Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

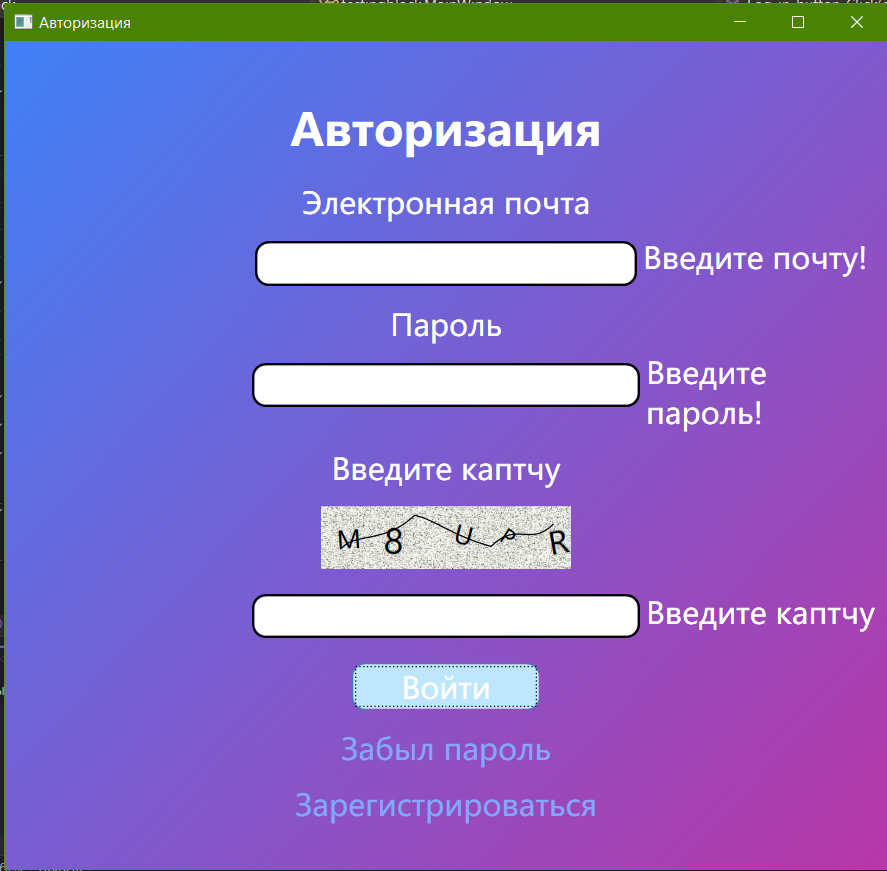
Тема проекта «Магазин авторских коктейлей»

Практическая работа №2 «Разработка тестовых сценариев для аутентификации»

МДК.01.02 Поддержка и тестирование программных модулей

|  |  |
| --- | --- |
|  | Преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л.Р. Набиева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|  | Студент гр. 22П-3  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Ф. Каскинбаев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

2024



Кнопка «Войти»:

private void Log\_in\_button\_Click(object sender, RoutedEventArgs e)

{

if (Check\_min\_length())

{

return;

}

if (Check\_low\_length())

{

Update\_capthca();

return;

}

if(Check\_emoji())

{

Update\_capthca();

return;

}

if (Check\_capthca())

{

Update\_capthca();

return;

}

List<User> users = Read\_Json();

User foundemail = users.FirstOrDefault(user => user.Email == email\_input.Text);

if(foundemail != null)

{

if(foundemail.Password == password\_input.Text)

{

if(IsBlocked(foundemail, users))

{

return;

}

if(F2A(foundemail))

{

return;

}

Empty\_window empty\_Window = new Empty\_window(foundemail);

empty\_Window.Show();

this.Close();

}

else

{

email\_error.Text = "Неверная почта|пароль";

email\_error.Opacity = 100;

if (IsBlocked(foundemail, users))

{

return;

}

Block(foundemail, users);

Update\_capthca();

}

}

else

{

email\_error.Text = "Неверная почта|пароль";

email\_error.Opacity = 100;

Update\_capthca();

}

Код каптчи:

private void Update\_capthca()

{

MyCaptcha.CreateCaptcha(Captcha.LetterOption.Alphanumeric, 5);

var answer = MyCaptcha.CaptchaText;

}

Код ссылок:

private void Replace\_password\_link\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

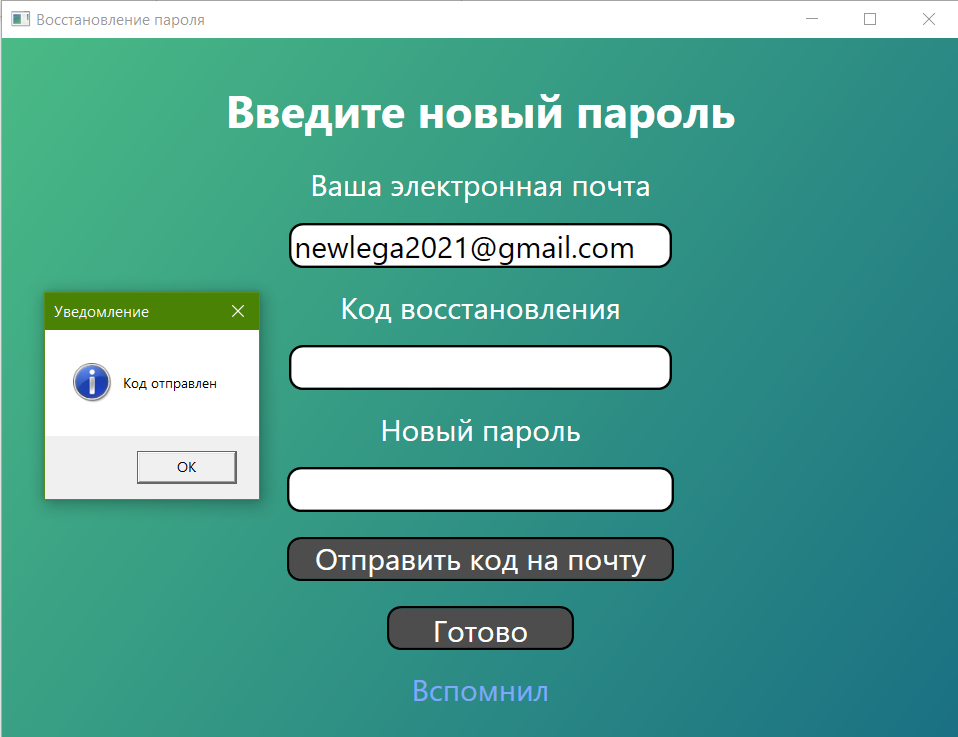
{

Renew\_password renew\_Password = new Renew\_password();

renew\_Password.Show();

this.Close();

}



Код кнопки «Отправить код на почту»

private void Recieve\_code\_Click(object sender, RoutedEventArgs e)

{

if (email\_input.Text.Length == 0)

{

email\_error.Text = "Введите почту!";

email\_error.Opacity = 100;

return;

}

else

{

email\_error.Text = "";

email\_error.Opacity = 0;

}

if(Check\_email\_format(email\_input.Text) || email\_error.Opacity != 0)

{

return;

}

List<User> users = Read\_Json();

User foundemail = users.FirstOrDefault(user => user.Email == email\_input.Text);

if (foundemail == null)

{

password\_input.Text = "";

code\_input.Text = "";

email\_error.Text = "Почта не зарегистрирована!";

code\_error.Text = "";

password\_error.Text = "";

email\_error.Opacity = 100;

return;

}

EmailService email = new EmailService();

email.SendEmail(email\_input.Text, "Восстановление пароля", GenCode());

}

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Код кнопки «Готово»

private void Log\_in\_button\_Click(object sender, RoutedEventArgs e)

{

if (Check\_min\_length())

{

return;

}

if (Check\_low\_length())

{

return;

}

if (Check\_email\_format(email\_input.Text))

{

return;

}

List<User> users = Read\_Json();

User foundemail = users.FirstOrDefault(user => user.Email == email\_input.Text);

if (foundemail == null)

{

password\_input.Text = "";

code\_input.Text = "";

email\_error.Text = "Почта не зарегистрирована!";

email\_error.Opacity = 100;

return;

}

if(Check\_code())

{

return;

}

if(Is\_dubling(foundemail))

{

return;

}

foundemail.Password = password\_input.Text;

File.WriteAllText("Users.json", JsonSerializer.Serialize(users));

MessageBox.Show("Пароль успешно изменен!");

current\_code = "";

email\_input.Text = "";

code\_input.Text = "";

password\_input.Text = "";

}

**Код ссылки идентичен коду ссылок на других окнах**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Кнопка «Зарегистрироваться»

private void Reg\_button\_Click(object sender, RoutedEventArgs e)

{

if (Check\_input())

{

return;

}

if(Check\_emoji())

{

return;

}

if (!new Regex(@"^[-\w.]+@([A-Za-z0-9][-A-Za-z0-9]+**\.**)+[A-Za-z]{2,4}$").IsMatch(email\_input.Text))

{

email\_error.Text = "Некорректная почта!";

email\_error.Opacity = 100;

return;

}

User user = new User(Create\_ID(), surname\_input.Text, name\_input.Text, lastname\_input.Text, date\_input.SelectedDate.Value, email\_input.Text, password\_input.Text, Convert.ToDateTime("00:00:00"), false);

List<User> users = Read\_Json();

foreach (User current\_user in users)

{

if (current\_user.Email == email\_input.Text)

{

email\_error.Text = "Такая почта уже зарегистрирована!";

email\_error.Opacity = 100;

return;

}

}

users.Add(user);

string data = JsonSerializer.Serialize(users);

File.WriteAllText("Users.json", data);

MessageBox.Show("Вы успешно зарегистрировались!");

surname\_input.Text = "";

name\_input.Text = "";

lastname\_input.Text = "";

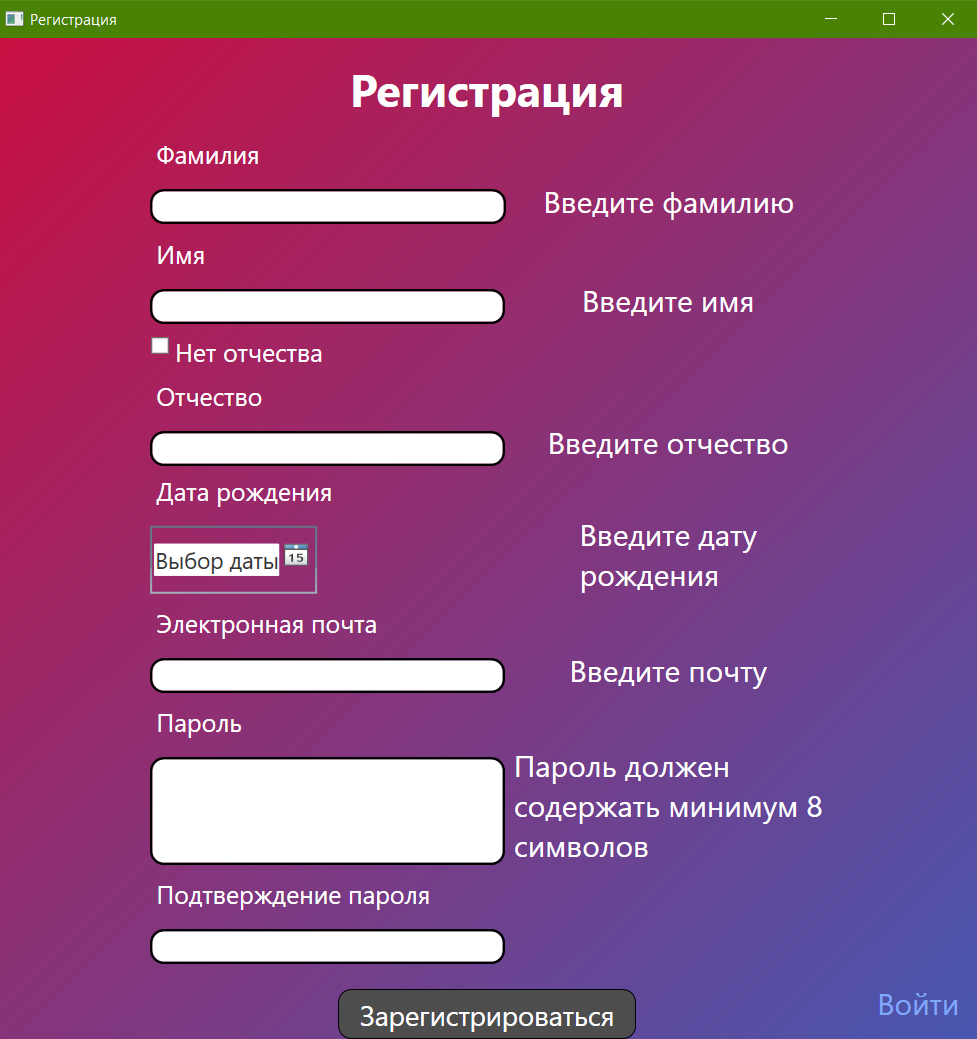
date\_input.Text = "";

email\_input.Text = "";

password\_input.Text = "";

equal\_password\_input.Text = "";

}



Изображение выглядит как текст, снимок экрана, Шрифт, Операционная система

Автоматически созданное описание

Чек/анчек бокса:

private void F2A\_check\_Checked(object sender, RoutedEventArgs e)

{

List<User> users = new MainWindow().Read\_Json();

User user = users.FirstOrDefault(us => us.Email == User.Email);

if (user != null)

{

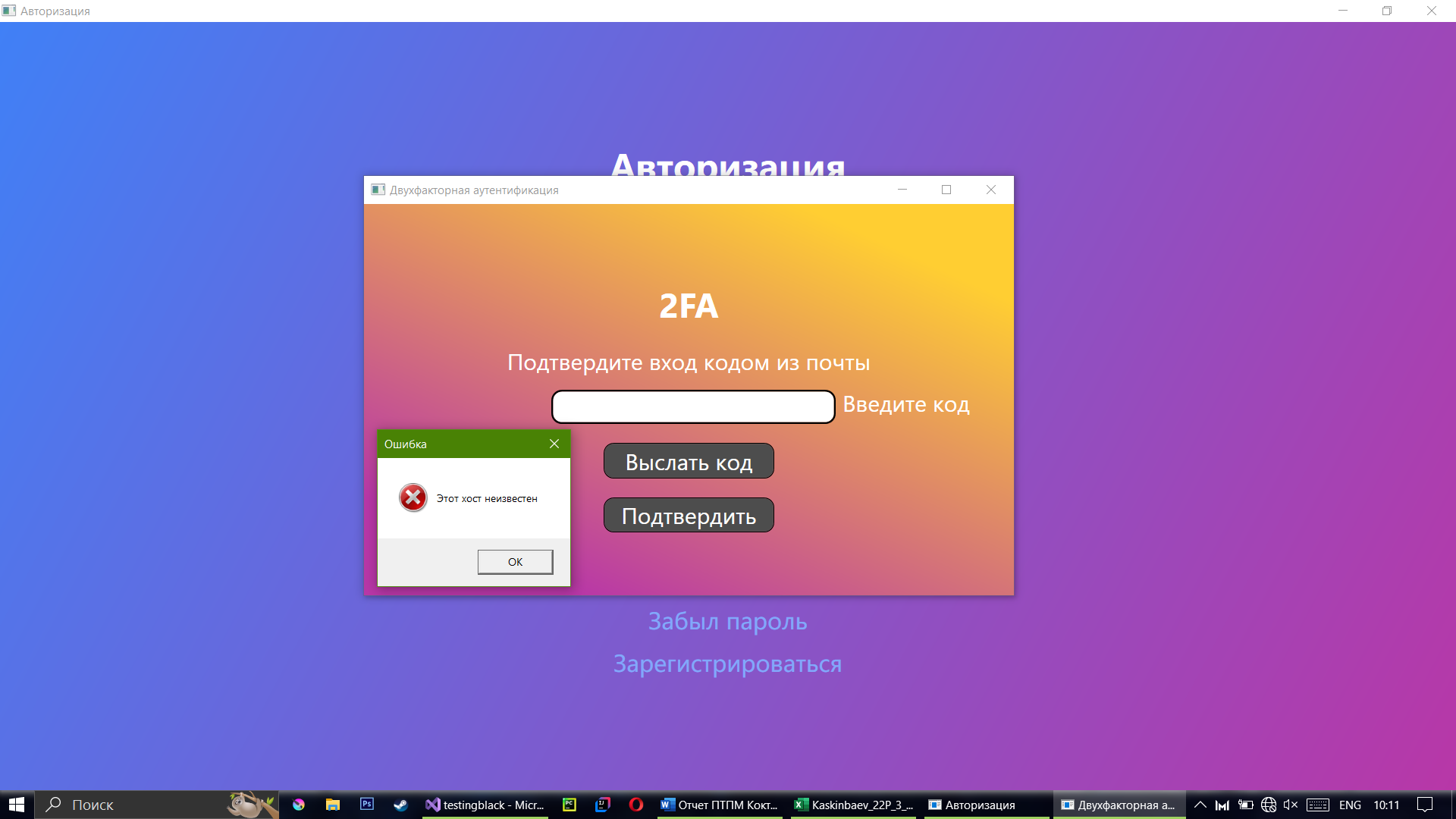
user.F2A = true;

File.WriteAllText("Users.json", JsonSerializer.Serialize(users));

}

}

в uncheck : user.F2A = false;



Код окна:

public partial class F2A\_window : Window

{

string Email { get; set; }

private string current\_code = "";

public F2A\_window(string email)

{

InitializeComponent();

Email = email;

}

private void Accept\_button\_Click(object sender, RoutedEventArgs e)

{

if(code\_input.Text.Length == 0)

{

code\_error.Text = "Введите код";

code\_error.Opacity = 100;

return;

}

if (current\_code == code\_input.Text && code\_input.Text.Length>0)

{

this.DialogResult = true;

this.Close();

}

else

{

code\_error.Text = "Неверный код";

code\_error.Opacity = 100;

}

}

private void Send\_code\_Click(object sender, RoutedEventArgs e)

{

Send\_email();

}

private void Send\_email()

{

EmailService email = new EmailService();

email.SendEmail(Email, "Код подтверждения", GenCode(6));

}

private string GenCode(int len)

{

Random random = new Random();

StringBuilder stringBuilder = new StringBuilder();

string chars = "qwertyuiopQWERTYUIOP0";

for (int i = 0; i < len; i++)

{

stringBuilder.Append(chars[random.Next(chars.Length)]);

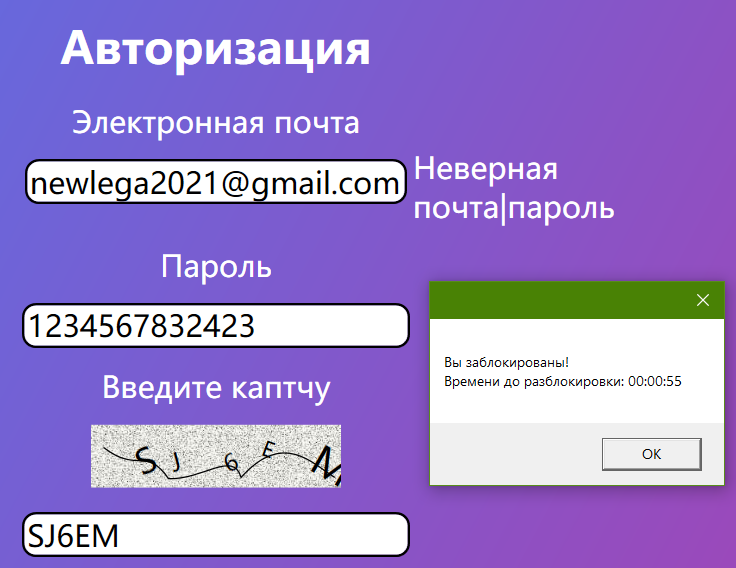
}

current\_code = stringBuilder.ToString();

return stringBuilder.ToString();

}

}

Блокировка аккаунта:  


Коды методов используемых для блокировки:

private bool IsBlocked(User user, List<User> users)

{

TimeSpan block\_time\_remaining = user.Time\_block - DateTime.Now;

if (block\_time\_remaining < TimeSpan.Zero)

{

user.Time\_block = Convert.ToDateTime("00:00:00");

File.WriteAllText("Users.json", JsonSerializer.Serialize(users));

}

if (user.Time\_block != Convert.ToDateTime("00:00:00"))

{

MessageBox.Show($"Вы заблокированы!\nВремени до разблокировки: {block\_time\_remaining.Hours:D2}:{block\_time\_remaining.Minutes:D2}:{block\_time\_remaining.Seconds:D2}");

count\_of\_uncorrect = 0;

return true;

}

return false;

}

private void Block(User user, List<User> users)

{

if (user.Time\_block.ToLongTimeString().Contains("0:00:00"))

{

count\_of\_uncorrect++;

}

if (count\_of\_uncorrect == 5)

{

user.Time\_block = DateTime.Now.AddSeconds(55);

File.WriteAllText("Users.json", JsonSerializer.Serialize(users));

IsBlocked(user, users);

}

}