

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №11

по дисциплине

Основы алгоритмизации и программирования

Тема: «Разработка подсистемы авторизации и регистрации
информационной системы»

Работу выполнил

Студент гр.4238

Бусов В.Р.

Принял

Преподаватель Шмидт И.Р.

Казань 2024

ВАРИАНТ 4

Цель работы

Изучение способов реализации подсистемы авторизации и регистрации пользователей информационной системы (ИС) с использованием базы данных.

Задание на лабораторную работу

Разработать приложение, имеющее следующий функционал:

1.Регистрация пользователя согласно атрибутам сущности «Пользователь», определенным по вариантам. Сохранение данных о пользователе производить в базе данных.

2.Авторизация пользователя с загрузкой формы, где указывается информация о пользователе. Загрузку данных, необходимых для аутентификации, производить из базы данных.

3.Восстановление доступа к учетной записи пользователя путем отправки в электронную почту кода доступа.

Пароли, сохраняемые в базе данных, должны хешироваться.

Вариант 4 - Покупатель
имя, фамилия, дата рождения, e-mail, номер телефона, адрес доставки, дата первой покупки.

Результат выполнения работы

Для начала зарегистрируем нового пользователя, чтобы возможно было выполнять следующие действия с аккаунтом (Рисунок 1).

Регистрация

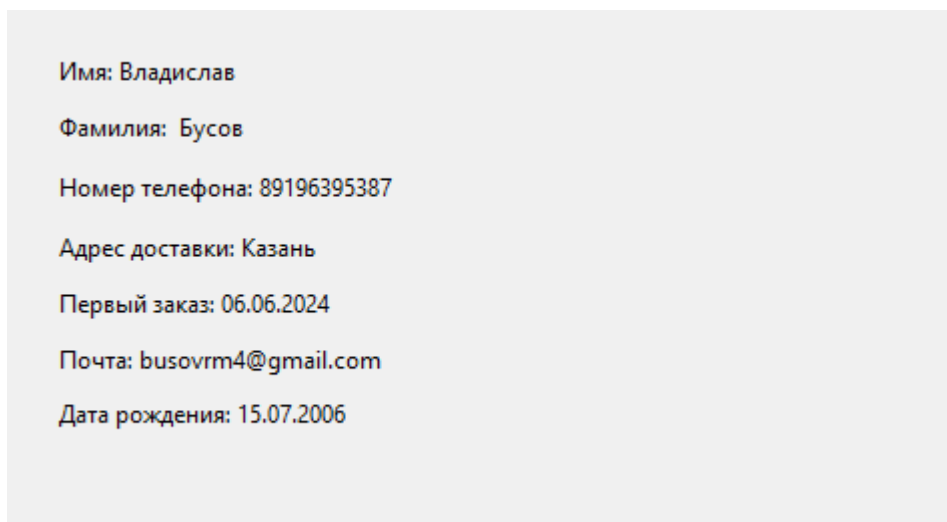
Владислав	Бусов
89196395387	Казань
15.07.2006	05.06.2024
123	busovrm4@gmail.com

Зарегистрироваться

[Уже есть аккаунт](#)

Рисунок 1 – Форма регистрации

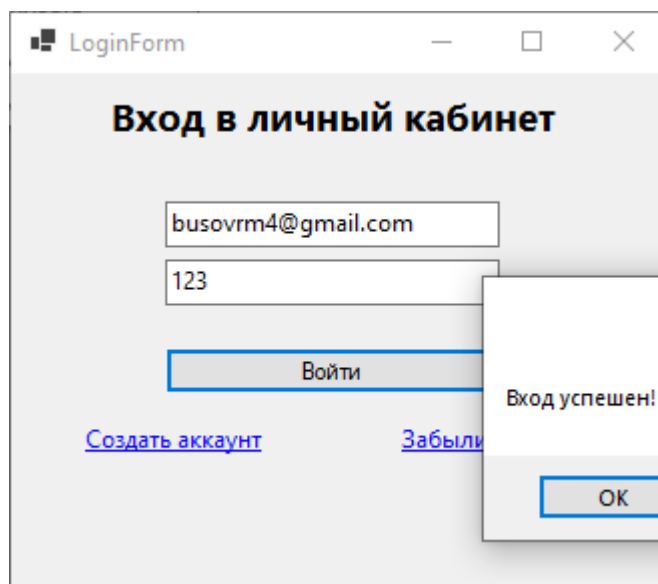
После успешной регистрации пользователь попадает на свою страницу с информацией (Рисунок 2).



Имя: Владислав
Фамилия: Бусов
Номер телефона: 89196395387
Адрес доставки: Казань
Первый заказ: 06.06.2024
Почта: busovrm4@gmail.com
Дата рождения: 15.07.2006

Рисунок 2 – Форма аккаунта

Далее попробуем зайти в уже созданный аккаунт пользователя. Введем наш логин(почту) и пароль (Рисунок 3).



LoginForm

Вход в личный кабинет

busovrm4@gmail.com

123

Войти

[Создать аккаунт](#) [Забыли](#)

Вход успешен!

OK

Рисунок 3 – Вход в аккаунт

Теперь нажмем на «Забыли пароль?». На зарегистрированную почту придет новый пароль, который нужно будет ввести для дальнейшего входа в аккаунт (Рисунок 4).

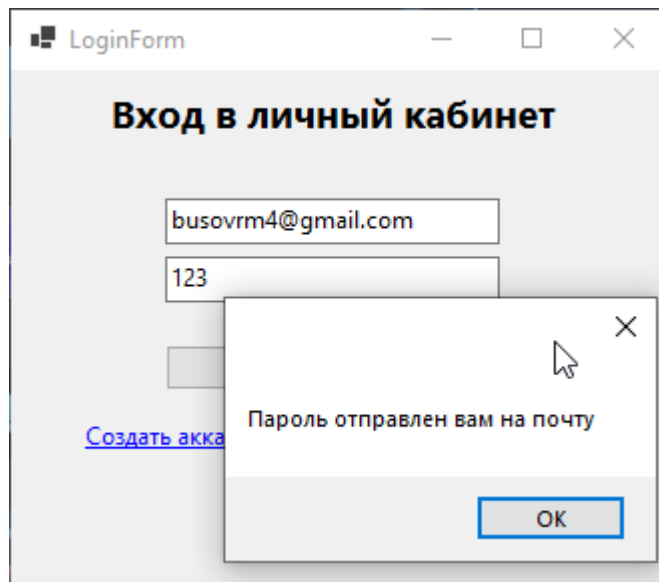


Рисунок 4 – Забытый пароль

Проверяем почту, на нее пришел новый пароль. Вводим его в форму аккаунта и успешно входим в аккаунт (Рисунок 5).

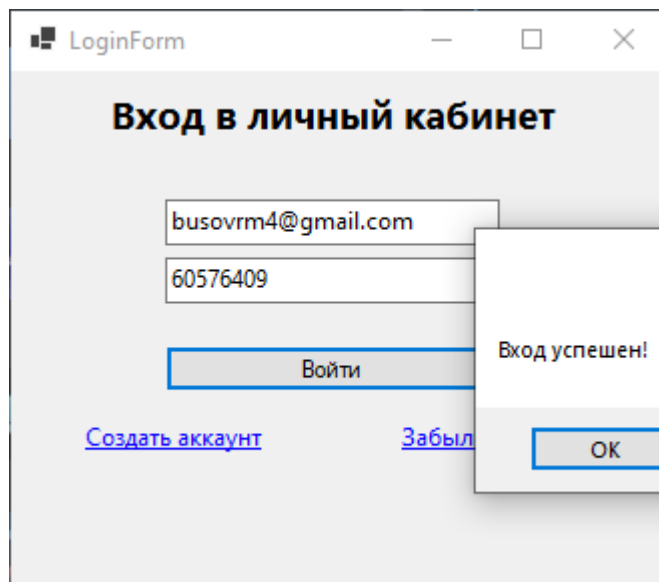


Рисунок 5 – Новый пароль

Вывод

Я изучил способы реализации подсистемы авторизации и регистрации пользователей информационной системы (ИС) с использованием базы данных.

Листинг

BuyerContext.cs

```
using System.Data.Entity;
using System.Data.Entity.Core.Mapping;
using System.Diagnostics.Eventing.Reader;
```

```

namespace WinFormsApp1
{
    public class BuyerContext : DbContext
    {
        public BuyerContext() : base() { }
        public DbSet<Buyer> Buyers { get; set; }

        public static bool email_check(string email, bool flag = true)
        {
            // flag - переменная, обозначающая проверку на наличие в бд
            // true - проверять, что такого пользователя нет (регистрация)
            // false - проверять, что такой пользователь точно есть (авторизация и почта)
            if (flag)
            {
                using (BuyerContext db = new BuyerContext())
                {
                    if (db.Buyers.Any(x => x.Email == email)) return false;
                }
            } else {
                using (BuyerContext db = new BuyerContext())
                {
                    if (!(db.Buyers.Any(x => x.Email == email))) return false;
                }
            }

            return (email != "" && email.Contains("@") && email.Contains("."));
        }

        public static bool date_check(string date)
        {
            string[] dates = date.Split(".");
            if (dates.Length == 3)
            {
                if (int.TryParse(dates[0], out int day) &&
                    int.TryParse(dates[1], out int month) &&
                    int.TryParse(dates[2], out int year)
                ) {
                    int[] months31 = { 1, 3, 5, 7, 8, 10, 12 };
                    int[] months30 = { 4, 6, 9, 11 };
                    int feb = 2;

                    if (months30.Contains(month))
                    {
                        if (!(day > 0 && day <= 30)) return false;
                    }
                    else if (months31.Contains(month))
                    {
                        if (!(day > 0 && day <= 31)) return false;
                    }
                }
            }
        }
    }
}

```

```

        if (month == feb) {
            if (year % 4 == 0 || year % 400 == 0)
            { // если год високосный
                if (!(day > 0 && day <= 29)) return false;
            } else if (!(day > 0 && day <= 28)) return false;
        }

        if (!(year <= DateTime.Now.Year && year > 0 &&
            month > 0 && month <= 12)) return false;

        return true;
    } else return false;
} else return false;
}

public static DateOnly to_date(string date)
{
    string[] nbrs = date.Split('.');
    int.TryParse(nbrs[2], out int year);
    int.TryParse(nbrs[1], out int month);
    int.TryParse(nbrs[0], out int day);

    return new DateOnly(year, month, day);
}
}
}

```

RegForm.cs

```

using System.Text;
using System.Security.Cryptography;

namespace WinFormsAppl
{
    public partial class RegForm : Form
    {
        public RegForm()
        {
            InitializeComponent();
        }

        private string GetHashString(string s)
        {
            byte[] bytes = Encoding.Unicode.GetBytes(s);
            MD5CryptoServiceProvider CSP = new MD5CryptoServiceProvider();
            byte[] byteHash = CSP.ComputeHash(bytes);
            string hash = "";
            foreach (byte b in byteHash)
            {
                hash += string.Format("{0:x2}", b);
            }
        }
    }
}

```

```

    }
    return hash;
}

private void linkLabel1_LinkClicked(object sender, EventArgs e)
{
    LoginForm lf = new LoginForm();
    lf.Show();
    this.Visible = false;
    return;
}

private void button1_Click(object sender, EventArgs e)
{
    string name = this.NameTextBox.Text;
    string surname = this.SurnameTextBox.Text;
    string bdate = this.BDateTextBox.Text;
    string adress = this.AdressTextBox.Text;
    string phone_number = this.PhoneNumberTextBox.Text;
    string first_buy_date = this.FirstBuyDateTextBox.Text;
    string password = this.PasswordTextBox.Text;
    string email = this.EmailTextBox.Text;

    if (!BuyerContext.email_check(email)) {
        MessageBox.Show("Îøèáêà! Íâââðíúé ôíðìàò ìì÷òú");
        return;
    } else {
        if (!BuyerContext.date_check(first_buy_date)) {
            MessageBox.Show("Îøèáêà! Íâââðíúé ôíðìàò ààòú ìâðâîé ìîéóîèè");
            return;
        } else {
            if (!BuyerContext.date_check(bdate)) {
                MessageBox.Show("Îøèáêà! Íâââðíúé ôíðìàò ààòú ðîæââîèý");
                return;
            }
        }
    }

    if (name != "" && name != " " &&
        surname != "" && surname != " " &&
        bdate != "" && bdate != " " &&
        adress != "" && adress != " " &&
        phone_number != "" && phone_number != " " &&
        phone_number.Length == 11 && long.TryParse(phone_number, out long _) &&
        first_buy_date != "" && first_buy_date != " " &&
        password != "" && password != " " && password.Length >= 8 || true)
    {

        using (BuyerContext db = new BuyerContext())

```

```

        {
            DateOnly bd_date = BuyerContext.to_date(bdate);
            DateOnly fbd_date = BuyerContext.to_date(first_buy_date);

            Buyer buyer = new Buyer(
                Name: name,
                Surname: surname,
                BDate: bd_date,
                PhoneNumber: phone_number,
                Adress: adress,
                FirstBuyDate: fbd_date,
                Password: GetHashString(password),
                Email: email
            );

            db.Buyers.Add(buyer);
            db.SaveChanges();

            BuyerForm bf = new BuyerForm();
            bf.Show();
            bf.setBuyerData(buyer);
        }
    } else MessageBox.Show("İşlemleriniz tamamlanmıştır! +
        "Özellikleriniz kaydedildi, tekrar giriş yapabilirsiniz. 11 Ocak 2023");
    }
}
}

```

Buyer.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WinFormsApp1
{
    public class Buyer
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Surname { get; set; }
        public string BDate { get; set; }
        public string PhoneNumber { get; set; }
        public string Adress { get; set; }
        public string FirstBuyDate { get; set; }
        public string Email { get; set; }
        public string Password { get; set; }

        public Buyer()
    }
}

```



```

    {

    }

    public Buyer(string Name, string Surname, string Email, string Password,
        DateOnly BDate, string PhoneNumber, string Adress, DateOnly FirstBuyDate)
    {
        this.Name = Name;
        this.Surname = Surname;
        this.Email = Email;
        this.Password = Password;
        this.BDate = BDate.ToString();
        this.PhoneNumber = PhoneNumber;
        this.Adress = Adress;
        this.FirstBuyDate = FirstBuyDate.ToString();
    }
}

```

LoginForm.cs

```

using System.Text;
using System.Security.Cryptography;
using Microsoft.VisualBasic.ApplicationServices;
using System.Net.Mail;
using System.Net;

namespace WinFormsApp1
{
    public partial class LoginForm : Form
    {
        public LoginForm()
        {
            InitializeComponent();
        }

        private string GetHashString(string s)
        {
            byte[] bytes = Encoding.Unicode.GetBytes(s);
            MD5CryptoServiceProvider CSP = new MD5CryptoServiceProvider();
            byte[] byteHash = CSP.ComputeHash(bytes);
            string hash = "";
            foreach (byte b in byteHash)
            {
                hash += string.Format("{0:x2}", b);
            }
            return hash;
        }

        private void button1_Click(object sender, EventArgs e)
        {

```

```

        string email = this.EmailTextBox.Text;
        string psswd = this.PasswordTextBox.Text;

        if (!(BuyerContext.email_check(email, false) && psswd != "" && psswd != "
    "))
    {
        MessageBox.Show("Ошибка! Неверный формат пароля или email");
        return;
    }

    using (BuyerContext db = new BuyerContext())
    {
        string hash = GetHashString(psswd);
        var buyer = db.Buyers.FirstOrDefault
            (x => x.Email == email && x.Password == hash);

        if (buyer != null)
        {
            MessageBox.Show("Вход успешен!");
            BuyerForm bf = new BuyerForm();
            bf.setBuyerData(buyer);
            bf.Show();
        }
        else MessageBox.Show("Логин или пароль указан неверно!");
    }

}

private void CreateAccaunt_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    RegForm rf = new RegForm();
    rf.Show();
    return;
}

private void ForgotPassword_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    string email = this.EmailTextBox.Text;
    if (BuyerContext.email_check(email, false))
    {
        using (BuyerContext db = new BuyerContext())
        {
            Buyer buyer = db.Buyers.FirstOrDefault(x => x.Email == email);
            if (buyer != null)
            {
                MailAddress from = new MailAddress("busovrm4@mail.ru", "Влад");
                MailAddress to = new MailAddress(email);
                MailMessage m = new MailMessage(from, to);
            }
        }
    }
}

```

Journal of Management Inquiry 22(1)

```
        InitializeComponent();
    }

    public void setBuyerData(Buyer buyer)
    {
        this.NameLabel.Text += $" {buyer.Name}";
        this.SurnameLabel.Text += $" {buyer.Surname}";
        this.BDateLabel.Text += $" {buyer.BDate}";
        this.AdressLabel.Text += $" {buyer.Adress}";
        this.FirstBuyDateLabel.Text += $" {buyer.FirstBuyDate}";
        this.EmailLabel.Text += $" {buyer.Email}";
        this.PhoneNumberLabel.Text += $" {buyer.PhoneNumber}";
    }
}
}
```