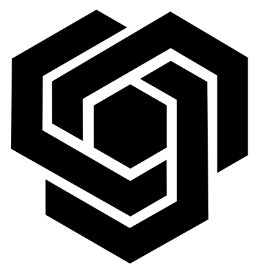
**Технически Университет София**

Факултет Компютърни системи и технологии

**Курсов Проект**

**По Програмиране в разпределени среди**

**Приложение за прилагане на ефекти върху изображения**

Изготвил:

Мила Иванова Иванова

Специалност: КСИ

Курс: Четвърти

Група: 42

Фак. № 121214080

София, 2018

СЪДЪРЖАНИЕ

Увод.......................................................................................................................................... 3

Проектиране............................................................................................................................ 3

Реализация............................................................................................................................... 4

Заключение............................................................................................................................. 6

Приложение............................................................................................................................ 7

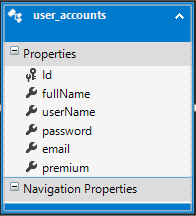
УВОД

В технологичното време, в което живеем, все повече хора желаят да се докоснат до обработката на изображения. Известни са много програми, които променят изображението с желания от нас ефект като правят снимките да изглеждат доста въздействащи.

Този проект представлява сайт за обработване на изображения с определени ефекти, като процесът е лесен и удобен за ползване от потребителя.

ПРОЕКТИРАНЕ

Базата данни е малка и има следния изглед:



Сайтът и функционалностите му ще се ползват от всеки човек, който има направена регистрация и желае да сложи ефекти на снимките си.

При стартиране на приложението, потребителят достъпва страницата за вписване в профила му. На страницата има бутон за вписване. Потребителят трябва да въведе своето потребителско име и парола. При коректно попълнени данни от потребителя, страницата пренасочва към другата страница за слагане на ефектите.

На нея има бутон за избиране на файл от компютъра на потребителя. След като потребителят е избрал снимка трябва да натисне бутона, за да може снимката да се визуализира на екрана. Под генерираното изображение има бутони с ефекти – първият се нарича Magic 1, вторият – Magic 2, третият – Magic 3. При натискането им изображението се променя със съответния ефект.

РЕАЛИЗАЦИЯ

Проектът има 5 папки – App\_Data, View, Model, Code, Images.

App\_Data – в тази папка се намира базата данни, където е таблицата със записите. user\_accounts.mdf е файлът.

View – в тази папка се намират двете страници на приложението – LogInPage.aspx Editor.aspx

Model – в тази папка се намират моделът на базата данни (Entity Model), за потребителите и за изображението: AccountsInfo.cs ImageDetails.cs UserAccountsEntityModel.edmx

Code – в тази папка се намират уеб сървисите във файл MagicServies.asmx и Helper.cs клас, който прави проверка на входящите данни при влизане в профил.

Images – в тази папка се запазват изображенията, които потребителят качва за обработка. В тази папка има папка bm, където се запазват вече обработените изображения.

Приложението използва бази данни sql, Entity Framework, Soap.

Необходим е html файл, койо да укаже дали заявките са POST и какви ще са входните данни.

Приложението стартира от LogInPage.aspx. На тази страница се появява формата за вписване, която съдържа бутон за вписване и полета за въвеждане на потребителско име

и парола.

<form id="form1" runat="server">

<fieldset>

Username or Email:

<p><asp:TextBox ID="tbUsername" runat="server"></asp:TextBox></p>

Password:

<p><asp:TextBox ID="tbPassword" runat="server" TextMode="Password"></asp:TextBox></p>

<p><a href="#">Forgot Password?</a></p>

<p><asp:Button ID="btnLogIn" runat="server" Text="Log In" OnClick="btnLogIn\_Click" /></p>

</fieldset>

</form>

Прави се проверка дали съответният потребител съществува и дали въведените от него данни са коректни. Ако се намери съвпадение, се опваря втората страница.

var user = users.user\_accounts.FirstOrDefault(u => u.userName == username);

if (user != null)

{

if (user.password == password)

{

if (user.premium == 0)

{

Server.Transfer("Editor.aspx");

}

}

else

{

lblPass.Text = "Incorrect login credentials";

}

}

На втората страница Editor.aspx се визуализира FileUploader контрола, чрез която потребителят избира изображение от компютъра си.

<form id="form1" runat="server">

<div>

<asp:FileUpload ID="fileUploader" runat="server" />

<asp:Button ID="btnUpload" runat="server" Text="Upload" OnClick="btnUpload\_Click" />

<br />

<asp:Image ID="imageBox" runat="server" />

<br />

<fieldset>

<asp:Button ID="btnMagic1" runat="server" Text="Magic 1" OnClick="btnMagic1\_Click" />

</fieldset>

</div>

</form>

Бутонът Upload записва изображението в папка в проекта и визуализира в Image контролата изображението.

protected void btnUpload\_Click(object sender, EventArgs e)

{

if (fileUploader.HasFile)

{

string fileName = System.IO.Path.GetFileName(fileUploader.PostedFile.FileName);

fileUploader.PostedFile.SaveAs(Server.MapPath("~/Images/" + fileName));

imageBox.ImageUrl = "~/Images/" + fileName;

Session["fileName"] = fileName;

bm = new Bitmap(Server.MapPath(imageBox.ImageUrl));

Session["bitMap"] = bm;

imgD.ImageWidth = bm.Width;

Session["w"] = imgD.ImageWidth;

imgD.ImageHeight = bm.Height;

Session["h"] = imgD.ImageHeight;

}

else

{

}

}

Бутонът Magic 1 вика уеб сървиса на приложението, който обработва изображението със съответния ефект.

protected void btnMagic1\_Click(object sender, EventArgs e)

{

imgD.ImageWidth = (int)Session["w"];

imgD.ImageHeight = (int)Session["h"];

string fileName = Session["fileName"].ToString();

MagicServicesReference.MagicServicesSoapClient magic = new MagicServicesReference.MagicServicesSoapClient();

string imageUrl = magic.Magic1(imgD.ImageWidth, imgD.ImageHeight, fileName);

imageBox.ImageUrl = imageUrl;

}

По същия начин работят и бутоните Magic 2 и Magic 3 – използват уеб сървисите, които им съответстват и визуализират новото генерирано изображение.

ЗАКЛЮЧЕНИЕ

Приложението изпълнява условието да има уеб сървис. Използван е лесен за употреба интерфейс. Основната функционалност за обработване на изображението с ефект е изпълнена.

Възможните подобрения биха били добавяне на още ефекти.

ПРИЛОЖЕНИЕ

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

</head>

<body>

<form id="1" action='Code/MagicServices.asmx/Magic1' method="post" target="\_blank">

<table>

<tr>

<td>

<input id="fileUploader" type="file" name="" />

</td>

</tr>

<tr>

<td></td>

<td align="right">

<input type="submit" value="Upload" class="button" />

</td>

</tr>

<tr>

<td></td>

<td align="left">

<input type="image" value="img" />

</td>

</tr>

<tr>

<td></td>

<td align="left">

<input type="submit" value="Magic 1" class="button" />

</td>

</tr>

</table>

</form>

<form id="2" action='Code/MagicServices.asmx/Magic2' method="post" target="\_blank">

<table>

<tr>

<td></td>

<td align="left">

<input type="submit" value="Magic 2" class="button" />

</td>

</tr>

</table>

</form>

<form id="3" action='Code/MagicServices.asmx/Magic3' method="post" target="\_blank">

<table>

<tr>

<td></td>

<td align="left">

<input type="submit" value="Magic 2" class="button" />

</td>

</tr>

</table>

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Drawing.Imaging;

using System.Linq;

using System.Web;

using System.Web.Services;

namespace Online\_Photo\_Effects.Code

{

/// <summary>

/// Summary description for MagicServices

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class MagicServices : System.Web.Services.WebService

{

private int matrixN;

private double dist;

private Bitmap inBitMap;

int[,] ditherMatrix;

[WebMethod(EnableSession = true)]

public string Magic1(int w, int h, string fileName)

{

double sum;

Bitmap bm = new Bitmap(Server.MapPath("~/Images/" + fileName));

matrixN = 2; // set the dimension of the matrix (matrix 2 by 2)

dist = 255 / 4.0; // set the step used to determine the four shades of gray

for (int i = 0; i < w; i = i + matrixN)

{

for (int j = 0; j < h; j = j + matrixN)

{

sum = 0;

for (int k = i; k < i + matrixN; k++)

for (int m = j; m < j + matrixN; m++)

{

sum += bm.GetPixel(k, m).R;

}

sum /= 4.0;

if (sum == 255.0)

{

bm.SetPixel(i, j, Color.White);

bm.SetPixel(i, j + 1, Color.White);

bm.SetPixel(i + 1, j, Color.White);

bm.SetPixel(i + 1, j + 1, Color.White);

}

else

{

if (sum >= 3 \* dist && sum < 255)

{

bm.SetPixel(i, j, Color.White);

bm.SetPixel(i, j + 1, Color.White);

bm.SetPixel(i + 1, j, Color.Black);

bm.SetPixel(i + 1, j + 1, Color.White);

}

else

{

if (sum >= 2 \* dist && sum < 3 \* dist)

{

bm.SetPixel(i, j, Color.White);

bm.SetPixel(i, j + 1, Color.Black);

bm.SetPixel(i + 1, j, Color.Black);

bm.SetPixel(i + 1, j + 1, Color.White);

}

else

{

if (sum >= dist && sum < 2 \* dist)

{

bm.SetPixel(i, j, Color.White);

bm.SetPixel(i, j + 1, Color.Black);

bm.SetPixel(i + 1, j, Color.Black);

bm.SetPixel(i + 1, j + 1, Color.Black);

}

if (sum >= 0.0 && sum < dist)

{

bm.SetPixel(i, j, Color.Black);

bm.SetPixel(i, j + 1, Color.Black);

bm.SetPixel(i + 1, j, Color.Black);

bm.SetPixel(i + 1, j + 1, Color.Black);

}

}

}

}

}

bm.Save(Server.MapPath("~/Images/bm/") + fileName + ".jpg", ImageFormat.Jpeg);

}

return "~/Images/bm/" + fileName + ".jpg";

}

[WebMethod(EnableSession = true)]

public string Magic2(int w, int h, string fileName)

{

int k;

int m;

inBitMap = new Bitmap(Server.MapPath("~/Images/" + fileName));

initializeMarixForOrderedDithering();

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

for (int j = 0; j < h; j++)

{

k = i % 2;

m = j % 2;

if (Aproxim(inBitMap.GetPixel(i, j).R) < ditherMatrix[k, m])

{

inBitMap.SetPixel(i, j, Color.White);

}

else

{

inBitMap.SetPixel(i, j, Color.Black);

}

}

inBitMap.Save(Server.MapPath("~/Images/bm/") + fileName + ".jpg", ImageFormat.Jpeg);

return "~/Images/bm/" + fileName + ".jpg";

}

[WebMethod(EnableSession = true)]

public string Magic3(int w, int h, string fileName)

{

int k;

int m;

inBitMap = new Bitmap(Server.MapPath("~/Images/" + fileName));

initializeMarixForOrderedDithering();

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

for (int j = 0; j < h; j++)

{

k = i % 2;

m = j % 2;

if (Aproxim2(inBitMap.GetPixel(i, j).R) < ditherMatrix[k, m])

{

inBitMap.SetPixel(i, j, Color.White);

}

else

{

inBitMap.SetPixel(i, j, Color.Black);

}

}

inBitMap.Save(Server.MapPath("~/Images/bm/") + fileName + ".jpg", ImageFormat.Jpeg);

return "~/Images/bm/" + fileName + ".jpg";

}

private void initializeMarixForOrderedDithering()

{

ditherMatrix = new int[4, 4];

ditherMatrix[0, 0] = 0;

ditherMatrix[0, 1] = 1;

ditherMatrix[0, 2] = 2;

ditherMatrix[0, 3] = 3;

ditherMatrix[1, 0] = 4;

ditherMatrix[1, 1] = 5;

ditherMatrix[1, 2] = 6;

ditherMatrix[1, 3] = 7;

ditherMatrix[2, 0] = 8;

ditherMatrix[2, 1] = 9;

ditherMatrix[2, 2] = 10;

ditherMatrix[2, 3] = 11;

ditherMatrix[3, 0] = 12;

ditherMatrix[3, 1] = 13;

ditherMatrix[3, 2] = 14;

ditherMatrix[3, 3] = 15;

}

private int Aproxim(int intensity)

{

if (intensity > 0 && intensity < 16)

return 15;

else

if (intensity >= 16 && intensity < 32)

return 14;

else

if (intensity >= 32 && intensity < 48)

return 13;

else

if (intensity >= 48 && intensity < 64)

return 12;

else

if (intensity >= 64 && intensity < 80)

return 11;

else

if (intensity >= 80 && intensity < 96)

return 10;

else

if (intensity >= 96 && intensity < 112)

return 9;

else

if (intensity >= 112 && intensity < 128)

return 8;

else

if (intensity >= 128 && intensity < 144)

return 7;

else

if (intensity >= 144 && intensity < 160)

return 6;

else

if (intensity >= 160 && intensity < 176)

return 5;

else

if (intensity >= 176 && intensity < 192)

return 4;

else

if (intensity >= 192 && intensity < 208)

return 3;

else

if (intensity >= 208 && intensity < 224)

return 2;

else

if (intensity >= 224 && intensity < 240)

return 1;

else

if (intensity >= 240 && intensity < 255)

return 0;

else

return 0;

}

private int Aproxim2(int intensity)

{

if (intensity > 0 && intensity < 16)

return 0;

else

if (intensity >= 16 && intensity < 32)

return 1;

else

if (intensity >= 32 && intensity < 48)

return 2;

else

if (intensity >= 48 && intensity < 64)

return 3;

else

if (intensity >= 64 && intensity < 80)

return 4;

else

if (intensity >= 80 && intensity < 96)

return 5;

else

if (intensity >= 96 && intensity < 112)

return 6;

else

if (intensity >= 112 && intensity < 128)

return 7;

else

if (intensity >= 128 && intensity < 144)

return 8;

else

if (intensity >= 144 && intensity < 160)

return 9;

else

if (intensity >= 160 && intensity < 176)

return 10;

else

if (intensity >= 176 && intensity < 192)

return 11;

else

if (intensity >= 192 && intensity < 208)

return 12;

else

if (intensity >= 208 && intensity < 224)

return 13;

else

if (intensity >= 224 && intensity < 240)

return 14;

else

if (intensity >= 240 && intensity < 255)

return 15;

else

return 0;

}

}

}

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="LogInPage.aspx.cs" Inherits="Online\_Photo\_Effects.View.LogInPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta charset="utf-8">

<title>Online Photo Effects</title>

<style type="text/css">

body {

background-color: #f4f4f4;

color: #5a5656;

font-family: 'Open Sans', Arial, Helvetica, sans-serif;

font-size: 16px;

line-height: 1.5em;

}

a {

text-decoration: none;

}

h1 {

font-size: 1em;

}

h1, p {

margin-bottom: 10px;

}

strong {

font-weight: bold;

}

.uppercase {

text-transform: uppercase;

}

/\* ---------- LOGIN ---------- \*/

#login {

margin: 50px auto;

width: 300px;

}

form fieldset input[type="text"], input[type="password"] {

background-color: #e5e5e5;

border: none;

border-radius: 3px;

-moz-border-radius: 3px;

-webkit-border-radius: 3px;

color: #5a5656;

font-family: 'Open Sans', Arial, Helvetica, sans-serif;

font-size: 14px;

height: 50px;

outline: none;

padding: 0px 10px;

width: 280px;

-webkit-appearance: none;

}

form fieldset input[type="submit"] {

background-color: #008dde;

border: none;

border-radius: 3px;

-moz-border-radius: 3px;

-webkit-border-radius: 3px;

color: #f4f4f4;

cursor: pointer;

font-family: 'Open Sans', Arial, Helvetica, sans-serif;

height: 50px;

text-transform: uppercase;

width: 300px;

-webkit-appearance: none;

}

form fieldset a {

color: #5a5656;

font-size: 10px;

}

form fieldset a:hover {

text-decoration: underline;

}

.btn-round {

background-color: #5a5656;

border-radius: 50%;

-moz-border-radius: 50%;

-webkit-border-radius: 50%;

color: #f4f4f4;

display: block;

font-size: 12px;

height: 50px;

line-height: 50px;

margin: 30px 125px;

text-align: center;

text-transform: uppercase;

width: 50px;

}

.facebook-before {

background-color: #0064ab;

border-radius: 3px 0px 0px 3px;

-moz-border-radius: 3px 0px 0px 3px;

-webkit-border-radius: 3px 0px 0px 3px;

color: #f4f4f4;

display: block;

float: left;

height: 50px;

line-height: 50px;

text-align: center;

width: 50px;

}

.facebook {

background-color: #0079ce;

border: none;

border-radius: 0px 3px 3px 0px;

-moz-border-radius: 0px 3px 3px 0px;

-webkit-border-radius: 0px 3px 3px 0px;

color: #f4f4f4;

cursor: pointer;

height: 50px;

text-transform: uppercase;

width: 250px;

}

.twitter-before {

background-color: #189bcb;

border-radius: 3px 0px 0px 3px;

-moz-border-radius: 3px 0px 0px 3px;

-webkit-border-radius: 3px 0px 0px 3px;

color: #f4f4f4;

display: block;

float: left;

height: 50px;

line-height: 50px;

text-align: center;

width: 50px;

}

.twitter {

background-color: #1bb2e9;

border: none;

border-radius: 0px 3px 3px 0px;

-moz-border-radius: 0px 3px 3px 0px;

-webkit-border-radius: 0px 3px 3px 0px;

color: #f4f4f4;

cursor: pointer;

height: 50px;

text-transform: uppercase;

width: 250px;

}

</style>

</head>

<body>

<form>

<div id="login">

<h1><strong>Welcome!</strong> Please login.</h1>

<form id="form1" runat="server">

<fieldset>

Username: <asp:Label ID="lblUsername" runat="server" Text=""></asp:Label>

<p><asp:TextBox ID="tbUsername" runat="server"></asp:TextBox></p>

Password: <asp:Label ID="lblPass" runat="server" Text=""></asp:Label>

<p><asp:TextBox ID="tbPassword" runat="server" TextMode="Password"></asp:TextBox></p>

<p><a href="#">Forgot Password?</a></p>

<p><asp:Button ID="btnLogIn" runat="server" Text="Log In" OnClick="btnLogIn\_Click" /></p>

</fieldset>

</form>

<p><span class="btn-round">or</span></p>

<p>

<a class="facebook-before"></a>

<button class="facebook">Login Using Facbook</button>

</p>

<p>

<a class="twitter-before"></a>

<button class="twitter">Login Using Twitter</button>

</p>

</div> <!-- end login -->

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Online\_Photo\_Effects.Model;

using Online\_Photo\_Effects.Code;

namespace Online\_Photo\_Effects.View

{

public partial class LogInPage : System.Web.UI.Page

{

userAccountsEntities users = new userAccountsEntities();

Helper helper = new Helper();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnLogIn\_Click(object sender, EventArgs e)

{

string username = tbUsername.Text;

string password = tbPassword.Text;

if (helper.checkUsername(username))

{

if (helper.checkPassword(password))

{

var user = users.user\_accounts.FirstOrDefault(u => u.userName == username);

if (user != null)

{

if (user.password == password)

{

if (user.premium == 0)

{

Server.Transfer("Editor.aspx");

}

}

else

{

lblPass.Text = "Incorrect login credentials";

}

}

}

else

{

lblPass.Text = "Check the password lenght";

}

}

else

{

lblUsername.Text = "Check the username lenght";

}

}

}

}

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Editor.aspx.cs" Inherits="Online\_Photo\_Effects.View.Editor" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

form fieldset input[type="submit"] {

background-color: #008dde;

border: none;

border-radius: 3px;

-moz-border-radius: 3px;

-webkit-border-radius: 3px;

color: #f4f4f4;

cursor: pointer;

font-family: 'Open Sans', Arial, Helvetica, sans-serif;

height: 50px;

text-transform: uppercase;

width: 300px;

-webkit-appearance: none;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:FileUpload ID="fileUploader" runat="server" />

<asp:Button ID="btnUpload" runat="server" Text="Upload" OnClick="btnUpload\_Click" />

<br />

<asp:Image ID="imageBox" runat="server" />

<br />

<fieldset>

<asp:Button ID="btnMagic1" runat="server" Text="Magic 1" OnClick="btnMagic1\_Click" />

&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnMagic2" runat="server" Text="Magic 2" OnClick="btnMagic2\_Click" />

&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnMagic3" runat="server" Text="Magic3" OnClick="btnMagic3\_Click" />

</fieldset>

</div>

</form>

</body>

</html>

using Online\_Photo\_Effects.Model;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Online\_Photo\_Effects.View

{

public partial class Editor : System.Web.UI.Page

{

private Bitmap bm;

ImageDetails imgD = new ImageDetails();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnUpload\_Click(object sender, EventArgs e)

{

if (fileUploader.HasFile)

{

string fileName = System.IO.Path.GetFileName(fileUploader.PostedFile.FileName);

fileUploader.PostedFile.SaveAs(Server.MapPath("~/Images/" + fileName));

imageBox.ImageUrl = "~/Images/" + fileName;

Session["fileName"] = fileName;

bm = new Bitmap(Server.MapPath(imageBox.ImageUrl));

Session["bitMap"] = bm;

imgD.ImageWidth = bm.Width;

Session["w"] = imgD.ImageWidth;

imgD.ImageHeight = bm.Height;

Session["h"] = imgD.ImageHeight;

}

else

{

}

}

protected void btnMagic1\_Click(object sender, EventArgs e)

{

imgD.ImageWidth = (int)Session["w"];

imgD.ImageHeight = (int)Session["h"];

string fileName = Session["fileName"].ToString();

MagicServicesReference.MagicServicesSoapClient magic = new MagicServicesReference.MagicServicesSoapClient();

string imageUrl = magic.Magic1(imgD.ImageWidth, imgD.ImageHeight, fileName);

imageBox.ImageUrl = imageUrl;

}

protected void btnMagic2\_Click(object sender, EventArgs e)

{

imgD.ImageWidth = (int)Session["w"];

imgD.ImageHeight = (int)Session["h"];

string fileName = Session["fileName"].ToString();

MagicServicesReference.MagicServicesSoapClient magic = new MagicServicesReference.MagicServicesSoapClient();

string imageUrl = magic.Magic2(imgD.ImageWidth, imgD.ImageHeight, fileName);

imageBox.ImageUrl = imageUrl;

}

protected void btnMagic3\_Click(object sender, EventArgs e)

{

imgD.ImageWidth = (int)Session["w"];

imgD.ImageHeight = (int)Session["h"];

string fileName = Session["fileName"].ToString();

MagicServicesReference.MagicServicesSoapClient magic = new MagicServicesReference.MagicServicesSoapClient();

string imageUrl = magic.Magic3(imgD.ImageWidth, imgD.ImageHeight, fileName);

imageBox.ImageUrl = imageUrl;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Online\_Photo\_Effects.Code

{

public class Helper

{

public bool checkUsername(string username)

{

if (username.Length != 0 && username.Length < 51)

{

if (username.All(Char.IsLetterOrDigit))

{

return true;

}

}

return false;

}

public bool checkPassword(string password)

{

if (password.Length > 5 && password.Length < 37)

{

return true;

}

return false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Online\_Photo\_Effects.Model

{

public class AccountsInfo

{

private int \_id;

private string \_fullName;

private string \_userName;

private string \_password;

private string \_email;

public int ID

{

get

{

return \_id;

}

set

{

\_id = value;

}

}

public string FullName

{

get

{

return \_fullName;

}

set

{

\_fullName = value;

}

}

public string UserName

{

get

{

return \_userName;

}

set

{

\_userName = value;

}

}

public string Password

{

get

{

return \_password;

}

set

{

\_password = value;

}

}

public string Email

{

get

{

return \_email;

}

set

{

\_email = value;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Online\_Photo\_Effects.Model

{

public class ImageDetails

{

private int \_w;

private int \_h;

public int ImageWidth

{

get

{

return \_w;

}

set

{

\_w = value;

}

}

public int ImageHeight

{

get

{

return \_h;

}

set

{

\_h = value;

}

}

}

}