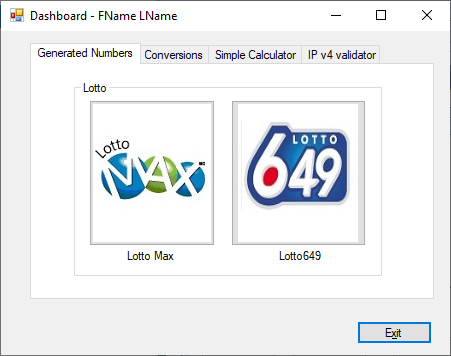
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Meet Hingu  11/8/2024  Version 1.0 |

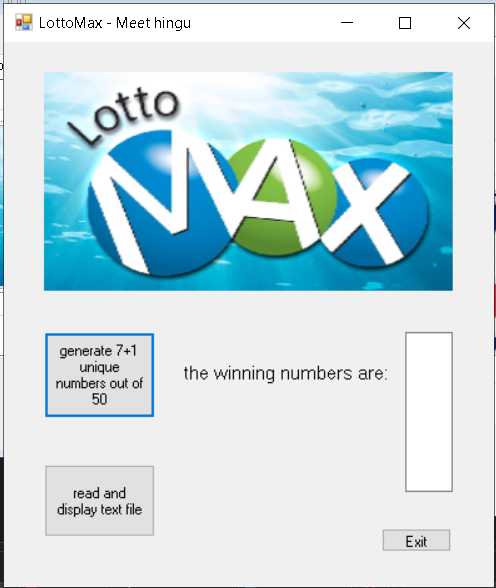
1. **Start by adding a short description of your project, and the languages (technologies) used:**

This first section of the project is about lottery and generating random numbers and display them.

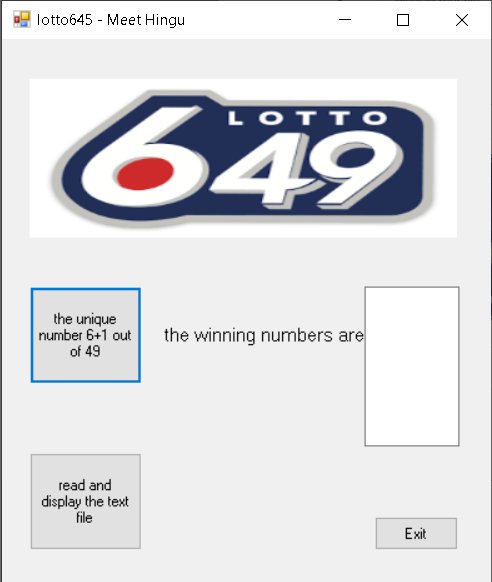
1. Language: C#
2. Used tool(s): VS2022 Windows Form (.Net Framework), MS Word, AI (ChatGPT)
3. **Present the print screens of yours forms and have a detailed description of the functionalities (step by step).**



1. If you click on tab LottoMax



1. If you click on tab Lotto649

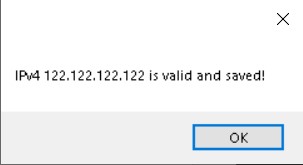


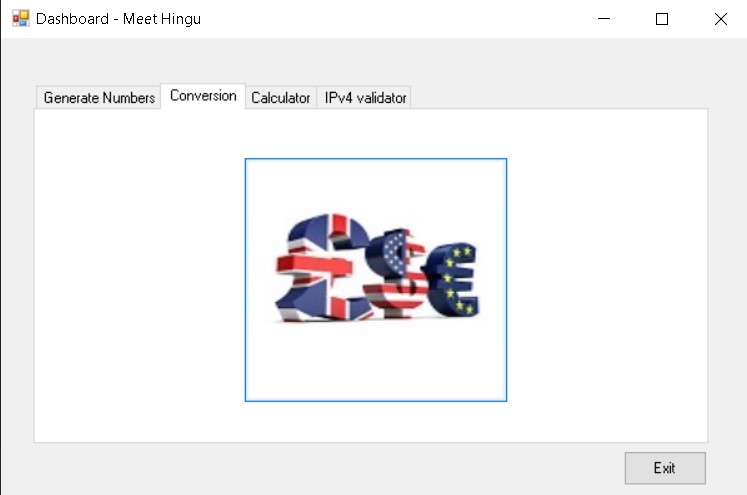
3.when you go to the IPv4 validator

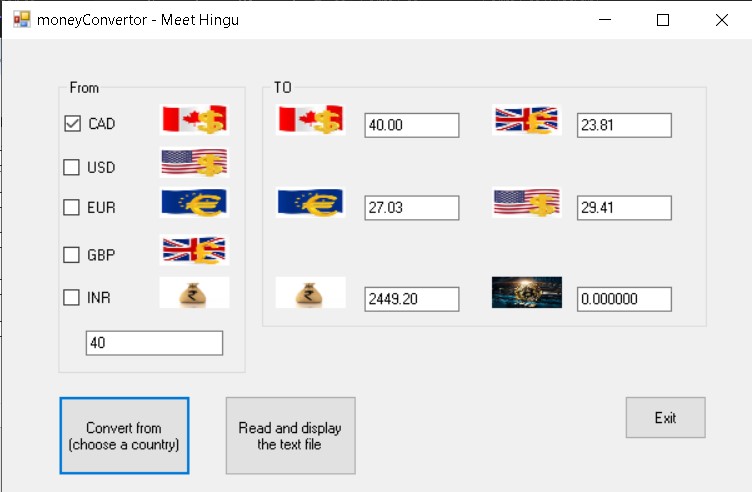


4.when click on the IP button.

5. success Message



6. When click in conversion

7. when you convert the amount into different currency.

1. **Present the code of your application (forms).**

**Dashboard**

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 meet\_project

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

lotto649 obj = new lotto649();

obj.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

lottoMax obj = new lottoMax();

obj.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button4\_Click(object sender, EventArgs e)

{

ipv4 obj = new ipv4();

obj.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

moneyConvertor obj = new moneyConvertor();

obj.Show();

}

}

}

… your code goes here

LotoMAX

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 MainProjectSection1

{

public partial class lottoMax : Form

{

public lottoMax()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Random random = new Random();

string randomNumbers = "";

for (int i = 0; i < 8; i++)

{

int randomNumber = random.Next(1, 50);

randomNumbers += randomNumber.ToString() + Environment.NewLine;

}

textBox1.Text = randomNumbers.Trim();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

… your code goes here

Loto649

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 MainProjectSection1

{

public partial class lotto645 : Form

{

public lotto645()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Random random = new Random();

string randomNumbers = "";

for (int i = 0; i < 7; i++)

{

int randomNumber = random.Next(1, 49);

randomNumbers += randomNumber.ToString() + Environment.NewLine;

}

textBox1.Text = randomNumbers.Trim();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

… your code goes here

Money Conversion

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace meet\_project

{

public partial class moneyConvertor : Form

{

public moneyConvertor()

{

InitializeComponent();

}

public class MoneyConv

{

// Conversion factors relative to CAD

private readonly double[] exchangeRates = { 1.00, 0.7353, 0.6757, 0.5952, 61.23, 0.000023 }; // INR and Bitcoin rates

private readonly string[] currencyCodes = { "CAD", "USD", "EUR", "GBP", "INR", "BTC" };

public bool IsValidAmount(string amount)

{

// Validate input format (0.00 to 999.99)

string pattern = @"^\d{1,3}(\.\d{1,2})?$";

return Regex.IsMatch(amount, pattern);

}

public double[] ConvertAmount(double baseAmount)

{

double[] results = new double[exchangeRates.Length];

for (int i = 0; i < exchangeRates.Length; i++)

{

results[i] = Math.Round(baseAmount \* exchangeRates[i], 2);

}

return results;

}

public void SaveConversions(string baseCurrency, double baseAmount, double[] conversions)

{

string filePath = "MoneyConversions.txt";

using (StreamWriter writer = new StreamWriter(filePath, true))

{

string dateTime = DateTime.Now.ToString("yyyy/MM/dd (hh:mm:ss tt)");

string log = $"{dateTime}, {baseAmount} {baseCurrency} = ";

for (int i = 0; i < conversions.Length; i++)

{

log += $"{conversions[i]} {currencyCodes[i]}";

if (i < conversions.Length - 1) log += "; ";

}

writer.WriteLine(log);

}

}

public string ReadConversions()

{

string filePath = "MoneyConversions.txt";

if (!File.Exists(filePath))

return "No conversion records found.";

using (StreamReader reader = new StreamReader(filePath))

{

return reader.ReadToEnd();

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

MoneyConv converter = new MoneyConv();

string amountText = textBox1.Text.Trim();

if (!converter.IsValidAmount(amountText))

{

MessageBox.Show("Invalid amount! Enter a value between 0.00 and 999.99.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

double baseAmount = double.Parse(amountText);

double[] conversions = converter.ConvertAmount(baseAmount);

// Display results in respective TextBoxes

textboxCAD.Text = conversions[0].ToString("F2");

textboxUSA.Text = conversions[1].ToString("F2");

textboxEUR.Text = conversions[2].ToString("F2");

textboxGBP.Text = conversions[3].ToString("F2");

textboxINR.Text = conversions[4].ToString("F2"); // INR

textboxBTC.Text = conversions[5].ToString("F6");

}

private void button2\_Click(object sender, EventArgs e)

{

MoneyConv converter = new MoneyConv();

string baseCurrency = "CAD"; // Assume CAD as default; modify as needed

string amountText = textBox1.Text.Trim();

if (!converter.IsValidAmount(amountText))

{

MessageBox.Show("Invalid amount! Save operation failed.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

double baseAmount = double.Parse(amountText);

double[] conversions = converter.ConvertAmount(baseAmount);

try

{

converter.SaveConversions(baseCurrency, baseAmount, conversions);

MessageBox.Show("Conversions saved successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Error saving conversions: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

… your code goes here

IPv4 & IP

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace meet\_project

{

public partial class ipv4 : Form

{

public ipv4()

{

InitializeComponent();

}

public class IPv4IPv6Conv

{

// Validate IPv4

public bool IsValidIPv4(string ip)

{

string pattern = @"^(\d{1,3}\.){3}\d{1,3}$";

if (Regex.IsMatch(ip, pattern))

{

string[] parts = ip.Split('.');

foreach (string part in parts)

{

if (int.Parse(part) > 255) return false;

}

return true;

}

return false;

}

// Validate IPv6

public bool IsValidIPv6(string ip)

{

string pattern = @"^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}$";

return Regex.IsMatch(ip, pattern);

}

public void SaveIP(string type, string ip)

{

string filePath = "IPRecords.bin";

using (FileStream fs = new FileStream(filePath, FileMode.Append))

using (BinaryWriter writer = new BinaryWriter(fs))

{

writer.Write($"{type}: {ip}");

writer.Write(DateTime.Now.ToString());

}

}

}

private void ipv4\_Load(object sender, EventArgs e)

{

label3.Text = "Today :" + DateTime.Now.ToLongDateString();

}

private void button4\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

MessageBox.Show("Data has been Reset.");

textBox1.Focus();

}

private void button1\_Click(object sender, EventArgs e)

{

string ipv4 = textBox1.Text.Trim(); // IPv4 input

string ipv6 = textBox2.Text.Trim(); // IPv6 input

try

{

IPv4IPv6Conv converter = new IPv4IPv6Conv();

if (converter.IsValidIPv4(ipv4))

{

converter.SaveIP("IPv4", ipv4);

MessageBox.Show($"IPv4 {ipv4} is valid and saved!");

}

else if (converter.IsValidIPv6(ipv6))

{

converter.SaveIP("IPv6", ipv6);

MessageBox.Show($"IPv6 {ipv6} is valid and saved!");

}

else

{

MessageBox.Show("Invalid IP Address!");

}

}

catch (Exception ex)

{

MessageBox.Show($"Error: {ex.Message}");

}

}

}

}

… your code goes here

Calculator

… your code goes here

1. **Present the classes and/or methods that you create or you did use in the project and the class diagram(s) and the algorithm of the method(s) .**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class Calc | Enter a short description of the class or of the method ………………. |
| 1. Method for Calc class |  |
|  |  |
| 1. Class IP | Enter a short description of the class or of the method ……………….. |
| 1. void Method(int total) | Enter a short description of the class or of the method ……….. |
|  |  |
| 1. Constructor() | Enter a short description of the class or of the method ………… |
|  |  |

**Class diagram:**

**Method(s) algorithm:**

1. **Present the difficulties that you have, what was the hardest and the easiest part of your project application development.**

2014-10-30 8:30 – Can’t open the form by clicking the button.

8:45 Solved: Asking somebody…

… your presentation goes here …