Учреждение образования «Гомельский торгово-экономический колледж» Белкоопсоюза

**База данных фитнес клуба**

ТЕКСТ ПРОГРАММЫ

КП.09551.12

Листов 10

Разработал Д.А. Клименко

Руководитель проекта О.Л. Иткина

//Получение данных из базы данных, используя классы

internal class AppCon : DbContext {

public DbSet<User> Users { get; set; }

public DbSet<Tovar> Tovars { get; set; }

public AppCon() : base("DefaultConnection") { }

}

//Кнопка скрытия программы

private void Image\_MouseDown(object sender, MouseButtonEventArgs e)

{ WindowState = WindowState.Minimized; }

//Кнопка выхода из программы

private void Image\_MouseDown\_1(object sender, MouseButtonEventArgs e)

{ this.Close(); }

//Кнопка входа

try

{

string logq = Login.Text;

string pass = password.Text;

MySqlCommand command = new MySqlCommand("SELECT \* FROM `users` WHERE `login` = @login AND `password`= @passw", mySQL.GetConnection());

command.Parameters.Add("@login", MySqlDbType.VarChar).Value = logq;

command.Parameters.Add("@passw", MySqlDbType.VarChar).Value = pass;

adapter.SelectCommand = command;

adapter.Fill(table);

GetSet.log = logq;

if (table.Rows.Count > 0)

{

MainApp main = new MainApp();

main.Show();

}

else

{

qwe.Text = "Пароль или логин неверный";

qwe.Visibility = Visibility.Visible;

}

}

catch (Exception)

{

qwe.Text = "Проверьте соединение ";

qwe.Visibility = Visibility.Visible;

qwe.Foreground = (Brush)bc.ConvertFrom("#FFF9D652");

}

//Кнопка для регистрации пользователя

string code = Random().ToString();

System.Net.Mail.MailMessage message = new System.Net.Mail.MailMessage();

message.IsBodyHtml = true;

message.From = new MailAddress("admin@company.com", "Моя компания");

message.To.Add(emailAdress);

message.Subject = "Сообщение от администрации";

message.Body = "<div style=\"color: black;\">Код для создание аккаунта + </div>" + code;

using (System.Net.Mail.SmtpClient client = new System.Net.Mail.SmtpClient("smtp.gmail.com"))

{

client.Credentials = new NetworkCredential("tixanuiop@gmail.com", "tixan159753fF");

client.Port = 587;

client.EnableSsl = true;

client.Send(message);

//Кнопка подтверждения кода

private void TextBlock\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (Login.Text != "")

{

if (Login.Text == GetSet.Code.ToString())

{

DataBase();

}

else

{

count--;

txtCount.Text = count.ToString();

if (count == 0)

{

txt.Visibility = Visibility.Hidden;

txtCount.Visibility = Visibility.Hidden;

textEmail.Text = "Вы ввели код подтверждения неправильно 3 раза. В следующий раз вы сможете ввести код через 30 мин";

textEmail.FontSize = 10;

}

}

}

else

{

textEmail.Text = "Поле не должно быть пустым";

}

//Click

} //Кнопка перехода на страницу входа

private void Window\_Activated(object sender, EventArgs e)

{

AutomaticForm.Content = new automatic();

} //Главная форма

//Кнопка просмотра новостей

private void AnimNews\_MouseDown(object sender, MouseButtonEventArgs e)

{

FrameNews.Content = new newss();

FrameNews.Visibility = Visibility.Visible;

FramePrice.Visibility = Visibility.Hidden;

FrameGym.Visibility = Visibility.Hidden;

} //Кнопка просмотра услуг

private void AnimPrice\_MouseDown(object sender, MouseButtonEventArgs e)

{

FramePrice.Content = new Price();

FrameNews.Visibility = Visibility.Hidden;

FramePrice.Visibility = Visibility.Visible;

FrameGym.Visibility = Visibility.Hidden;

} //Кнопка для выхода из программы

private void Button\_Click\_Exit(object sender, RoutedEventArgs e) {

if (MessageBox.Show("Вы действительно хотите выйти из аккаунта?",

"Подтвердите действие", MessageBoxButton.YesNo) == MessageBoxResult.Yes) {

AuthWin auth = new AuthWin(); auth.Show(); Close(); } }

//Событие при открытии главной формы

private void Window\_Activated(object sender, EventArgs e) { User authuser = null;

using (AppCon context = new AppCon()) {

authuser = context.Users.Where(b => b.id == id).FirstOrDefault(); } if (authuser.access == "1") {

Admin.Visibility = Visibility.Hidden; Director.Visibility = Visibility.Hidden; }

else if (authuser.access == "2") { Director.Visibility = Visibility.Hidden; } }

//Вывод новостей

public partial class newss : Page, INotifyPropertyChanged

{

public event PropertyChangedEventHandler PropertyChanged;

private DataTable \_items; // не используйте \_items в других местах кода

//DataTable

public DataTable Items // вместо этого используйте именно Items

{

get => \_items;

set

{

DataContext = this;

\_items = value;

//MessageBox.Show(nameof(Items));

PropertyChanged?.DynamicInvoke(new PropertyChangedEventArgs(nameof(Items)));

//DynamicInvoke

}

}

private void LoadData()

{

MySqlConnection conn = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=gymstudio");

conn.Open();

string cmd = "SELECT \* FROM news"; // Из какой таблицы нужен вывод

MySqlCommand createCommand = new MySqlCommand(cmd, conn);

createCommand.ExecuteNonQuery();

MySqlDataAdapter dataReader = new MySqlDataAdapter(createCommand);

DataTable dt = new DataTable("news"); // В скобках указываем название таблицы

dataReader.Fill(dt);

Items = dt; // Сам вывод

conn.Close();

}

public newss()

{

InitializeComponent();

DataContext = this;

LoadData();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

//ЕСЛИ НАЖАЛ НА СЕРДЕЧКО

}

}

//Кнопка изменение аватара

try

{

string connStr = "server=localhost;port=3306;username=root;password=root;database=gymstudio";

MySqlConnection conn = new MySqlConnection(connStr);

conn.Open();

string sqlPhoto = "SELECT PathAvatar FROM users WHERE login = '" + GetSet.log + "'";

MySqlCommand commandPhoto = new MySqlCommand(sqlPhoto, conn);

string PhotoAva = commandPhoto.ExecuteScalar().ToString();

AvatarSett.ImageSource = new BitmapImage(new Uri(PhotoAva));

conn.Close();

}

catch (NullReferenceException)

{

MessageBox.Show("Ваш аккаунт был удален!!!");

}

private void Button\_Click\_EditUser(object sender, RoutedEventArgs e) {

string idus = ""; int idu = 0;

if (us.usersGrid.SelectedItem != null) {

pages.editUser edu = new pages.editUser();

idus = us.usersGrid.SelectedItem.ToString(); User user = null;

idu = int.Parse(idus); edu.EditUser.Text = "Редактирование личного кабинета #" + idu;

using (db = new AppCon()) {

user = db.Users.Where(b => b.id == idu).FirstOrDefault();

edu.tbLog.Text = user.login; edu.tbEmail.Text = user.email;

edu.tbPass.Text = user.pass; edu.NumbAc.Text = idus;

edu.AccessList.Items.Add("Пользователь");

edu.AccessList.Items.Add("Администратор");

edu.AccessList.Items.Add("Директор");

edu.AccessList.SelectedIndex = 0;

if (user.access == "1") { edu.AccessList.SelectedIndex = 0; }

else if (user.access == "2") { edu.AccessList.SelectedIndex = 1; }

else if (user.access == "3") { edu.AccessList.SelectedIndex = 2; }

edu.StatusList.Items.Add("Активен");

edu.StatusList.Items.Add("Заблокирован");

if (user.status == "1") { edu.StatusList.SelectedIndex = 0;

} else if (user.status == "2") { edu.StatusList.SelectedIndex = 1; } }

edu.DataUser = user.id; Win.Content = edu; }

else MessageBox.Show("Вы не выбрали пользователя для редактирования",

"Сообщение", MessageBoxButton.OK); }

//Кнопка добавления новостей

private void Border\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (Photo != null)

{

String newsaddtext = "Статья добавлена";

byte[] photo = GetPhoto(GetSet.photo);

MySqlCommand command = new MySqlCommand("INSERT INTO `news` (`img`, `mtxt`,`txt`) VALUES (@photo, @mtext, @text)", MySQL.GetConnection());

command.Parameters.Add("@photo", MySqlDbType.Blob).Value = photo;

command.Parameters.Add("@mtext", MySqlDbType.Text).Value = nametxt.Text;

command.Parameters.Add("@text", MySqlDbType.Text).Value = txt.Text;

MySQL.openConBd();

command.ExecuteNonQuery();

newsAddText.Text = newsaddtext;

}

else

{

MessageBox.Show("Добавьте фото");

}

} //Кнопка записи

private void Button\_Click(object sender, RoutedEventArgs e)

{

MessageBox.Show("Выбранное вами время свободно");

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

MessageBox.Show("Выбранное вами время свободно");

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

MessageBox.Show("Выбранное вами время свободно");

}

}

private void Button\_Click\_Clear(object sender, RoutedEventArgs e) { clear();

MessageBox.Show("Вы успешно оичстили данные", "Сообщение", MessageBoxButton.OK); }

//Метод обновления таблицы

private void UpdateTable() {

using (db = new AppCon()) { phonesGrid.ItemsSource = db.Tovars.ToList<Tovar>(); }

MessageBox.Show("Вы обновили таблицу", "Сообщение", MessageBoxButton.OK); }

//Кнопка добавление аватара

private void Border\_MouseDown\_1(object sender, MouseButtonEventArgs e)

{

//Добавить фото

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.BMP;\*.JPG;\*.PNG)|\*.BMP;\*.JPG;\*.PNG|All files(\*.\*)|\*.\*";

if (open.ShowDialog() == true)

{

try

{

MySQL my = new MySQL();

Photo = open.FileName;

Uri uri = new Uri(Photo);

AvatarSett.ImageSource = new BitmapImage(new Uri(Photo));

GetSet.photoAvar = uri.ToString();

MySqlCommand command = new MySqlCommand("UPDATE users SET PathAvatar = '" + GetSet.photoAvar + "' WHERE login = '" + GetSet.log + "' ", my.GetConnection());

my.openConBd();

command.ExecuteNonQuery();

my.CloseConBd();

}

catch (Exception)

{

throw;

}

}

} //Кнопка выхода из аккаунта

private void Border\_MouseDown(object sender, MouseButtonEventArgs e)

{

this.Close();

MainWindow main = new MainWindow();

//Метод конвертации фото

private static byte[] GetPhoto(string filePath)

{

FileStream stream = new FileStream(filePath, FileMode.Open, FileAccess.Read);

BinaryReader reader = new BinaryReader(stream);

byte[] photo = reader.ReadBytes((int)stream.Length);

reader.Close();

stream.Close();

return photo;

}

//Метод анимированной ошибки

private void animError()

{

TranslateTransform translateTransform = new TranslateTransform();

loginError.RenderTransform = translateTransform;

mailError.RenderTransform = translateTransform;

Pass.RenderTransform = translateTransform;

PassRepead.RenderTransform = translateTransform;

//1

DoubleAnimation animationX = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(5));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationX);

DoubleAnimation animationX2 = new DoubleAnimation(-4, -10, TimeSpan.FromSeconds(5));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationX2);

DoubleAnimation animationY = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(5));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationY);

DoubleAnimation animationY2 = new DoubleAnimation(-4, 0, TimeSpan.FromSeconds(5));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationY2);

//2

DoubleAnimation animationXTwo = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationXTwo);

DoubleAnimation animationX2Two = new DoubleAnimation(-4, 0, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationX2Two);

DoubleAnimation animationYTwo = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationYTwo);

DoubleAnimation animationY2Two = new DoubleAnimation(-4, 0, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationY2Two);

//3

DoubleAnimation animationXThree = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationXThree);

DoubleAnimation animationX2Three = new DoubleAnimation(-4, 0, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.XProperty, animationX2Three);

DoubleAnimation animationYThree = new DoubleAnimation(0, 4, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationYThree);

DoubleAnimation animationY2Three = new DoubleAnimation(-4, 0, TimeSpan.FromSeconds(0.1));

translateTransform.BeginAnimation(TranslateTransform.YProperty, animationY2Three);

}

//Метод рандомных чисел для подтверждения

private int Random()

{

Random rnd = new Random();

int value = rnd.Next(1000, 9999);

GetSet.Code = value;

return value;

}

//Методы сравнения введённых данных

private bool ComparePasswords(string one, string two) {

bool check = false;

string pattPass = @"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[0-9])(?=.\*[^a-zA-Z0-9])\S{1,16}$";

if (one == two && Regex.IsMatch(one, pattPass) && Regex.IsMatch(two, pattPass))

{

check = true;

}

else

{

check = false;

}

return check;

}

private bool CompareMail(string Mail)

{

bool check = false;

string pattEmail = @"(\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*)";

if (Regex.IsMatch(Mail, pattEmail))

{

check = true;

}

else

{

check = false;

}

return check;

}

private bool CompareLogin(string login)

{

bool check = false;

string pattLog = @"^[\w-[\d\_]][\w-[\_]]{1,9}$";

if (Regex.IsMatch(login, pattLog))

{

check = true;

}

else

{

check = false;

}

return check;

}

private void InputErrorOrEntry(bool pass, bool mail, bool log)

{

bool one = false;

bool two = false;

bool three = false;

if (pass == false)

{

Pass.Visibility = Visibility.Visible;

PassRepead.Visibility = Visibility.Visible;

}

else { one = true; }

if (mail == false)

{

mailError.Visibility = Visibility.Visible;

}

else { two = true; }

if (log == false)

{

loginError.Visibility = Visibility.Visible;

}

else { three = true; }

if (one == true)

{

Pass.Visibility = Visibility.Hidden;

PassRepead.Visibility = Visibility.Hidden;

}

if (two == true)

{

mailError.Visibility = Visibility.Hidden;

}

if (three == true)

{

loginError.Visibility = Visibility.Hidden;

}

if (one == true && two == true && three == true )

{

MainRegView.Visibility = Visibility.Hidden;

Loading.Visibility = Visibility.Visible;

SendMessage(\_mail);

Code code = new Code();

code.Show();

GetSet.log = \_login;

GetSet.pass = \_password;

GetSet.email = \_mail;

}

}