**Код подключения базы данных**

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StaffOfSamokat

{

internal class Database //класс подключения к базе данных

{

SqlConnection con = new SqlConnection(@"Data Source=LAPTOP-74E9C0IK; Initial Catalog=StaffOfSamokat; Integrated Security=true");

public void OpenConnection()

{

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

con.Open();

}

public void CloseConnection()

{

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

con.Close();

}

public SqlConnection GetConnection()

{

return con;

}

}

}

**Код формы AuthorizationForm**

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 StaffOfSamokat

{

public partial class AuthorizationForm : Form

{

Database db = new Database(); //подключение к бд

public AuthorizationForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

}

public static int ID\_User { get; set; }

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

{

PasswordTextBox.PasswordChar = '\*'; //скрытие пароля символом

}

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

{

var login = LoginTextBox.Text;

var password = PasswordTextBox.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string query = $"select \* from [User] where Login ='{login}' and Password ='{password}'"; //выборка данных из таблицы Пользователь

SqlCommand command = new SqlCommand(query, db.GetConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count == 1) //проверка наличия такого пользователя

{

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

if (login=="m" & password == "m")

{

ManagerMainForm managerMainForm = new ManagerMainForm();

managerMainForm.Show(); //переход к главной форме для менеджера

this.Hide();

}

else

{

db.OpenConnection();

SqlDataReader dataReader = command.ExecuteReader();

while (dataReader.Read()) //вывод данных вакансии

{

AuthorizationForm.ID\_User = (int)dataReader["ID\_User"];

}

db.CloseConnection();

UserMainForm userMainForm = new UserMainForm();

userMainForm.Show(); //переход к главной форме для пользователя

this.Hide();

}

}

else

{

MessageBox.Show("Неверный логин или пароль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

RegistrationForm registrationForm= new RegistrationForm();

registrationForm.Show(); //переход к форме регистрации

this.Hide();

}

}

}

**Код формы RegistrationForm**

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 StaffOfSamokat

{

public partial class RegistrationForm : Form

{

Database db = new Database(); //подключение к бд

public RegistrationForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

}

private Boolean Check() //метод, проверяющий чтобы все поля были заполнены

{

var login = LoginTextBox.Text;

var password = PasswordTextBox.Text;

if (string.IsNullOrEmpty(login) || string.IsNullOrEmpty(password))

{

MessageBox.Show("Введите логин и пароль!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

else

{

return false;

}

}

private Boolean Repeat() //метод, проверяющий повторение пользователей

{

var login = LoginTextBox.Text;

var password = PasswordTextBox.Text;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string query = $"select \* from [User] where Login='{login}' and Password='{password}'"; //выбока данных из таблицы Пользователь

SqlCommand command = new SqlCommand(query, db.GetConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0) //проверка наличия такого пользователя

{

MessageBox.Show("Такой пользователь уже существует!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

else

{

return false;

}

}

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

{

if (Check()) //активация метода проверки пустых полей

{

return;

}

if (Repeat()) //активация метода проверки повторения пользователей

{

return;

}

var login = LoginTextBox.Text;

var password = PasswordTextBox.Text;

string query = $"insert into [User] (Login, Password) values ('{login}','{password}')"; //занесение пользователя в базу данных

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Регистрация прошла успешно!", "Успех!", MessageBoxButtons.OK, MessageBoxIcon.Information);

AuthorizationForm authorizationForm = new AuthorizationForm();

this.Close();

authorizationForm.Show(); //переход к форме авторизации

}

else

MessageBox.Show("Такой аккаунт не может быть создан", "Ошибка регистрации", MessageBoxButtons.OK, MessageBoxIcon.Information);

db.CloseConnection();

}

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

{

AuthorizationForm authorizationForm = new AuthorizationForm();

this.Close();

authorizationForm.Show(); //переход к форме авторизации

}

}

}

**Код формы UserMainForm**

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;

namespace StaffOfSamokat

{

public partial class UserMainForm : Form

{

public UserMainForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

}

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

{

ResumeForm resumeForm = new ResumeForm();

this.Close();

resumeForm.Show(); //переход к форме резюме

}

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

{

InformationForm informationForm = new InformationForm();

this.Close();

informationForm.Show(); //переход к форме информации

}

}

}

**Код формы InformationForm**

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;

namespace StaffOfSamokat

{

public partial class InformationForm : Form

{

public InformationForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

}

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

{

UserMainForm userMainForm = new UserMainForm();

this.Close();

userMainForm.Show(); //переход к главной форме пользователя

}

}

}

**Код формы ResumeForm**

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.Xml;

using System.Xml.Linq;

using System.Net;

namespace StaffOfSamokat

{

public partial class ResumeForm : Form

{

Database db = new Database(); //подключение к бд

public ResumeForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

bool hasResume = Presence(); //активация метода проверки наличия резюме у пользователя

string vacancyQuery = $"select name from Post"; //выбока вакансий

SqlCommand vacancyCommand = new SqlCommand(vacancyQuery, db.GetConnection());

db.OpenConnection();

SqlDataReader vacancyDataReader = vacancyCommand.ExecuteReader();

while (vacancyDataReader.Read()) //заполнение PostComboBox данными из запроса

{

string post = vacancyDataReader["Name"].ToString();

PostComboBox.Items.Add(post);

}

db.CloseConnection();

string cityQuery = $"select name from City"; //выбока городов

SqlCommand cityCommand = new SqlCommand(cityQuery, db.GetConnection());

db.OpenConnection();

SqlDataReader cityDataReader = cityCommand.ExecuteReader();

while (cityDataReader.Read()) //заполнение CityComboBox данными из запроса

{

string city = cityDataReader["Name"].ToString();

CityComboBox.Items.Add(city);

}

db.CloseConnection();

}

private void ReplaceButton(Button oldResumeButton, Button newResumeButton) //метод, изменяющий кнопку

{

Control parent = oldResumeButton.Parent;

parent.Controls.Remove(oldResumeButton);

parent.Controls.Add(newResumeButton);

newResumeButton.Location = oldResumeButton.Location;

newResumeButton.Size = oldResumeButton.Size;

newResumeButton.Anchor = oldResumeButton.Anchor;

newResumeButton.Font = oldResumeButton.Font;

newResumeButton.BackColor = oldResumeButton.BackColor;

newResumeButton.ForeColor = oldResumeButton.ForeColor;

newResumeButton.FlatStyle = oldResumeButton.FlatStyle;

newResumeButton.Text = "Изменить резюме";

}

private Boolean Presence() //метод, проверяющий наличие резюме у пользователя

{

int userId = AuthorizationForm.ID\_User;

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string query = $"select ID\_Resume, First\_Name, Last\_Name, Patronymic, Date\_Of\_Birth, Phone\_Number," +

$" Post.Name as Post, City.Name as City, CFZ.Address as CFZ,Experience, Status.Name as Status, ID\_User" +

$" from Resume join Post on Post = ID\_Post join City on City = ID\_City join" +

$" CFZ on CFZ = ID\_CFZ join Status on Status = ID\_Status where ID\_User='{userId}'"; //выбока резюме определенного пользователя

SqlCommand command = new SqlCommand(query, db.GetConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0) //проверка наличия резюме у пользователя

{

db.OpenConnection();

SqlDataReader dataReader = command.ExecuteReader();

while (dataReader.Read()) //вывод данных резюме

{

FirstNameTextBox.Text = dataReader[1].ToString();

LastNameTextBox.Text = dataReader[2].ToString();

PatronymicTextBox.Text = dataReader[3].ToString();

DateOfBirthTextBox.Text = DateTime.Parse(dataReader[4].ToString()).ToString("yyyy-MM-dd"); ;

PhoneNumberTextBox.Text = dataReader[5].ToString();

PostComboBox.Text = dataReader[6].ToString();

CityComboBox.Text = dataReader[7].ToString();

CFZComboBox.Text = dataReader[8].ToString();

ExperienceTextBox.Text = dataReader[9].ToString();

string status = dataReader[10].ToString();

StatusLabel.Text = $"Статус\nвашего\nрезюме\n{status}!";

Button ChangeResumeButton = new Button();

ReplaceButton(AddResumeButton, ChangeResumeButton);

ChangeResumeButton.Click += ChangeResumeButton\_Click;

}

db.CloseConnection();

return true;

}

else

{

return false;

}

}

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

{

CFZComboBox.Items.Clear();

string cityUser = CityComboBox.SelectedItem.ToString();

string CFZQuery = $"select CFZ.Address from CFZ join City on CFZ.City=City.ID\_City where" +

$" City.Name='{cityUser}'"; //выбока ЦФЗ в выбранном городе

SqlCommand CFZCommand = new SqlCommand(CFZQuery, db.GetConnection());

db.OpenConnection();

SqlDataReader CFZDataReader = CFZCommand.ExecuteReader();

while (CFZDataReader.Read()) //заполнение PostComboBox данными из запроса

{

string CFZ = CFZDataReader["Address"].ToString();

CFZComboBox.Items.Add(CFZ);

}

db.CloseConnection();

}

private Boolean Check() //метод, проверяющий чтобы все обязательные поля были заполнены

{

var firstName = FirstNameTextBox.Text;

var lastName = LastNameTextBox.Text;

var patronymic = PatronymicTextBox.Text;

var dateOfBirth = DateOfBirthTextBox.Text;

var phoneNumber = PhoneNumberTextBox.Text;

var post = PostComboBox.Text;

var city = CityComboBox.Text;

var CFZ = CFZComboBox.Text;

if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) || string.IsNullOrEmpty(patronymic)

|| string.IsNullOrEmpty(dateOfBirth) || string.IsNullOrEmpty(phoneNumber) || string.IsNullOrEmpty(post)

|| string.IsNullOrEmpty(city) || string.IsNullOrEmpty(CFZ))

{

MessageBox.Show("Вы заполнили не все обязательные поля!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

else

{

return false;

}

}

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

{

if (Check()) //активация метода проверки пустых полей

{

return;

}

var firstName = FirstNameTextBox.Text;

var lastName = LastNameTextBox.Text;

var patronymic = PatronymicTextBox.Text;

var dateOfBirth = DateOfBirthTextBox.Text;

var phoneNumber = PhoneNumberTextBox.Text;

var post = PostComboBox.SelectedItem.ToString();

var city = CityComboBox.SelectedItem.ToString();

var CFZ = CFZComboBox.SelectedItem.ToString();

var experience = ExperienceTextBox.Text;

int userId = AuthorizationForm.ID\_User;

string query = $"insert into Resume (First\_Name, Last\_Name, Patronymic, Date\_Of\_Birth," +

$" Phone\_Number, Post, City, CFZ, Experience, Status, ID\_User) " +

$"select '{firstName}','{lastName}','{patronymic}', '{dateOfBirth}', '{phoneNumber}'," +

$"post.ID\_Post, city.ID\_City, cfz.ID\_CFZ, '{experience}', 1, '{userId}'" +

$"from Post post join City city on 1=1 join CFZ cfz on 1=1" +

$"where post.Name = '{post}' and city.Name = '{city}' and cfz.Address = '{CFZ}'"; // занесение заполненного резюме в базу данных

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Данные сохранены!\nВаше резюме опубликовано.", "Успех!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

db.CloseConnection();

}

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

{

if (Check()) //активация метода проверки пустых полей

{

return;

}

var firstName = FirstNameTextBox.Text;

var lastName = LastNameTextBox.Text;

var patronymic = PatronymicTextBox.Text;

var dateOfBirth = DateOfBirthTextBox.Text;

var phoneNumber = PhoneNumberTextBox.Text;

var post = PostComboBox.Text;

var city = CityComboBox.Text;

var CFZ = CFZComboBox.Text;

var experience = ExperienceTextBox.Text;

int userId = AuthorizationForm.ID\_User;

string query = $"update Resume set First\_Name = '{firstName}', Last\_Name = '{lastName}'," +

$"Patronymic = '{patronymic}', Date\_Of\_Birth = '{dateOfBirth}',Phone\_Number = '{phoneNumber}'," +

$" Post = (select ID\_Post from Post where Name = '{post}'), " +

$"City = (select ID\_City from City where Name = '{city}'), " +

$"CFZ = (select ID\_CFZ from CFZ where Address = '{CFZ}'), Experience = '{experience}' " +

$"where ID\_User = '{userId}'"; // изменение резюме пользователя

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

if (command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Данные вашего резюме изменены!", "Успех!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

db.CloseConnection();

}

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

{

UserMainForm userMainForm = new UserMainForm();

this.Close();

userMainForm.Show(); //переход к главной форме пользователя

}

}

}

**Код формы ManagerMainForm**

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 StaffOfSamokat

{

public partial class ManagerMainForm : Form

{

Database db = new Database(); //подключение к бд

int selectedRow;

public ManagerMainForm()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen; //расположение по центру экрана

}

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

{

CreateColumns();

SelectRow(ResumeDataGridView);

}

private void CreateColumns()

{

ResumeDataGridView.Columns.Add("ID\_Resume", "ID\_Resume");

ResumeDataGridView.Columns["ID\_Resume"].Visible = false;

ResumeDataGridView.Columns.Add("First\_Name", "Имя");

ResumeDataGridView.Columns["First\_Name"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Last\_Name", "Фамилия");

ResumeDataGridView.Columns["Last\_Name"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Patronymic", "Отчество");

ResumeDataGridView.Columns["Patronymic"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Date\_Of\_Birth", "Дата рождения");

ResumeDataGridView.Columns["Date\_Of\_Birth"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Phone\_Number", "Номер телефона");

ResumeDataGridView.Columns["Phone\_Number"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Post", "Должность");

ResumeDataGridView.Columns["Post"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("City", "Город");

ResumeDataGridView.Columns["City"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("CFZ", "ЦФЗ");

ResumeDataGridView.Columns["CFZ"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns.Add("Experience", "Опыт работы");

ResumeDataGridView.Columns["Experience"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

ResumeDataGridView.Columns["Experience"].DefaultCellStyle.WrapMode = DataGridViewTriState.True;

ResumeDataGridView.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCellsExceptHeaders;

ResumeDataGridView.Columns.Add("Status", "Статус");

ResumeDataGridView.Columns["Status"].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

}

private void ReadRow(DataGridView dgv, IDataRecord record)

{

dgv.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetString(2), record.GetString(3),

record.GetDateTime(4).ToString("yyyy-MM-dd"), record.GetInt64(5), record.GetString(6),

record.GetString(7), record.GetString(8), record.GetString(9), record.GetString(10));

}

private void SelectRow(DataGridView dgv)

{

dgv.Rows.Clear();

string query = $"select ID\_Resume, First\_Name, Last\_Name, Patronymic, Date\_Of\_Birth, Phone\_Number," +

$" Post.Name as Post, City.Name as City, CFZ.Address as CFZ,Experience, Status.Name as Status" +

$" from Resume join Post on Post = ID\_Post join City on City = ID\_City join" +

$" CFZ on CFZ = ID\_CFZ join Status on Status = ID\_Status"; //выбока всех резюме

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadRow(dgv, reader);

}

reader.Close();

}

private void Search(DataGridView dgv)

{

dgv.Rows.Clear();

string searchQuery = $"select ID\_Resume, First\_Name, Last\_Name, Patronymic, Date\_Of\_Birth, Phone\_Number," +

$" Post.Name as Post, City.Name as City, CFZ.Address as CFZ,Experience, Status.Name as Status" +

$" from Resume join Post on Resume.Post = Post.ID\_Post join City on Resume.City = City.ID\_City join" +

$" CFZ on Resume.CFZ = CFZ.ID\_CFZ join Status on Resume.Status = Status.ID\_Status where concat (First\_Name," +

$" Last\_Name, Patronymic, convert(varchar, Date\_Of\_Birth, 102), convert(varchar, Phone\_Number), Post.Name, City.Name, CFZ.Address," +

$" Experience, Status.Name) like '%" + SearchTextBox.Text + "%'"; //выбока определенных резюме

SqlCommand command = new SqlCommand(searchQuery, db.GetConnection());

db.OpenConnection();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

ReadRow(dgv,reader);

}

reader.Close();

}

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

{

Search(ResumeDataGridView);

}

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

{

int rowIndex = ResumeDataGridView.CurrentRow.Index;

string idResume = ResumeDataGridView.Rows[rowIndex].Cells["ID\_Resume"].Value.ToString();

string firstName = ResumeDataGridView.Rows[rowIndex].Cells["First\_Name"].Value.ToString();

string lastName = ResumeDataGridView.Rows[rowIndex].Cells["Last\_Name"].Value.ToString();

string patronymic = ResumeDataGridView.Rows[rowIndex].Cells["Patronymic"].Value.ToString();

string post = ResumeDataGridView.Rows[rowIndex].Cells["Post"].Value.ToString();

string phoneNumber = ResumeDataGridView.Rows[rowIndex].Cells["Phone\_number"].Value.ToString();

string status = ResumeDataGridView.Rows[rowIndex].Cells["Status"].Value.ToString();

if (status=="На рассмотрении")

{

MessageBox.Show("Резюме уже отправлено на рассмотрение!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

string query = $"update Resume set Status = 2 where ID\_Resume = {idResume}";

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

command.ExecuteNonQuery();

db.CloseConnection();

SelectRow(ResumeDataGridView);

MessageBox.Show($"Резюме соискателя {firstName} {lastName} {patronymic} на должность {post} на рассмотрении.\n" +

$"Соискатель уведомлен о вашем решении.\n" +

$"Обратитесь по номеру соискателя {phoneNumber}.", "Резюме отправлено на рассмотрение!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

int rowIndex = ResumeDataGridView.CurrentRow.Index;

string idResume = ResumeDataGridView.Rows[rowIndex].Cells["ID\_Resume"].Value.ToString();

string firstName = ResumeDataGridView.Rows[rowIndex].Cells["First\_Name"].Value.ToString();

string lastName = ResumeDataGridView.Rows[rowIndex].Cells["Last\_Name"].Value.ToString();

string patronymic = ResumeDataGridView.Rows[rowIndex].Cells["Patronymic"].Value.ToString();

string post = ResumeDataGridView.Rows[rowIndex].Cells["Post"].Value.ToString();

string status = ResumeDataGridView.Rows[rowIndex].Cells["Status"].Value.ToString();

if (status == "Принято")

{

MessageBox.Show("Резюме уже принято!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

string query = $"update Resume set Status = 3 where ID\_Resume = {idResume}";

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

command.ExecuteNonQuery();

db.CloseConnection();

SelectRow(ResumeDataGridView);

MessageBox.Show($"Вы приняли соискателя {firstName} {lastName} {patronymic} на должность {post}.",

"Соискатель принят!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

int rowIndex = ResumeDataGridView.CurrentRow.Index;

string idResume = ResumeDataGridView.Rows[rowIndex].Cells["ID\_Resume"].Value.ToString();

string firstName = ResumeDataGridView.Rows[rowIndex].Cells["First\_Name"].Value.ToString();

string lastName = ResumeDataGridView.Rows[rowIndex].Cells["Last\_Name"].Value.ToString();

string patronymic = ResumeDataGridView.Rows[rowIndex].Cells["Patronymic"].Value.ToString();

string post = ResumeDataGridView.Rows[rowIndex].Cells["Post"].Value.ToString();

string status = ResumeDataGridView.Rows[rowIndex].Cells["Status"].Value.ToString();

if (status == "Отклонено")

{

MessageBox.Show("Резюме уже отклонено!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

string query = $"update Resume set Status = 4 where ID\_Resume = {idResume}";

SqlCommand command = new SqlCommand(query, db.GetConnection());

db.OpenConnection();

command.ExecuteNonQuery();

db.CloseConnection();

SelectRow(ResumeDataGridView);

MessageBox.Show($"Вы отклонили резюме соискателя {firstName} {lastName} {patronymic} на должность {post}.\n" +

$"Соискатель уведомлен о вашем решении.", "Резюме отклонено!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

}