```csharp
using LiveCharts;
using LiveCharts.Wpf;
using System.Collections.Generic;
using System;
using System.Net.Http;
using System.Text.Json;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using WebBug;
using System.Linq;

namespace WebBug
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        string url = "https://data.moenv.gov.tw/api/v2/aqx_p_432?api_key=e8dd42e6-9b8b-43f8-991e-b3dee723a52d&limit=1000&sort=ImportDate%20desc&format=JSON";
        AQIdata aqidata = new AQIdata();
        List<Field> fields = new List<Field>();
        List<Record> records = new List<Record>();
        List<Record> selectedRecords = new List<Record>();
        SeriesCollection seriesCollection = new SeriesCollection();
        public MainWindow()
        {
            InitializeComponent();
            UrlTextBox.Text = url;
            selectedRecords.Clear();
        }

        private async void GetWebDataButton_Click(object sender, RoutedEventArgs e)
        {
            ContentTextBox.Text = "正在抓取網路資料...";

            string jsontext = await FetchContentAsync(url);
            ContentTextBox.Text = jsontext;
            aqidata = JsonSerializer.Deserialize<AQIdata>(jsontext);
            fields = aqidata.fields.ToList();
            records = aqidata.records.ToList();
            selectedRecords = records;
            StatusTextBlock.Text = $"共有 {records.Count} 筆資料";
            DisplayAQIData();
        }

        private void DisplayAQIData()
        {
            RecordDataGrid.ItemsSource = records;
            DataWrapPanel.Children.Clear();

            foreach (var field in fields)
            {
                var propertyInfo = typeof(Record).GetProperty
                  (field.id);
                if (propertyInfo != null)
                {
                    string value = propertyInfo.GetValue(records[0]) as
                   string;
                    if (double.TryParse(value, out double v))
                    {
                        CheckBox cb = new CheckBox
                        {
                            Content = field.info.label,
                            Tag = field.id,
                            Margin = new Thickness(3),
                            Width = 120,
                            FontSize = 14,
                            FontWeight = FontWeights.Bold,
                        };

                        cb.Checked += UpdateChart;
                        cb.Unchecked += UpdateChart;
                        DataWrapPanel.Children.Add(cb);
                    }
                }
            }
        }

        private void UpdateChart(object sender, RoutedEventArgs e)
        {
            seriesCollection.Clear();

            foreach (CheckBox cb in DataWrapPanel.Children)
            {
                if (cb.IsChecked == true)
                {
                    var tag = cb.Tag as String;
                    ColumnSeries columnSeries = new ColumnSeries();
                    ChartValues<double> values = new
                  ChartValues<double>();
                    List<String> labels = new List<String>();

                    foreach (var record in selectedRecords)
                    {
                        var propertyInfo = typeof(Record).GetProperty
                  (tag);
                        if (propertyInfo != null)
                        {
                            string value = propertyInfo.GetValue
                  (record) as string;
                            if (double.TryParse(value, out double v))
                            {
                                values.Add(v);
                                labels.Add(record.sitename);
                            }
                        }
                    }
                    columnSeries.Values = values;
                    columnSeries.Title = tag;
                    columnSeries.LabelPoint = point => $"{labels[(int)point.X]}: {point.Y.ToString()}";
                    seriesCollection.Add(columnSeries);
                }
            }
            AQIChart.Series = seriesCollection;
        }

        private async Task<string> FetchContentAsync(string url)
        {
            try
            {
                using (HttpClient client = new HttpClient())
                {
                    return await client.GetStringAsync(url);
                }
            }
            catch (Exception ex)
            {
                return $"發生錯誤: {ex.Message}";
            }
        }

        private void RecordDataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            selectedRecords = RecordDataGrid.SelectedItems.Cast<Record>().ToList();
            StatusTextBlock.Text = $"共選取 {selectedRecords.Count} 筆資料";
        }

        private void RecordDataGrid_LoadingRow(object sender, DataGridRowEventArgs e)
        {
            e.Row.Header = e.Row.GetIndex() + 1;
        }
    }
}
```