ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

Класс DataBase

CREATE DATABASE WWII;

USE WWII;

-- Таблица для хранения информации о военных частях

CREATE TABLE MilitaryUnits (

MilitaryUnitID INT PRIMARY KEY IDENTITY(1,1),

UnitName NVARCHAR(255) NOT NULL,

Description NVARCHAR(500)

);

-- Таблица для хранения информации о событиях войны

CREATE TABLE WarEvents (

WarEventID INT PRIMARY KEY IDENTITY(1,1),

EventName NVARCHAR(255) NOT NULL,

EventDate DATE,

EventLocation NVARCHAR(255),

Description NVARCHAR(500)

);

-- Таблица для хранения информации о ветеранах

CREATE TABLE Veterans (

VeteranID INT PRIMARY KEY IDENTITY(1,1),

FullName NVARCHAR(255) NOT NULL,

BirthDate DATE,

DeathDate DATE,

MilitaryRank NVARCHAR(100),

UnitID INT,

FOREIGN KEY (UnitID) REFERENCES MilitaryUnits(MilitaryUnitID)

);

-- Таблица для хранения информации о медалях и наградах

CREATE TABLE Medals (

MedalID INT PRIMARY KEY IDENTITY(1,1),

MedalName NVARCHAR(255) NOT NULL,

Description NVARCHAR(500)

);

-- Таблица для связи ветеранов и наград

CREATE TABLE VeteranMedals (

VeteranMedalID INT PRIMARY KEY IDENTITY(1,1),

VeteranID INT,

MedalID INT,

AwardDate DATE,

FOREIGN KEY (VeteranID) REFERENCES Veterans(VeteranID),

FOREIGN KEY (MedalID) REFERENCES Medals(MedalID)

);

-- Таблица для хранения информации о военной технике

CREATE TABLE MilitaryEquipment (

MilitaryEquipmentID INT PRIMARY KEY IDENTITY(1,1),

EquipmentName NVARCHAR(255) NOT NULL,

EquipmentType NVARCHAR(100),

Description NVARCHAR(500)

);

-- Таблица для хранения информации о военных маршрутах

CREATE TABLE MilitaryRoutes (

MilitaryRouteID INT PRIMARY KEY IDENTITY(1,1),

RouteName NVARCHAR(255) NOT NULL,

StartLocation NVARCHAR(255),

EndLocation NVARCHAR(255),

Description NVARCHAR(500)

);

-- Таблица для связи между событиями и техникой

CREATE TABLE EventEquipment (

EventEquipmentID INT PRIMARY KEY IDENTITY(1,1),

EventID INT,

EquipmentID INT,

FOREIGN KEY (EventID) REFERENCES WarEvents(WarEventID),

FOREIGN KEY (EquipmentID) REFERENCES MilitaryEquipment(MilitaryEquipmentID)

);

CREATE TABLE Registration (

RegistrationID INT PRIMARY KEY IDENTITY(1,1),

UserLogin NVARCHAR(50) NOT NULL,

UserPassword NVARCHAR(50) NOT NULL,

IsAdmin BIT DEFAULT 0

);

INSERT INTO MilitaryUnits (UnitName, Description) VALUES

('Пулемет ДП-27', 'Дегтярев пехотный'),

('Пулемет ДТ', 'Дегтярев танковый'),

('Пулемет ДШК', 'Дегтярев-Шпагин крупнокалиберный'),

('Автомат ППД-34/38', 'Пистолет-пулемет Дегтярева'),

('Автомат ППД-40', 'Автомат ППД-40'),

('ПТРД-41', 'Противотанковое ружье Дегтярева'),

('Т-26', 'Танк, оснащался пулеметом ДТ'),

('БТ-7', 'Танк, оснащался пулеметом ДТ'),

('Т-34', 'Танк, оснащался пулеметом ДТ'),

('КВ-1', 'Танк, оснащался пулеметом ДТ'),

('ИС-2', 'Танк, оснащался пулеметом ДШК'),

('ИЛ-2', 'Самолет, оснащался пулеметом ДШК'),

('Пе-8', 'Тяжелый бомбардировщик, вооруженный ДШК'),

('Ла-5', 'Самолет, в некоторых модификациях мог использовать ДШК'),

('Пулемет Мадсена', 'Пулемет Мадсена'),

('К-125', 'Автомобиль или техника К-125'),

('АК-47', 'Автомат АК-47'),

('ПТУРС "Шмель"', 'Противотанковая управляемая ракета "Шмель"'),

('ПЗРК "Стрела-2"', 'Переносной зенитный ракетный комплекс "Стрела-2"'),

('Ракета B-75', 'Ракета B-75');

-- Вставка данных в таблицу MilitaryEquipment (военная техника)

INSERT INTO MilitaryEquipment (EquipmentName, EquipmentType, Description) VALUES

('Пулемет ДП-27', 'Пехотный пулемет', 'Дегтярев пехотный'),

('Пулемет ДТ', 'Танковый пулемет', 'Дегтярев танковый'),

('Пулемет ДШК', 'Крупнокалиберный пулемет', 'Дегтярев-Шпагин крупнокалиберный'),

('Автомат ППД-34/38', 'Пистолет-пулемет', 'Пистолет-пулемет Дегтярева'),

('Автомат ППД-40', 'Пистолет-пулемет', 'Автомат ППД-40'),

('ПТРД-41', 'Противотанковое ружье', 'Противотанковое ружье Дегтярева'),

('Т-26', 'Танк', 'Оснащался пулеметом ДТ'),

('БТ-7', 'Танк', 'Оснащался пулеметом ДТ'),

('Т-34', 'Танк', 'Оснащался пулеметом ДТ'),

('КВ-1', 'Танк', 'Оснащался пулеметом ДТ'),

('ИС-2', 'Танк', 'Оснащался пулеметом ДШК'),

('ИЛ-2', 'Самолет', 'Оснащался пулеметом ДШК'),

('Пе-8', 'Бомбардировщик', 'Тяжелый бомбардировщик, вооруженный ДШК'),

('Ла-5', 'Самолет', 'Мог использовать пулемет ДШК в некоторых модификациях'),

('Пулемет Мадсена', 'Пулемет', 'Пулемет Мадсена'),

('К-125', 'Техника', 'Автомобиль или техника К-125'),

('АК-47', 'Автомат', 'Автомат АК-47'),

('ПТУРС "Шмель"', 'Противотанковая ракета', 'Противотанковая управляемая ракета "Шмель"'),

('ПЗРК "Стрела-2"', 'Зенитная ракета', 'Переносной зенитный ракетный комплекс "Стрела-2"'),

('Ракета B-75', 'Ракета', 'Ракета B-75');

-- Вставка данных в таблицу WarEvents (события войны)

INSERT INTO WarEvents (EventName, EventDate, EventLocation, Description) VALUES

('Битва под Москвой', '1941-10-02', 'Московская область', 'Массированные танковые атаки и использование пулеметов ДТ на танках Т-34 и других моделях.'),

('Блокада Ленинграда', '1941-09-08', 'Ленинград', 'Сопротивление блокады, использование различных видов техники, включая пулеметы и танки'),

('Курская битва', '1943-07-05', 'Курская область', 'Применение танков ИС-2 и Т-34, а также авиации с вооружением ДШК'),

('Операция Багратион', '1944-06-22', 'Белоруссия', 'Активное применение пушек, танков и авиации в наступательных действиях'),

('Победа над Японией', '1945-08-15', 'Дальний Восток', 'Применение новых образцов вооружений, включая автомат АК-47 и ПЗРК "Стрела-2"');

-- Вставка данных в таблицу Medals (медали и награды)

INSERT INTO Medals (MedalName, Description) VALUES

('Орден Отечественной войны', 'Высшая награда для солдат и командиров во время Великой Отечественной войны'),

('Медаль "За отвагу"', 'Медаль, вручаемая за личную храбрость и мужество'),

('Орден Красного Знамени', 'Высшая награда для за особые боевые заслуги'),

('Медаль "За победу над Германией"', 'Медаль, вручавшаяся за победу в Великой Отечественной войне');

INSERT INTO Veterans (FullName, BirthDate, DeathDate, MilitaryRank, UnitID) VALUES

('Иванов Иван Иванович', '1905-03-12', '1985-11-23', 'Старший лейтенант', 1),

('Петров Петр Петрович', '1908-06-05', '1970-02-18', 'Лейтенант', 2),

('Сидоров Сидор Сидорович', '1910-07-22', '1975-09-11', 'Майор', 3),

('Кузнецов Александр Алексеевич', '1912-09-30', '1965-05-06', 'Подполковник', 4),

('Михайлова Мария Ивановна', '1915-01-16', '1982-08-09', 'Капитан', 5);

INSERT INTO VeteranMedals (VeteranID, MedalID, AwardDate) VALUES

(1, 1, '1942-12-01'),

(1, 2, '1943-04-15'),

(2, 3, '1944-07-10'),

(3, 4, '1945-05-01'),

(4, 1, '1943-03-19'),

(5, 2, '1944-11-23');

INSERT INTO MilitaryRoutes (RouteName, StartLocation, EndLocation, Description) VALUES

('Маршрут 1', 'Москва', 'Смоленск', 'Танк Т-34 и пулемет ДП-27 использовались на этом маршруте в ходе операции'),

('Маршрут 2', 'Киев', 'Харьков', 'Военная техника ПТРД-41 применялась против немецких танков'),

('Маршрут 3', 'Севастополь', 'Краснодар', 'Применение авиации с оружием ДШК'),

('Маршрут 4', 'Минск', 'Вильнюс', 'Танк ИС-2 и стрелковое вооружение использовались для защиты этого маршрута'),

('Маршрут 5', 'Сталинград', 'Ростов-на-Дону', 'Сопротивление вражеским силам, применение пушек и артиллерии');

INSERT INTO EventEquipment (EventID, EquipmentID) VALUES

(1, 7), -- Битва под Москвой использовались танки Т-34

(1, 8), -- Битва под Москвой использовались танки БТ-7

(2, 12), -- Блокада Ленинграда использовалась авиация ИЛ-2

(3, 10), -- Курская битва использовались танки КВ-1

(4, 15), -- Операция Багратион использовались Пулеметы Мадсена

(5, 18); -- Победа над Японией использовались ПЗРК "Стрела-2"

INSERT INTO Registration (UserLogin, UserPassword, IsAdmin) VALUES

('admin', 'admin', 1),

('user', 'user', 0);

SELECT \* FROM MilitaryUnits;

SELECT \* FROM WarEvents;

SELECT \* FROM Veterans;

SELECT \* FROM Medals;

SELECT \* FROM VeteranMedals;

SELECT \* FROM MilitaryEquipment;

SELECT \* FROM MilitaryRoutes;

SELECT \* FROM EventEquipment;

SELECT \* FROM Registration;

-- Отчет 1: Список ветеранов с их наградами

SELECT v.FullName, v.MilitaryRank, m.MedalName, vm.AwardDate

FROM Veterans v

JOIN VeteranMedals vm ON v.VeteranID = vm.VeteranID

JOIN Medals m ON vm.MedalID = m.MedalID;

-- Отчет 2: Список всех событий войны и задействованной техники

SELECT we.EventName, we.EventDate, me.EquipmentName

FROM WarEvents we

JOIN EventEquipment ee ON we.WarEventID = ee.EventID

JOIN MilitaryEquipment me ON ee.EquipmentID = me.MilitaryEquipmentID;

-- Отчет 3: Список военных маршрутов и их локаций

SELECT RouteName, StartLocation, EndLocation, Description

FROM MilitaryRoutes;

Класс DataBase

using System.Data.SqlClient;

namespace WWII

{

public class DataBase

{

private readonly SqlConnection sqlConnection = new(@"Data Source=DESKTOP-06EKML1\SQLEXPRESS;Initial Catalog=WWII;Integrated Security=True");

/// <summary>

/// OpenConnection() вызывается при открытии соединения с базой данных

/// </summary>

public void OpenConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Closed)

{

sqlConnection.Open();

}

}

/// <summary>

/// CloseConnection() вызывается при закрытии соединения с базой данных

/// </summary>

public void CloseConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Open)

{

sqlConnection.Close();

}

}

/// <summary>

/// GetConnection() вызывается при получении соединения с базой данных

/// </summary>

/// <returns></returns>

public SqlConnection GetConnection()

{

return sqlConnection;

}

}

}

Класс Login

using System.Data;

using System.Data.SqlClient;

namespace WWII

{

public partial class Login : Form

{

private readonly DataBase dataBase = new();

public Login()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

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

{

var loginUser = textBoxLogin.Text;

var passwordUser = textBoxPassword.Text;

SqlDataAdapter sqlDataAdapter = new();

DataTable dataTable = new();

string querystring = $"select RegistrationID, UserLogin, UserPassword, IsAdmin from Registration where UserLogin = '{loginUser}' and UserPassword = '{passwordUser}'";

SqlCommand sqlCommand = new(querystring, dataBase.GetConnection());

sqlDataAdapter.SelectCommand = sqlCommand;

sqlDataAdapter.Fill(dataTable);

if (dataTable.Rows.Count == 1)

{

MessageBox.Show("Вы успешно вошли!", "Успешно!", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form1 form1 = new(Convert.ToBoolean(dataTable.Rows[0]["IsAdmin"]));

Hide();

form1.ShowDialog();

Show();

}

else

{

MessageBox.Show("Такого аккаунта не существует!", "Аккаунта не существует!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

Signup formLogin = new();

Hide();

formLogin.ShowDialog();

Show();

}

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

{

textBoxLogin.Text = "";

textBoxPassword.Text = "";

}

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

{

if (textBoxPassword.PasswordChar == '•')

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword0;

textBoxPassword.PasswordChar = '\0';

}

else

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword1;

textBoxPassword.PasswordChar = '•';

}

}

}

}

Класс SignUp

using System.Data;

using System.Data.SqlClient;

namespace WWII

{

public partial class Signup : Form

{

private readonly DataBase dataBase = new();

public Signup()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

}

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

{

var login = textBoxLogin.Text;

var password = textBoxPassword.Text;

SqlDataAdapter sqlDataAdapter = new();

DataTable dataTable = new();

string querystringUsers = $"SELECT UserLogin FROM Registration WHERE UserLogin = '{login}'";

SqlCommand sqlCommandUsers = new(querystringUsers, dataBase.GetConnection());

sqlDataAdapter.SelectCommand = sqlCommandUsers;

sqlDataAdapter.Fill(dataTable);

if (dataTable.Rows.Count > 0)

{

MessageBox.Show("Такой логин уже существует!", "Ошибка!");

}

else

{

string querystring = $"INSERT INTO Registration(UserLogin, UserPassword) VALUES('{login}', '{password}')";

SqlCommand sqlCommand = new(querystring, dataBase.GetConnection());

dataBase.OpenConnection();

if (sqlCommand.ExecuteNonQuery() == 1 && textBoxLogin.Text != "")

{

MessageBox.Show("Аккаунт успешно создан!", "Успех!");

Login formLogin = new();

Close();

formLogin.ShowDialog();

}

else

{

MessageBox.Show("Аккаунт не создан!");

}

dataBase.CloseConnection();

}

}

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

{

textBoxLogin.Text = "";

textBoxPassword.Text = "";

}

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

{

if (textBoxPassword.PasswordChar == '•')

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword0;

textBoxPassword.PasswordChar = '\0';

}

else

{

buttonShow.BackgroundImage = Properties.Resources.ShowPassword1;

textBoxPassword.PasswordChar = '•';

}

}

}

}