**1 LoginForm.cs**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace EduApp

{

public partial class LoginForm : Form

{

SqlConnection connect = new SqlConnection(@"Data Source=DESKTOP-MBJQ0QD;Initial Catalog=Демо;Integrated Security=True;Connect Timeout=30;Encrypt=True;TrustServerCertificate=True");

public LoginForm()

{

InitializeComponent();

}

public void buttonLogin\_Click(object sender, EventArgs e)

{

string login = textBoxLogin.Text;

string password = textBoxPassword.Text;

try

{

connect.Open();

string query = "SELECT Role FROM Users WHERE Login = @login AND Password = @password";

SqlCommand cmd = new SqlCommand(query, connect);

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

cmd.Parameters.AddWithValue("@password", password);

var role = cmd.ExecuteScalar();

if (role == null)

{

MessageBox.Show("Неверный логин или пароль");

return;

}

if (role.ToString() == "Teacher")

{

TeacherForm tf = new TeacherForm(login);

tf.Show();

this.Hide();

}

else if (role.ToString() == "Student")

{

StudentForm sf = new StudentForm(login);

sf.Show();

this.Hide();

}

else

{

MessageBox.Show("Неизвестная роль пользователя");

}

}

catch (Exception ex)

{

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

}

finally

{

connect.Close();

}

}

}

}

**Модульное тестирование**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using EduApp;

namespace testMS

{

[TestClass]

public class testLoginForm

{

[TestMethod]

public void ValideTestLoginForm()

{

var loginForm = new LoginForm();

loginForm.textBoxLogin.Text = "преподаватель1";

loginForm.textBoxPassword.Text = "123";

loginForm.buttonLogin\_Click(null, null);

}

[TestMethod]

public void InvalideTestLoginForm()

{

var loginForm = new LoginForm();

loginForm.textBoxLogin.Text = "учитель";

loginForm.textBoxPassword.Text = "123";

loginForm.buttonLogin\_Click(null, null);

}

[TestMethod]

public void NullLoginForm()

{

var loginForm = new LoginForm();

loginForm.textBoxLogin.Text = "";

loginForm.textBoxPassword.Text = "";

loginForm.buttonLogin\_Click(null, null);

}

}

}

**StudentForm.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace EduApp

{

public partial class StudentForm : Form

{

private string login;

SqlConnection connect = new SqlConnection(@"Data Source=DESKTOP-MBJQ0QD;Initial Catalog=Демо;Integrated Security=True;Connect Timeout=30;Encrypt=True;TrustServerCertificate=True");

public StudentForm(string login)

{

InitializeComponent();

this.login = login;

LoadGrades();

}

private void LoadGrades()

{

try

{

connect.Open();

string query = @"SELECT d.Название, o.Оценка

FROM Оценки o

JOIN Дисциплина d ON d.Код\_дисциплины = o.Код\_дисциплины

JOIN Студент s ON s.Номер\_студента = o.Номер\_студента

JOIN Users u ON u.UserId = s.Номер\_студента

WHERE u.Login = @login";

SqlDataAdapter da = new SqlDataAdapter(query, connect);

da.SelectCommand.Parameters.AddWithValue("@login", login);

DataTable table = new DataTable();

da.Fill(table);

dataGridView1.DataSource = table;

}

catch (Exception ex)

{

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

}

finally

{

connect.Close();

}

}

}

}

**TeacherForm.cs**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace EduApp

{

public partial class TeacherForm : Form

{

private readonly string login;

private readonly string connectionString = @"Data Source=DESKTOP-MBJQ0QD;Initial Catalog=Демо;Integrated Security=True;Connect Timeout=30;Encrypt=True;TrustServerCertificate=True";

public TeacherForm(string login)

{

InitializeComponent();

this.login = login;

LoadDisciplines();

}

private void LoadDisciplines()

{

comboBoxDisciplines.Items.Clear();

string query = @"SELECT DISTINCT d.Название

FROM Дисциплина d

JOIN Нагрузка\_преподавателя n ON d.Код\_дисциплины = n.Код\_дисциплины

JOIN Преподаватель p ON p.Номер\_преподавателя = n.Номер\_преподавателя

JOIN Users u ON u.UserId = p.Номер\_преподавателя

WHERE u.Login = @login";

try

{

using (SqlConnection connect = new SqlConnection(connectionString))

using (SqlCommand cmd = new SqlCommand(query, connect))

{

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

connect.Open();

using (SqlDataReader reader = cmd.ExecuteReader())

{

while (reader.Read())

{

comboBoxDisciplines.Items.Add(reader.GetString(0));

}

}

}

if (comboBoxDisciplines.Items.Count > 0)

comboBoxDisciplines.SelectedIndex = 0;

}

catch (Exception ex)

{

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

}

}

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

{

if (!ValidateInput(textBoxStudentId.Text, "ID студента")) return;

string studentId = textBoxStudentId.Text;

string discipline = comboBoxDisciplines.SelectedItem.ToString();

string query = @"SELECT d.Название AS 'Дисциплина', o.Оценка

FROM Оценки o

JOIN Дисциплина d ON o.Код\_дисциплины = d.Код\_дисциплины

WHERE o.Номер\_студента = @studentId AND d.Название = @discipline";

LoadData(query, ("@studentId", studentId), ("@discipline", discipline));

}

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

{

if (!ValidateInput(textBoxGroupId.Text, "ID группы")) return;

string groupId = textBoxGroupId.Text;

string discipline = comboBoxDisciplines.SelectedItem.ToString();

string query = @"SELECT s.Фамилия AS 'Фамилия студента', d.Название AS 'Дисциплина', o.Оценка

FROM Оценки o

JOIN Студент s ON s.Номер\_студента = o.Номер\_студента

JOIN Дисциплина d ON o.Код\_дисциплины = d.Код\_дисциплины

WHERE s.Номер\_группы = @groupId AND d.Название = @discipline";

LoadData(query, ("@groupId", groupId), ("@discipline", discipline));

}

private void LoadData(string query, params (string name, object value)[] parameters)

{

DataTable table = new DataTable();

try

{

using (SqlConnection connect = new SqlConnection(connectionString))

using (SqlDataAdapter adapter = new SqlDataAdapter(query, connect))

{

foreach (var param in parameters)

{

adapter.SelectCommand.Parameters.AddWithValue(param.name, param.value);

}

adapter.Fill(table);

dataGridView1.DataSource = table;

}

}

catch (Exception ex)

{

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

}

}

private bool ValidateInput(string input, string fieldName)

{

if (string.IsNullOrWhiteSpace(input))

{

MessageBox.Show($"Введите {fieldName}");

return false;

}

if (comboBoxDisciplines.SelectedItem == null)

{

MessageBox.Show("Выберите дисциплину");

return false;

}

return true;

}

}

}

**2 LOGINFORM.CS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Education

{

public partial class LoginForm : Form

{

public LoginForm()

{

InitializeComponent();

}

public void btnLogin\_Click(object sender, EventArgs e)

{

string login = txtLogin.Text;

string password = txtPassword.Text;

using (SqlConnection conn = new SqlConnection(@"Data Source=DESKTOP-MBJQ0QD;Initial Catalog=Демо;Integrated Security=True;Connect Timeout=30;Encrypt=True;TrustServerCertificate=True"))

{

conn.Open();

string query = "SELECT UserId, Role FROM Users WHERE Login = @login AND Password = @password";

SqlCommand cmd = new SqlCommand(query, conn);

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

cmd.Parameters.AddWithValue("@password", password);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

int userId = reader.GetInt32(0);

string role = reader.GetString(1);

this.Hide();

if (role == "Student")

new StudentForm(userId).Show();

else if (role == "Teacher")

new TeacherForm(userId).Show();

else

MessageBox.Show("Неизвестная роль.");

}

else

{

MessageBox.Show("Неверный логин или пароль.");

}

}

}

}

}

**STUDENTFORM.CS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Education

{

public partial class StudentForm : Form

{

private int userId;

private string connectionString = "Data Source=.;Initial Catalog=Демо;Integrated Security=True";

public StudentForm(int userId)

{

this.userId = userId;

InitializeComponent();

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = @"

SELECT d.Название AS Дисциплина, o.Оценка

FROM Оценки o

JOIN Дисциплина d ON o.Код\_дисциплины = d.Код\_дисциплины

WHERE o.Номер\_студента = (SELECT Номер\_студента FROM Студент WHERE UserId = @UserId)";

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

adapter.SelectCommand.Parameters.AddWithValue("@UserId", userId);

DataTable table = new DataTable();

adapter.Fill(table);

dgvGrades.DataSource = table;

}

}

}

}

**TEACHERFORM.CS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Education

{

public partial class TeacherForm : Form

{

private int userId;

private string connectionString = "Data Source=.;Initial Catalog=Демо;Integrated Security=True";

private DataTable gradesTable;

public TeacherForm(int userId)

{

this.userId = userId;

InitializeComponent();

}

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

{

LoadStudents();

LoadSubjects();

LoadGrades();

cboGrade.Items.AddRange(new object[] { "2", "3", "4", "5" });

}

private void LoadStudents()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = "SELECT Номер\_студента, Фамилия + ' ' + Имя AS ФИО FROM Студент";

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable table = new DataTable();

adapter.Fill(table);

cboStudents.DataSource = table;

cboStudents.DisplayMember = "ФИО";

cboStudents.ValueMember = "Номер\_студента";

}

}

private void LoadSubjects()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = @"

SELECT d.Код\_дисциплины, d.Название

FROM Дисциплина d

JOIN Нагрузка\_преподавателя n ON d.Код\_дисциплины = n.Код\_дисциплины

WHERE n.Номер\_преподавателя = (SELECT Номер\_преподавателя FROM Преподаватель WHERE UserId = @userId)";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@userId", userId);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable table = new DataTable();

adapter.Fill(table);

cboSubjects.DataSource = table;

cboSubjects.DisplayMember = "Название";

cboSubjects.ValueMember = "Код\_дисциплины";

}

}

private void LoadGrades()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = @"

SELECT s.Фамилия + ' ' + s.Имя AS Студент, d.Название AS Дисциплина, o.Оценка

FROM Оценки o

JOIN Студент s ON o.Номер\_студента = s.Номер\_студента

JOIN Дисциплина d ON o.Код\_дисциплины = d.Код\_дисциплины

WHERE o.Код\_дисциплины IN (

SELECT Код\_дисциплины

FROM Нагрузка\_преподавателя

WHERE Номер\_преподавателя = (SELECT Номер\_преподавателя FROM Преподаватель WHERE UserId = @userId)

)";

SqlCommand cmd = new SqlCommand(query, connection);

cmd.Parameters.AddWithValue("@userId", userId);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

gradesTable = new DataTable();

adapter.Fill(gradesTable);

dgvGrades.DataSource = gradesTable;

}

}

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

{

if (cboStudents.SelectedValue == null || (cboSubjects.SelectedValue == null || cboGrade.SelectedItem == null))

{

MessageBox.Show("Пожалуйста, выберите студента, дисциплину и оценку.");

return;

}

int studentId = Convert.ToInt32(cboStudents.SelectedValue);

int subjectId = Convert.ToInt32(cboSubjects.SelectedValue);

int grade = Convert.ToInt32(cboGrade.SelectedItem);

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

// Проверка: есть ли уже такая оценка

string checkQuery = @"SELECT COUNT(\*) FROM Оценки WHERE Номер\_студента = @studentId AND Код\_дисциплины = @subjectId";

SqlCommand checkCmd = new SqlCommand(checkQuery, connection);

checkCmd.Parameters.AddWithValue("@studentId", studentId);

checkCmd.Parameters.AddWithValue("@subjectId", subjectId);

// Вставка

string insertQuery = @"INSERT INTO Оценки (Номер\_студента, Код\_дисциплины, Оценка)

VALUES (@studentId, @subjectId, @grade)";

SqlCommand insertCmd = new SqlCommand(insertQuery, connection);

insertCmd.Parameters.AddWithValue("@studentId", studentId);

insertCmd.Parameters.AddWithValue("@subjectId", subjectId);

insertCmd.Parameters.AddWithValue("@grade", grade);

insertCmd.ExecuteNonQuery();

}

MessageBox.Show("Оценка успешно добавлена!");

LoadGrades(); // обновляем таблицу

}

}

}

**DASHBOARD.CS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace Dashboard

{

public partial class DashboardForm : Form

{

string connectionString = @"Data Source=DESKTOP-MBJQ0QD;Initial Catalog=Демо;Integrated Security=True;Connect Timeout=30;Encrypt=True;TrustServerCertificate=True";

public DashboardForm()

{

InitializeComponent();

}

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

{

LoadGroups();

}

private void LoadGroups()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

var command = new SqlCommand("SELECT Номер\_группы, Специальность FROM Группа", connection);

var reader = command.ExecuteReader();

var table = new DataTable();

table.Load(reader);

comboBoxGroups.DisplayMember = "Специальность";

comboBoxGroups.ValueMember = "Номер\_группы";

comboBoxGroups.DataSource = table;

}

}

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

{

if (comboBoxGroups.SelectedValue == null) return;

int groupId = Convert.ToInt32(comboBoxGroups.SelectedValue);

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

var command = new SqlCommand(@"

SELECT Номер\_студента, Фамилия + ' ' + Имя + ' ' + Отчество AS ФИО

FROM Студент WHERE Номер\_группы = @groupId", connection);

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

var reader = command.ExecuteReader();

var table = new DataTable();

table.Load(reader);

comboBoxStudents.DisplayMember = "ФИО";

comboBoxStudents.ValueMember = "Номер\_студента";

comboBoxStudents.DataSource = table;

}

}

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

{

if (comboBoxStudents.SelectedValue == null) return;

int studentId = Convert.ToInt32(comboBoxStudents.SelectedValue);

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

var adapter = new SqlDataAdapter(@"

SELECT Дисциплина.Название, Оценки.Оценка

FROM Оценки

INNER JOIN Дисциплина ON Оценки.Код\_дисциплины = Дисциплина.Код\_дисциплины

WHERE Оценки.Номер\_студента = @studentId", connection);

adapter.SelectCommand.Parameters.AddWithValue("@studentId", studentId);

var table = new DataTable();

adapter.Fill(table);

dataGridViewGrades.DataSource = table;

chartGrades.Series.Clear();

var series = new Series("Оценки")

{

ChartType = SeriesChartType.Column

};

foreach (DataRow row in table.Rows)

{

series.Points.AddXY(row["Название"].ToString(), Convert.ToInt32(row["Оценка"]));

}

chartGrades.Series.Add(series);

}

}

private Dictionary<string, double> GetGroupAverageGrades(int groupId)

{

var result = new Dictionary<string, double>();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

var command = new SqlCommand(@"

SELECT Дисциплина.Название, AVG(CAST(Оценки.Оценка AS FLOAT)) AS Средний\_Балл

FROM Оценки

INNER JOIN Дисциплина ON Оценки.Код\_дисциплины = Дисциплина.Код\_дисциплины

INNER JOIN Студент ON Оценки.Номер\_студента = Студент.Номер\_студента

WHERE Студент.Номер\_группы = @groupId

GROUP BY Дисциплина.Название", connection);

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

var reader = command.ExecuteReader();

while (reader.Read())

{

string subject = reader.GetString(0);

double average = reader.GetDouble(1);

result[subject] = average;

}

}

return result;

}

}

}