



Barani Institute of Information Technology

NAME : TAYYABA KOUSAR

CLASS : BSCS-4A

REG NO : 2024-Arid-0200

SUBMITTED TO : SIR AFTAB

ASSINGMENT NO : 2

```

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 ASSINMENT2
{
    public partial class Form1 : Form
    {
        private Skills skillsManager = new Skills();
        private Color themeColor = Color.Blue;
        private Color backgroundColor = Color.White;
        public Form1()
        {
            InitializeComponent();

            DisplayWelcomeMessage();
        }

        private void DisplayWelcomeMessage()
        {
            labelWelcome.Text = $"Welcome - {DateTime.Now:dddd, MMMM dd, yyyy
hh:mm tt}";
        }

        private void Form1_Load(object sender, EventArgs e)
        {

            List<Occupation> occupations = new List<Occupation>()
            {
                new Occupation(1, "Student"),
                new Occupation(2, "Engineer"),
                new Occupation(3, "Designer"),
                new Occupation(4, "Teacher")
            };

            comboBox1.DataSource = occupations;
            comboBox1.DisplayMember = "Occp";
            comboBox1.ValueMember = "Code";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            openFileDialog1.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.bmp";

            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
                UpdatePreview();
            }
        }

        private void buttonThemeColor_Click(object sender, EventArgs e)
    }
}

```

```

{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        themeColor = colorDialog1.Color;
        groupBoxPreview.ForeColor = themeColor;
        UpdatePreview();
    }
}

private void UpdatePreview()
{
    labelPreviewName.Text = textBox1.Text;
    labelPreviewUsername.Text = textBox2.Text;
    labelPreviewEmail.Text = textBox3.Text;
    labelPreviewAge.Text = textBox4.Text;

    string gender = radioButton1.Checked ? "Male" : "Female";
    labelPreviewGender.Text = gender;

    string interests = GetSelectedInterests();
    labelPreviewInterests.Text = interests;

    labelPreviewOccupation.Text =
comboBox1.SelectedItem?.ToString() ?? "None";

    string skillsText = string.Join(", ",
skillsManager.GetAllSkills());
    labelPreviewSkills.Text = string.IsNullOrEmpty(skillsText) ? "No
skills" : skillsText;
}

private void buttonBackgroundColor_Click(object sender, EventArgs e)
{
    if (colorDialog2.ShowDialog() == DialogResult.OK)
    {
        backgroundColor = colorDialog2.Color;
        groupBoxPreview.BackColor = backgroundColor;
        UpdatePreview();
    }
}

private void buttonAddSkill_Click(object sender, EventArgs e)
{
    string skill = textBoxSkill.Text.Trim();

    if (string.IsNullOrEmpty(skill))
    {
        MessageBox.Show("Please enter a skill.", "Input Error");
        return;
    }

    skillsManager.AddSkill(skill);

    if (skillsManager.GetAllSkills().Count > 0
&& listBoxSkills.Items.Contains(skill))
    {
}

```

```

        listBoxSkills.Items.Add(skill);
    }
    else if (skillsManager.GetAllSkills().Contains(skill) &&
listBoxSkills.Items.Contains(skill))
    {
        MessageBox.Show("This skill already exists.", "Duplicate
Error");
        return;
    }

    textBoxSkill.Clear();
    UpdatePreview();
}

private void buttonRemoveSkill_Click(object sender, EventArgs e)
{
    if (listBoxSkills.SelectedIndex != -1)
    {
        string skill = listBoxSkills.SelectedItem.ToString();
        skillsManager.RemoveSkill(skill);
        listBoxSkills.Items.RemoveAt(listBoxSkills.SelectedIndex);
        UpdatePreview();
    }
}

private bool ValidateFields()
{
    if (string.IsNullOrWhiteSpace(textBox1.Text))
    {
        MessageBox.Show("Full Name cannot be empty.", "Validation
Error");
        return false;
    }

    if (string.IsNullOrWhiteSpace(textBox2.Text))
    {
        MessageBox.Show("Username cannot be empty.", "Validation
Error");
        return false;
    }

    if (!IsValidEmail(textBox3.Text))
    {
        MessageBox.Show("Email must contain @ and .", "Validation
Error");
        return false;
    }

    if (!int.TryParse(textBox4.Text, out int age) || age < 13 ||

age > 100)
    {
        MessageBox.Show("Age must be between 13 and 100.",
"Validation Error");
        return false;
    }

    return true;
}

```

```

private string GetSelectedInterests()
{
    List<string> interests = new List<string>();

    if (checkBox1.Checked) interests.Add("Reading");
    if (checkBox2.Checked) interests.Add("Sports");
    if (checkBox3.Checked) interests.Add("Art");
    if (checkBox4.Checked) interests.Add("Traveling");
    if (checkBox5.Checked) interests.Add("Coding");

    return interests.Count > 0 ? string.Join(", ", interests) :
"None";
}
private bool IsValidEmail(string email)
{
    return email.Contains("@") && email.Contains(".");
}

private void buttonSave_Click(object sender, EventArgs e)
{
    if (!ValidateFields())
        return;

    MessageBox.Show("Profile saved successfully!", "Success");
}
private void buttonReset_Click(object sender, EventArgs e)
{
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
    radioButton1.Checked = true;
    checkBox1.Checked = false;
    checkBox2.Checked = false;
    checkBox3.Checked = false;
    checkBox4.Checked = false;
    checkBox5.Checked = false;
    comboBox1.SelectedIndex = 0;
    listBoxSkills.Items.Clear();
    skillsManager = new Skills();
    pictureBox1.Image = null;
    themeColor = Color.Blue;
    backgroundColor = Color.White;
    UpdatePreview();
    MessageBox.Show("All fields reset.", "Reset");
}
private void buttonViewSummary_Click(object sender, EventArgs e)
{
    if (!ValidateFields())
        return;

    Form summaryForm = new Form();
    summaryForm.Text = "Profile Summary";
    summaryForm.Width = 400;
    summaryForm.Height = 500;
    summaryForm.StartPosition = FormStartPosition.CenterParent;
}

```

```

    TextBox textBoxSummary = new TextBox();
    textBoxSummary.Multiline = true;
    textBoxSummary.ReadOnly = true;
    textBoxSummary.Dock = DockStyle.Fill;
    textBoxSummary.Font = new Font("Arial", 10);

    string summary = $@"== PROFILE SUMMARY ==

Full Name: {textBox1.Text}
Username: {textBox2.Text}
Email: {textBox3.Text}
Age: {textBox4.Text}
Gender: {(radioButton1.Checked ? "Male" : "Female")}
Interests: {GetSelectedInterests()}
Occupation: {comboBox1.SelectedItem}
Skills: {string.Join(", ", skillsManager.GetAllSkills())}
Created: {DateTime.Now:dddd, MMMM dd, yyyy};

    textBoxSummary.Text = summary;
    summaryForm.Controls.Add(textBoxSummary);
    summaryForm.ShowDialog();
}

private void textBox1_TextChanged(object sender, EventArgs e) =>
UpdatePreview();
private void textBox2_TextChanged(object sender, EventArgs e) =>
UpdatePreview();
private void textBox3_TextChanged(object sender, EventArgs e) =>
UpdatePreview();
private void textBox4_TextChanged(object sender, EventArgs e) =>
UpdatePreview();

private void checkBox1_CheckedChanged(object sender, EventArgs e) =>
UpdatePreview();
private void checkBox2_CheckedChanged(object sender, EventArgs e) =>
UpdatePreview();
private void checkBox3_CheckedChanged(object sender, EventArgs e) =>
UpdatePreview();
private void checkBox4_CheckedChanged(object sender, EventArgs e) =>
UpdatePreview();
private void checkBox5_CheckedChanged(object sender, EventArgs e) =>
UpdatePreview();

private void radioButton1_CheckedChanged(object sender, EventArgs e)
=> UpdatePreview();
private void radioButton2_CheckedChanged(object sender, EventArgs e)
=> UpdatePreview();

private void comboBox1_SelectedIndexChanged(object sender, EventArgs
e) => UpdatePreview();
}

```

```

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

namespace ASSINMENT2
{
    internal class Occupation
    {
        public int Code { get; set; }
        public String Occp { get; set; }

        public Occupation(int code, String occp)
        {
            this.Code = code;
            this.Occp = occp;
        }

        public override string ToString()
        {
            return Occp;
        }
    }
}

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

namespace ASSINMENT2
{
    internal class Skills
    {
        public List<string> SkillList { get; set; }

        public Skills()
        {
            SkillList = new List<string>();
        }

        public void AddSkill(string skill)
        {
            if (!SkillList.Contains(skill) && !string.IsNullOrEmpty(skill))
            {
                SkillList.Add(skill);
            }
        }

        public void RemoveSkill(string skill)
        {
            SkillList.Remove(skill);
        }

        public List<string> GetAllSkills()
        {
            return SkillList;
        }
    }
}

```

}
}