columnIndex].Value) == roomNumberToRemove)

{

dataGridView1.Rows.Remove(row);

UpdateComboBox(roomNumberToRemove);

break;

}

}

}

else

{

MessageBox.Show("Столбец 'Number' не найден в DataGridView.", "Ошибка удаления");

}

}

private void UpdateComboBox(int roomNumberToRemove)

{

if (comboBox1.InvokeRequired)

{

comboBox1.Invoke(new Action(() => UpdateComboBox(roomNumberToRemove)));

}

else

{

if (comboBox1.Items.Contains(roomNumberToRemove))

{

comboBox1.Items.Remove(roomNumberToRemove);

}

}

}

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

{

}

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

{

}

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

{

int rooms, storey, beds;

bool tv, fridge, balcony;

int.TryParse(textBox1.Text, out rooms);

int.TryParse(textBox2.Text, out storey);

int.TryParse(textBox3.Text, out beds);

tv = checkBox1.Checked;

fridge = checkBox2.Checked;

balcony = checkBox3.Checked;

if (rooms <= 0 && storey <= 0 && beds <= 0 && !tv && !fridge && !balcony)

{

string defaultQuery = "SELECT Number, Rooms, Storey, TV, Fridge, Bed, Type, Balcony FROM Rooms";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand defaultCommand = new SqlCommand(defaultQuery, connection);

SqlDataAdapter defaultAdapter = new SqlDataAdapter(defaultCommand);

DataTable defaultDataTable = new DataTable();

defaultAdapter.Fill(defaultDataTable);

dataGridView1.DataSource = defaultDataTable;

}

}

else

{

string query = "SELECT Number, Rooms, Storey, TV, Fridge, Bed, Type, Balcony FROM Rooms WHERE 1=1 ";

if (storey > 0)

{

query += "AND Storey = @Storey ";

}

if (rooms > 0)

{

query += "AND Rooms = @Rooms ";

}

if (tv)

{

query += "AND TV = @TV ";

}

if (fridge)

{

query += "AND Fridge = @Fridge ";

}

if (beds > 0)

{

query += "AND Bed = @Beds ";

}

if (balcony)

{

query += "AND Balcony = @Balcony ";

}

query += "ORDER BY ";

if (storey > 0)

{

query += "Storey ASC, ";

}

if (rooms > 0)

{

query += "Rooms ASC, ";

}

if (tv)

{

query += "TV DESC, ";

}

if (fridge)

{

query += "Fridge DESC, ";

}

if (beds > 0)

{

query += "Bed ASC, ";

}

if (balcony)

{

query += "Balcony DESC, ";

}

query = query.TrimEnd(',', ' ');

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (storey > 0)

{

command.Parameters.AddWithValue("@Storey", storey);

}

if (rooms > 0)

{

command.Parameters.AddWithValue("@Rooms", rooms);

}

if (tv)

{

command.Parameters.AddWithValue("@TV", tv);

}

if (fridge)

{

command.Parameters.AddWithValue("@Fridge", fridge);

}

if (beds > 0)

{

command.Parameters.AddWithValue("@Beds", beds);

}

if (balcony)

{

command.Parameters.AddWithValue("@Balcony", balcony);

}

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

dataGridView1.DataSource = dataTable;

}

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

int selectedRoomNumber = Convert.ToInt32(comboBox1.SelectedItem);

string phoneNumber = maskedTextBox1.Text;

ReservationRequested?.Invoke(this, new ReservationEventArgs(selectedRoomNumber, phoneNumber));

SaveReservationData(selectedRoomNumber, phoneNumber);

}

protected virtual void OnReservationRequested(ReservationEventArgs e)

{

ReservationRequested?.Invoke(this, e);

}

public class ReservationEventArgs : EventArgs

{

public int RoomNumber { get; }

public string PhoneNumber { get; }

public ReservationEventArgs(int roomNumber, string phoneNumber)

{

RoomNumber = roomNumber;

PhoneNumber = phoneNumber;

}

}

private void SaveReservationData(int roomNumber, string phoneNumber)

{

string filePath = "ReservationData.txt";

using (StreamWriter writer = new StreamWriter(filePath))

{

writer.WriteLine($"{roomNumber},{phoneNumber}");

}

}

private void maskedTextBox1\_MaskInputRejected(object sender, MaskInputRejectedEventArgs e)

{

}

}

}

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;

using System.Data.SqlClient;

namespace Деятельность\_гостинцы

{

public partial class AdminForm : Form

{

private const string ConnectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=HotelManagement;Integrated Security=True;";

public AdminForm()

{

InitializeComponent();

}

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

{

LoadTables();

}

private void LoadTables()

{

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

DataTable schema = connection.GetSchema("Tables");

foreach (DataRow row in schema.Rows)

{

string tableName = row["TABLE\_NAME"].ToString();

comboBox1.Items.Add(tableName);

}

}

if (comboBox1.Items.Count > 0)

{

comboBox1.SelectedIndex = 0;

LoadTableData(comboBox1.SelectedItem.ToString());

}

}

private void LoadTableData(string tableName)

{

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

string query = $"SELECT \* FROM {tableName}";

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

{

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

dataGridView1.DataSource = dataTable;

}

}

}

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

{

}

private void comboBox1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

string selectedTable = comboBox1.SelectedItem.ToString();

LoadTableData(selectedTable);

}

}

}