JK INSTITUTE OF APPLIED PHYSICS AND TECHNOLOGY

University of Allahabad

Project Report on

Weather Application using C#

Prepared By:

Avinash Kumar Yadav

Guided By:

Mrs. Vanadana Rathore

Table Of Contents

JK INSTITUTE OF APPLIED PHYSICS AND TECHNOLOGY, UNIVERSITY OF ALLAHABAD

Certificate of Original

This is to certify that Avinash Kumar Yadav of B. Tech 6th Semester, Computer Science & Engineering, having enrollment no. 16AU/429 has successfully completed his project on Weather Application using C# and Open Weather Api for the session 2018-2019 as stated within the syllabus. This project is a bonafide piece of work carried out with the consultation of her guide Mrs. Vandana Rathore.

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.
We would like to take the opportunity to express my humble gratitude to Mrs. Vandana Rathore under whom we executed this project. Her constant guidance and willingness to share his vast knowledge made me understand this project and its manifestations in great depths and helped me to complete the assigned tasks.
We would like to thank all faculty members and staffs of the Department of Computer Science and Engineering, J.K Institute of Applied Physics & Technology for their generous help in various ways for the completion of this thesis.
Finally, yet importantly, We would like to express my heartfelt thanks to my beloved parents for their blessings, my friends and classmates for their help and wishes for the successful completion of this project.

AVINASH KUMAR YADAV
B.Tech(6th Semester)
Computer Science & Engineering

Introduction

It is an application which gives you the details of the present weather such as temperature conditions, wind speed, humidity etc. It also provides a brief forecast for the weather ahead.

To get a forecast or other data, you build a URL describing your request and navigate to it. You then parse the returned XML or JSON data to see what the details provided by forecasting API.

The first step in using openweathermap.org is to get an API ID. You can get one for free. You will need to sign up for a free account. After you get an API ID, you can use it in your URLs.

The second step is using the data provided by weather Api store the value in variable and write a program to show the details upon submitting request by enter location by the user.

The third step is to retrieve vale of variable and display it in GUI (Graphical User Interface). C#, the most rapid and convenient way to create your user interface is to do so visually, using the Windows Forms Designer and Toolbox. Windows Forms controls are reusable components that encapsulate user interface functionality and are used in client side Windows based applications.

Open Weather API

OpenWeatherMap is an online service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications. For data sources, it utilizes meteorological broadcast services, raw data from airport weather stations, raw data from radar stations, and raw data from other official weather stations. All data is processed by OpenWeatherMap in a way that it attempts to provide accurate online weather forecast data and weather maps, such as those for clouds or precipitation. Beyond that, the service is focused on the social aspect by involving weather station owners in connecting to the service and thereby increasing weather data accuracy.

Open Weather API is the easiest one for working with weather data. To get started you need to receive your unique API key (APPID) after signup. After getting API key you can access current weather data for any location on Earth including over 200,000 cities. Current weather is frequently updated based on global models and data from more than 40,000 weather stations. Data is available in JSON, XML, or HTML format.
You can call API:

- By city name
- By city ID
- By geographic coordinates
- By ZIP code

5-days forecast is available for any location or city. It includes weather data every 3 hours. Forecast is available in JSON or XML format.

Proposed Methodology

Windows Forms is a Graphical User Interface(GUI) class library which is bundled in .*Net Framework*. Its main purpose is to provide an easier interface to develop the applications for desktop, tablet, PCs. It is also termed as the WinForms. The applications which are developed by using Windows Forms are known as the Windows Forms Applications that runs on the desktop computer. WinForms applications can contain the different type of controls like labels, list boxes, tooltip etc. Open the Visual Studio then Go to File -> New -> Project to create a new project and then select the language as *Visual C#* from the left menu. Click on *Windows Forms App(.NET Framework)* in the middle of current window. After that give the project name and Click **OK**. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole. On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms. In this step, we will now add a label to the Form which will display "Hello World." From the toolbox, you will need to choose the Label control and simply drag it onto the Form. The next step is to go to the properties of the control and Change the text to 'Hello World'. To go to the properties of a control, you need to right-click the control and choose the Properties menu option. The properties panel also shows up in Visual Studio. So for the label control, in the properties control, go to the Text section and enter "Hello World". Each Control has a set of properties which describe the control. Using this methodology we create whole design of weather Application.

Implementation

The first goal is to create a project in visual studio using C# Windows Form Application.

And then create the GUI of the application using common controls from toolbox from left side pane of visual studio. We have used Label for labeling data , Textbox for input field of location from user, Picturebox for picture of weather and Button for submitting request after entering location. With the help of textbox input we build the url for getting data from open weather api using user location and Apikey in the form of Json and after that we format the retrieved Json data and put them in separate variable for each queried data and then displayed them in label using setting the value of label item with that variable name.We have displayed the icon of weather condition using PictureBox after receiving the icon provided by weather map api and updating the value of variable. If any error occurred during processing query, we have show user error occurred during processing with the help of MessageBox. We have converted the Kelvin temperature we received from open weather api using Celsius temperature conversion formula and then updated the value of Celsius Label variable value. We have used label for showing humidity, wind pressure, latitude and longitude, weather description of input user location. After entering any location provided by user in textbox the application proceeds to form url to get Json data from Open weather Api and after that if there is no any error it get displayed in label provided using common control form of Visual Studio.

Tools Required

Hardware:-

The given Application can run in any desktop computer, laptop that support windows operating system which is main requirement of windows form Application with any minimal configuration.

Software:-

The Application required Visual Studio 2017 version and Windows 10 Home OS. You must install version 2.0, 3.0, or 3.5 of the Microsoft .NET Framework on the same computer as PowerBuilder. For intelligent update applications, you must also install the .NET Framework 2.0, 3.0, or 3.5 SDK (x86).

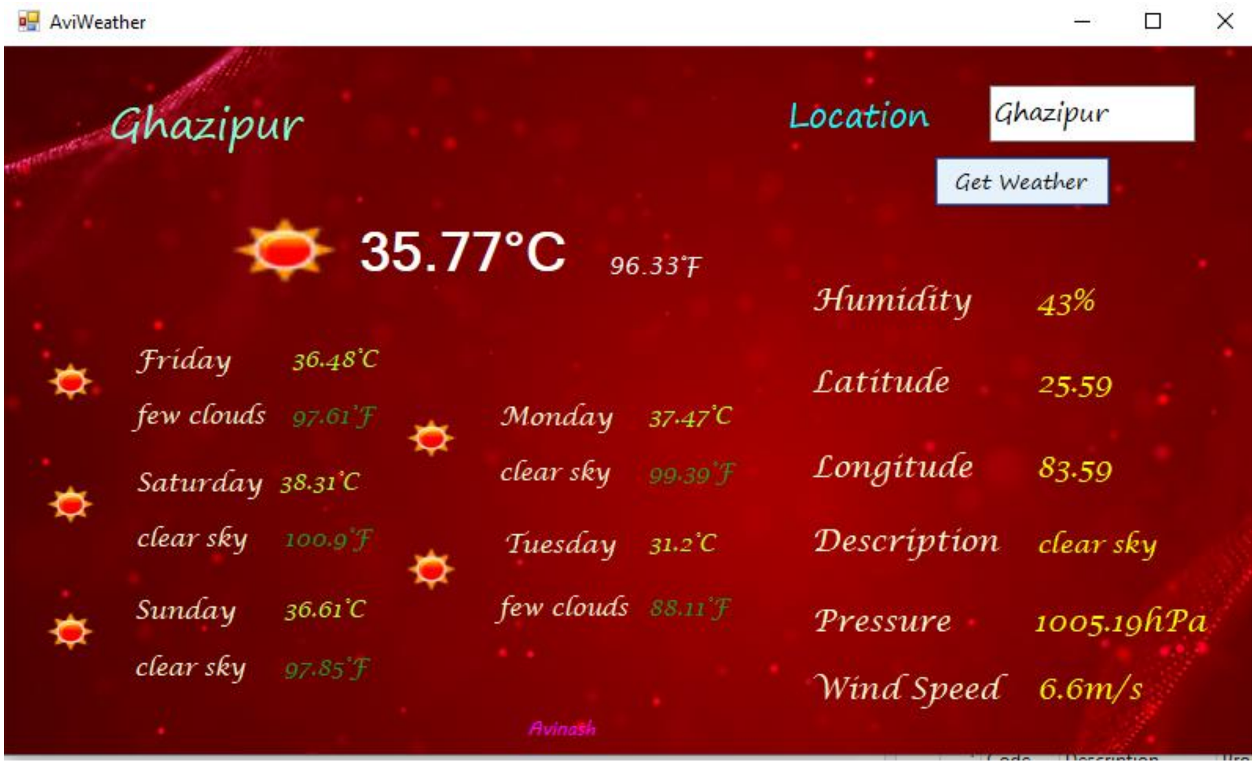
Codes

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Json;
using System.Net;

namespace AviWeather
{
    public partial class Form1 : Form
    {
        String City;
        String TemperatureF;
        String Temperature;
        String Humidity;
        String Pressure;
        String WindSpeed;
        String Country;
        String Icon;
        String Description;
        String Latitude;
        String Longitude;
        String OneIcon;
        String TwoIcon;
        String ThreeIcon;
        String FourIcon;
        String FiveIcon;
        String FirstDay;
        String SecondDay;
        String ThirdDay;
        String FourDay;
        String FiveDay;
        String FirstDescription;
        String SecondDescription;
        String ThirdDescription;
        String FourDescription;
        String FiveDescription;
        String TempCOne;
        String TempFOne;
        String TempCTwo;
        String TempFTwo;
        String TempCThree;
        String TempFThree;
        String TempCFour;
        String TempFFour;
        String TempCFive;
        String TempFFive;
```

```
public Form1()
        {
            InitializeComponent();
            labelCity.BackColor = Color.Transparent;
            labelDescription.BackColor = Color.Transparent;
            labelHumidity.BackColor = Color.Transparent;
            labelHumidityRes.BackColor = Color.Transparent;
            labelLatitude.BackColor = Color.Transparent;
            labelLatitudeRes.BackColor = Color.Transparent;
            labelLongitude.BackColor = Color.Transparent;
            labelLongitudeRes.BackColor = Color.Transparent;
            labelPressure.BackColor = Color.Transparent;
            labelPressureRes.BackColor = Color.Transparent;
            labelTemperature.BackColor = Color.Transparent;
            labelTemperatureF.BackColor = Color.Transparent;
            labelWindSpeed.BackColor = Color.Transparent;
            labelWindSpeedRes.BackColor = Color.Transparent;
            label6.BackColor = Color.Transparent;
            labelLocation.BackColor = Color.Transparent;
            pictureBox1.BackColor = Color.Transparent;
            pictureBoxFive.BackColor = Color.Transparent;
            pictureBoxFour.BackColor = Color.Transparent;
            pictureBoxT.BackColor = Color.Transparent;
            pictureBoxS.BackColor = Color.Transparent;
            pictureBoxF.BackColor = Color.Transparent;
            labelDayOne.BackColor = Color.Transparent;
            labelDayFour.BackColor = Color.Transparent;
            labelDayFive.BackColor = Color.Transparent;
            labelDayThree.BackColor = Color.Transparent;
            labelDayTwo.BackColor = Color.Transparent;
            labelDescFive.BackColor = Color.Transparent;
            labelDescFour.BackColor = Color.Transparent;
            labelDescOne.BackColor = Color.Transparent;
            labelDescTwo.BackColor = Color.Transparent;
            labelDescThree.BackColor = Color.Transparent;
            labelTemcFive.BackColor = Color.Transparent;
            labelTemcFour.BackColor = Color.Transparent;
            labelTemcThree.BackColor = Color.Transparent;
            labeltemcTwo.BackColor = Color.Transparent;
            labelTemcOne.BackColor = Color.Transparent;
            labelTemfFive.BackColor = Color.Transparent;
            labelTemfFour.BackColor = Color.Transparent;
            labelTemfThree.BackColor = Color.Transparent;
            labelTemfTwo.BackColor = Color.Transparent;
            labelTemfOne.BackColor = Color.Transparent;
            label1.BackColor = Color.Transparent;
        }
        void setIcon(PictureBox pb,String icon)
        {
            try
            {
                pb.Load("http://openweathermap.org/img/w/" + Icon + ".png");
            }
            catch (Exception exc)
            {
                MessageBox.Show("Sorry Cannot Load Image");
            }
        }
```

```
private Double celsiusTemp(String temp)
        {
            Double fahTemp = Convert.ToDouble(temp);
            Double celTem = (fahTemp - 32) * 0.556;
            celTem = Math.Round(celTem, 2);
            return celTem;
        }
        private void getForecastWeather(String location)
        {
            try
            {
                WebClient client = new WebClient();

                dynamic res =
JsonValue.Parse(client.DownloadString("http://api.openweathermap.org/data/2.5/forecast?q=
" + location + "&units=imperial&appid=d99414c365ff0845c676431520203ea0"));
                dynamic list = res["list"];
                dynamic firstDay = list[9];
                dynamic secondDay = list[16];
                dynamic thirdDay = list[24];
                dynamic fourthDay = list[32];
                dynamic fifthDay = list[38];
                dynamic fifthDesc = fifthDay["weather"][0];
                FiveDescription = fifthDesc["description"];
                FiveIcon = fifthDesc["icon"];
                dynamic temp5 = fifthDay["main"];
                double TemFive = temp5["temp"];
                TempFFive = Convert.ToString(TemFive);
                TempCFive =  Convert.ToString(celsiusTemp(TempFFive));
                dynamic fourthDesc = fourthDay["weather"][0];
                dynamic temp4 = fourthDay["main"];
                double TemFour = temp4["temp"];
                TempFFour = Convert.ToString(TemFour);
                TempCFour = Convert.ToString(celsiusTemp(TempFFour));
                dynamic temp3 = thirdDay["main"];
                double TemThree = temp3["temp"];
                TempFThree = Convert.ToString(TemThree);
                TempCThree = Convert.ToString(celsiusTemp(TempFThree));
                dynamic temp2 = secondDay["main"];
                double TemTwo = temp2["temp"];
                TempFTwo = Convert.ToString(TemTwo);
                TempCTwo = Convert.ToString(celsiusTemp(TempFTwo));
                dynamic temp1 = firstDay["main"];
                double TemOne = temp1["temp"];
                TempFOne = Convert.ToString(TemOne);
                TempCOne = Convert.ToString(celsiusTemp(TempFOne));
                FourDescription = fourthDesc["description"];
                FourIcon = fourthDesc["icon"];
                dynamic thirdDesc = thirdDay["weather"][0];
                ThirdDescription = thirdDesc["description"];
                ThreeIcon = thirdDesc["icon"];
                dynamic secondDesc = secondDay["weather"][0];
                SecondDescription = secondDesc["description"];
                TwoIcon = secondDesc["icon"];
                dynamic firstDesc = firstDay["weather"][0];
                FirstDescription = firstDesc["description"];
                OneIcon = firstDesc["icon"];
                String dateTime = firstDay["dt_txt"];
                DateTime daysCon = Convert.ToDateTime(dateTime);
                FirstDay = daysCon.DayOfWeek.ToString();
                String dt2 = secondDay["dt_txt"];
                DateTime daysTwo = Convert.ToDateTime(dt2);
                SecondDay = daysTwo.DayOfWeek.ToString();
                String dt3 = thirdDay["dt_txt"];
                DateTime daysThree = Convert.ToDateTime(dt3);
                ThirdDay = daysThree.DayOfWeek.ToString();
                String dt4 = fourthDay["dt_txt"];
                DateTime daysFour = Convert.ToDateTime(dt4);
                FourDay = daysFour.DayOfWeek.ToString();
                String dt5 = fifthDay["dt_txt"];
                DateTime daysfif = Convert.ToDateTime(dt5);
                FiveDay = daysfif.DayOfWeek.ToString();
            }
            catch (Exception excep)
            {
                MessageBox.Show("There is some error");
            }

        }

        private void button1_Click(object sender, EventArgs e)
        {
            String location = textBoxLocation.Text;
            WebClient webClient = new WebClient();
            try
            {
                String ans = 
webClient.DownloadString("http://api.openweathermap.org/data/2.5/weather?q=" + location + 
"&units=imperial&appid=d99414c365ff0845c676431520203ea0");
                dynamic result = JsonValue.Parse(ans);
                dynamic ans1 = result["main"];
                double hum = ans1["humidity"];
                double temp = ans1["temp"];
                double cel = (temp - 32) * 0.556;
                cel = Math.Round(cel, 2);
                Temperature = Convert.ToString(cel);
                TemperatureF = Convert.ToString(temp);
                double press = ans1["pressure"];
                Pressure = Convert.ToString(press);
                Humidity = Convert.ToString(hum);
                dynamic ans2 = result["sys"];
                Country = ans2["country"];
                dynamic ans3 = result["wind"];
                double speed = ans3["speed"];
                WindSpeed = Convert.ToString(speed);
                dynamic ans4 = result["coord"];
                double lon = ans4["lon"];
                Longitude = Convert.ToString(lon);
                double lat = ans4["lat"];
                Latitude = Convert.ToString(lat);
                City = result["name"];
                dynamic ans5 = result["weather"];
                dynamic ans6 = ans5[0];
                Description = ans6["description"];
                Icon = ans6["icon"];
```

```
                     Console.Write("");

                }
                catch (System.Net.WebException ep)
                {
                    MessageBox.Show("Please Check Your Internet Connection");
                }
                catch (Exception ex)
                {
                    //Console.WriteLine("Sorry");
                    //if(typeof(ex) ==System.Net.WebException)
                    MessageBox.Show("Sorry, There is some error.");
                }

                labelCity.Text = City;
                labelHumidityRes.Text = Humidity +"%";
                labelLatitudeRes.Text = Latitude;
                labelLongitudeRes.Text = Longitude;
                labelPressureRes.Text = Pressure +"hPa";
                labelTemperature.Text = Temperature + "°C";
                labelWindSpeedRes.Text = WindSpeed +"m/s";
                labelTemperatureF.Text = TemperatureF + "°F";
                label6.Text = Description;
                setIcon(pictureBox1,Icon);
                getForecastWeather(location);
                labelDayFive.Text = FiveDay;
                labelDayFour.Text = FourDay;
                labelDayThree.Text = ThirdDay;
                labelDayTwo.Text = SecondDay;
                labelDayOne.Text = FirstDay;
                labelDescFive.Text = FiveDescription;
                labelDescFour.Text = FourDescription;
                labelDescOne.Text = FirstDescription;
                labelDescThree.Text = ThirdDescription;
                labelDescTwo.Text = SecondDescription;
                labelTemfFive.Text = TempFFive + "°F";
                labelTemfFour.Text = TempFFour + "°F";
                labelTemfThree.Text = TempFThree + "°F";
                labelTemfTwo.Text = TempFTwo + "°F";
                labelTemfOne.Text = TempFOne + "°F";
                labelTemcFive.Text = TempCFive + "°C";
                labelTemcFour.Text = TempCFour + "°C";
                labelTemcThree.Text = TempCThree + "°C";
                labeltemcTwo.Text = TempCTwo + "°C";
                labelTemcOne.Text = TempCOne + "°C";
                setIcon(pictureBoxFive, FiveIcon);
                setIcon(pictureBoxFour, FourIcon);
                setIcon(pictureBoxT, ThreeIcon);
                setIcon(pictureBoxS, TwoIcon);
                setIcon(pictureBoxF, OneIcon);

        }
    }
    }
```



Ghazipur

Location Ghazipur

Get Weather

35.77°C 96.33˚F

Friday 36.48˚C
few clouds 97.61˚F

Saturday 38.31˚C
clear sky 100.9˚F

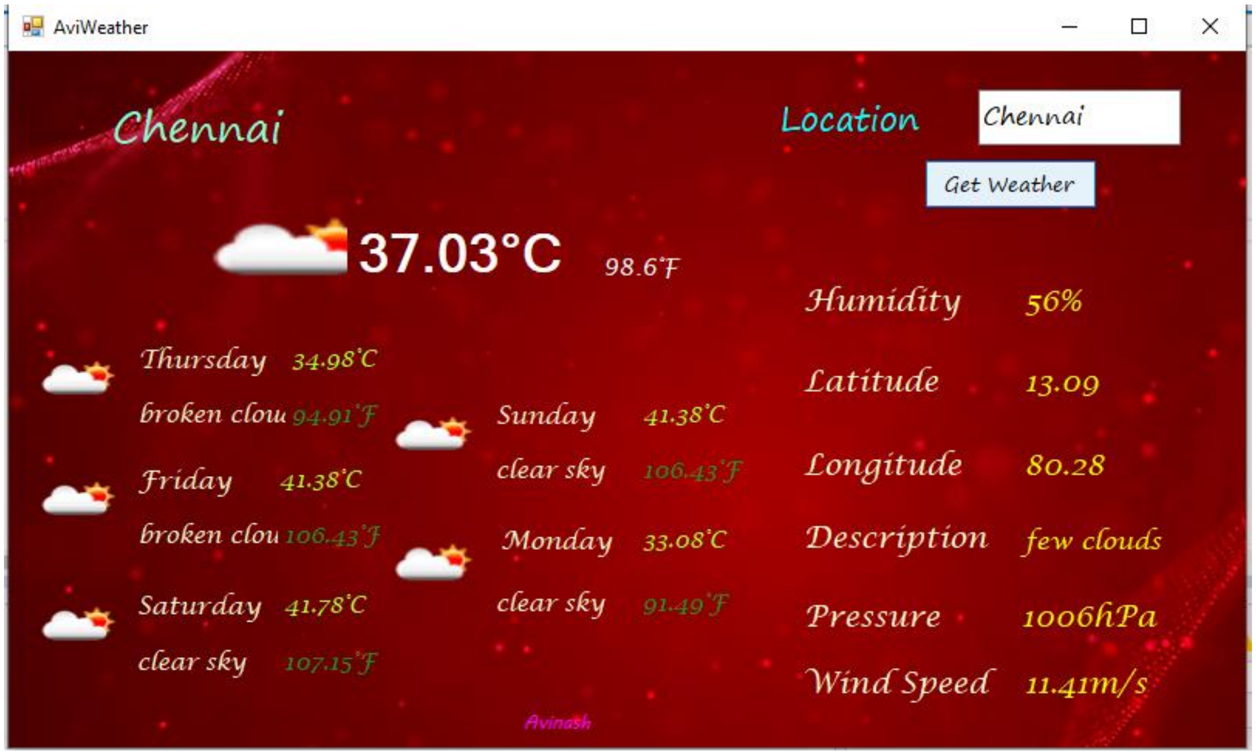
Sunday 36.61˚C
clear sky 97.85˚F

Monday 37.47˚C
clear sky 99.39˚F

Tuesday 31.2˚C
few clouds 88.11˚F

Humidity	43%
Latitude	25.59
Longitude	83.59
Description	clear sky
Pressure	1005.19hPa
Wind Speed	6.6m/s

Avinash



Chennai

Location Chennai

Get Weather

37.03°C 98.6˚F

Thursday 34.98˚C
broken clou 94.91˚F

Friday 41.38˚C
broken clou 106.43˚F

Saturday 41.78˚C
clear sky 107.15˚F

Sunday 41.38˚C
clear sky 106.43˚F

Monday 33.08˚C
clear sky 91.49˚F

Humidity	56%
Latitude	13.09
Longitude	80.28
Description	few clouds
Pressure	1006hPa
Wind Speed	11.41m/s

Avinash

Conclusion

OpenWeatherMap API helps to provide real time weather data in form of Json and Xml format so that developer can easily develop windows and web based application for user. It is very simple to get forecast weather data and easily integrate to the application. I have implemented five days weather forecast and current weather data based on location. I could not further extend my work to show data based on other parameters like latitude and longitude, or by ZIP code of any place. The application can be extended to use current location from GPS and show the weather data. It can be further extend to view old weather data of the city and can also be used to store data in database and retrieve from that. It can also be extend to user favourite place weather data and any saved location data. For future, I will try to add these features to the application so that it became easy to use for daily users.

References

1. https://en.wikipedia.org/wiki/OpenWeatherMap
2. https://openweathermap.org/guide
3. https://www.guru99.com/c-sharp-tutorial.html
4. https://docs.microsoft.com/en-us/dotnet/csharp/
5. https://docs.microsoft.com/en-us/visualstudio/ide/step-1-create-a-windows-forms-application-project
6. https://www.c-sharpcorner.com/technologies/windows-forms
7. https://www.c-sharpcorner.com/article/json-serialization-and-deserialization-in-c-sharp/
8. https://stackoverflow.com/questions/2246694/how-to-convert-json-object-to-custom-c-sharp-object
9. https://www.nuget.org/packages/Newtonsoft.Json/