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;

using Newtonsoft.Json.Linq;

using System.Net;

using System.Runtime.InteropServices;

using Microsoft.Win32;

using System.Net.Http;

using meteo;

using System.Data.SQLite;

using System.Globalization;

using System.Messaging;

using Guna.UI2.WinForms;

using System.Windows.Forms.DataVisualization.Charting;

using Guna.Charts.WinForms;

using System.IO;

namespace meteo

{

public partial class mainForm : Form

{

private DateTime lastApiRequest = DateTime.Now;

public string data;

public mainForm()

{

InitializeComponent();

}

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

{

RequestApi();

tabDashboard();

timer1.Interval = 60000;

timer1.Tick += timer1\_Tick;

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

RequestApi();

}

public async void RequestApi()

{

guna2ProgressIndicator1.Start();

labelProcess.Text = "API request...";

var api = new NarodmonApi();

data = await api.GetData();

Console.WriteLine(data);

labelProcess.Text = "Success!";

radialGaugeFiller();

guna2ProgressIndicator1.Stop();

}

private void CsvTemp1()

{

string csvFilePath8622 = $"C:\\Users\\dyudy\\Desktop\\csharp3\\csv\\year\\D8622-20220702-20230406-1H.csv";

string connectionString = "Data Source=meteo.db;";

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

//temp1 aynabylak

using (StreamReader reader = new StreamReader(csvFilePath8622))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] fields = line.Split(';');

int unix = int.Parse(fields[0]);

string date = fields[1];

string time = fields[2];

double value = double.Parse(fields[3], CultureInfo.InvariantCulture);

using (SQLiteCommand command = new SQLiteCommand("INSERT INTO dataTemp1 (unix, date, time, value) VALUES (@unix, @date, @time, @value);", connection))

{

command.Parameters.AddWithValue("@unix", unix);

command.Parameters.AddWithValue("@date", date);

command.Parameters.AddWithValue("@time", time);

command.Parameters.AddWithValue("@value", value);

command.ExecuteNonQuery();

}

}

}

}

}

private void CsvToTable()

{

string csvFilePath9674 = $"C:\\Users\\dyudy\\Desktop\\csharp3\\csv\\year\\D9674-20230219-20230406-1H.csv";

string csvFilePath9737 = $"C:\\Users\\dyudy\\Desktop\\csharp3\\csv\\year\\D9737-20221206-20230406-1H.csv";

string connectionString = "Data Source=meteo.db;";

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

//temp2 stanitsa

using (StreamReader reader = new StreamReader(csvFilePath9737))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] fields = line.Split(';');

int unix = int.Parse(fields[0]);

string date = fields[1];

string time = fields[2];

double value = double.Parse(fields[3], CultureInfo.InvariantCulture);

using (SQLiteCommand command = new SQLiteCommand("INSERT INTO dataTemp2 (unix, date, time, value) VALUES (@unix, @date, @time, @value);", connection))

{

command.Parameters.AddWithValue("@unix", unix);

command.Parameters.AddWithValue("@date", date);

command.Parameters.AddWithValue("@time", time);

command.Parameters.AddWithValue("@value", value);

command.ExecuteNonQuery();

}

}

}

//temp3 orbita

using (StreamReader reader = new StreamReader(csvFilePath9674))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] fields = line.Split(';');

int unix = int.Parse(fields[0]);

string date = fields[1];

string time = fields[2];

double value = double.Parse(fields[3], CultureInfo.InvariantCulture);

using (SQLiteCommand command = new SQLiteCommand("INSERT INTO dataTemp3 (unix, date, time, value) VALUES (@unix, @date, @time, @value);", connection))

{

command.Parameters.AddWithValue("@unix", unix);

command.Parameters.AddWithValue("@date", date);

command.Parameters.AddWithValue("@time", time);

command.Parameters.AddWithValue("@value", value);

command.ExecuteNonQuery();

}

}

}

//pressure stanitsa

using (StreamReader reader = new StreamReader(csvFilePath9737))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] fields = line.Split(';');

int unix = int.Parse(fields[0]);

string date = fields[1];

string time = fields[2];

try

{

double value = double.Parse(fields[5], CultureInfo.InvariantCulture);

using (SQLiteCommand command = new SQLiteCommand("INSERT INTO dataPressure (unix, date, time, value) VALUES (@unix, @date, @time, @value);", connection))

{

command.Parameters.AddWithValue("@unix", unix);

command.Parameters.AddWithValue("@date", date);

command.Parameters.AddWithValue("@time", time);

command.Parameters.AddWithValue("@value", value);

command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

double value = 700;

}

}

}

//humadity stanitsa

using (StreamReader reader = new StreamReader(csvFilePath9737))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] fields = line.Split(';');

int unix = int.Parse(fields[0]);

string date = fields[1];

string time = fields[2];

double value = double.Parse(fields[4], CultureInfo.InvariantCulture);

using (SQLiteCommand command = new SQLiteCommand("INSERT INTO dataHumadity (unix, date, time, value) VALUES (@unix, @date, @time, @value);", connection))

{

command.Parameters.AddWithValue("@unix", unix);

command.Parameters.AddWithValue("@date", date);

command.Parameters.AddWithValue("@time", time);

command.Parameters.AddWithValue("@value", value);

command.ExecuteNonQuery();

}

}

}

}

}

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

{

db.WriteDB(data);

}

private void tabDashboard()

{

var api = new db();

// Создаем DataTable

DataTable dt = new DataTable();

// Добавляем столбцы в DataTable

dt.Columns.Add("Date", typeof(string));

dt.Columns.Add("Time", typeof(string));

dt.Columns.Add("Value", typeof(double));

dt.Columns.Add("Table", typeof(int));

var dsTemp1 = new Guna.Charts.WinForms.GunaAreaDataset();

dsTemp1.PointStyle = PointStyle.Dash;

dsTemp1.FillColor = Color.DodgerBlue;

var dsTemp2 = new Guna.Charts.WinForms.GunaAreaDataset();

dsTemp2.PointStyle = PointStyle.Dash;

dsTemp2.FillColor = Color.DeepSkyBlue;

var dsTemp3 = new Guna.Charts.WinForms.GunaAreaDataset();

dsTemp3.PointStyle = PointStyle.Dash;

dsTemp3.FillColor = Color.CadetBlue;

var dsPressure = new Guna.Charts.WinForms.GunaAreaDataset();

dsPressure.PointStyle = PointStyle.Dash;

var dsHumidity = new Guna.Charts.WinForms.GunaAreaDataset();

dsHumidity.PointStyle = PointStyle.Dash;

// Получаем данные из каждой таблицы и добавляем их в DataTable

List<Tuple<string, string, double>> data1 = api.ReadDB(1);

foreach (Tuple<string, string, double> row in data1)

{

dt.Rows.Add(row.Item1, row.Item2, row.Item3, 1);

dsTemp1.DataPoints.Add(row.Item1, row.Item3);

}

List<Tuple<string, string, double>> data2 = api.ReadDB(2);

foreach (Tuple<string, string, double> row in data2)

{

dt.Rows.Add(row.Item1, row.Item2, row.Item3, 2);

dsTemp2.DataPoints.Add(row.Item1, row.Item3);

}

List<Tuple<string, string, double>> data3 = api.ReadDB(3);

foreach (Tuple<string, string, double> row in data3)

{

dt.Rows.Add(row.Item1, row.Item2, row.Item3, 3);

dsTemp3.DataPoints.Add(row.Item1, row.Item3);

}

List<Tuple<string, string, double>> data4 = api.ReadDB(4);

foreach (Tuple<string, string, double> row in data4)

{

dt.Rows.Add(row.Item1, row.Item2, row.Item3, 4);

dsPressure.DataPoints.Add(row.Item1, row.Item3);

}

List<Tuple<string, string, double>> data5 = api.ReadDB(5);

foreach (Tuple<string, string, double> row in data5)

{

dt.Rows.Add(row.Item1, row.Item2, row.Item3, 5);

dsHumidity.DataPoints.Add(row.Item1, row.Item3);

}

// Привязываем DataTable к DataGridView

dataGridViewDashboard.DataSource = dt;

chartDashboard.Datasets.Add(dsTemp1);

chartDashboard.Datasets.Add(dsTemp2);

chartDashboard.Datasets.Add(dsTemp3);

//chartDashboard.Datasets.Add(dsPressure);

//chartDashboard.Datasets.Add(dsHumidity);

chartDashboard.Update();

}

private void tabTemp1()

{

var api = new db();

List<Tuple<string, string, double>> rows = api.ReadDB(1);

// Создание DataTable и добавление столбцов

DataTable table = new DataTable();

table.Columns.Add("Date", typeof(string));

table.Columns.Add("Time", typeof(string));

table.Columns.Add("Value", typeof(double));

var dataset = new Guna.Charts.WinForms.GunaAreaDataset();

dataset.PointStyle = PointStyle.Dash;

// Заполнение таблицы данными из списка Tuple

foreach (Tuple<string, string, double> row in rows)

{

table.Rows.Add(row.Item1, row.Item2, row.Item3);

dataset.DataPoints.Add(row.Item1, row.Item3);

}

// Установка DataTable в качестве источника данных для DataGridView

dataGridViewTemp1.DataSource = table;

//Add a new dataset to a chart.Datasets

chartTemp1.Datasets.Add(dataset);

//An update was made to re-render the chart

chartTemp1.Update();

}

private void tabTemp2()

{

var api = new db();

List<Tuple<string, string, double>> rows = api.ReadDB(2);

DataTable table = new DataTable();

table.Columns.Add("Date", typeof(string));

table.Columns.Add("Time", typeof(string));

table.Columns.Add("Value", typeof(double));

var dataset = new Guna.Charts.WinForms.GunaAreaDataset();

dataset.PointStyle = PointStyle.Dash;

foreach (Tuple<string, string, double> row in rows)

{

table.Rows.Add(row.Item1, row.Item2, row.Item3);

dataset.DataPoints.Add(row.Item1, row.Item3);

}

dataGridViewTemp2.DataSource = table;

chartTemp2.Datasets.Add(dataset);

chartTemp2.Update();

}

private void tabTemp3()

{

var api = new db();

List<Tuple<string, string, double>> rows = api.ReadDB(3);

DataTable table = new DataTable();

table.Columns.Add("Date", typeof(string));

table.Columns.Add("Time", typeof(string));

table.Columns.Add("Value", typeof(double));

var dataset = new Guna.Charts.WinForms.GunaAreaDataset();

dataset.PointStyle = PointStyle.Dash;

foreach (Tuple<string, string, double> row in rows)

{

table.Rows.Add(row.Item1, row.Item2, row.Item3);

dataset.DataPoints.Add(row.Item1, row.Item3);

}

dataGridViewTemp3.DataSource = table;

chartTemp3.Datasets.Add(dataset);

chartTemp3.Update();

}

private void tabPressure()

{

var api = new db();

List<Tuple<string, string, double>> rows = api.ReadDB(4);

DataTable table = new DataTable();

table.Columns.Add("Date", typeof(string));

table.Columns.Add("Time", typeof(string));

table.Columns.Add("Value", typeof(double));

var dataset = new Guna.Charts.WinForms.GunaAreaDataset();

dataset.PointStyle = PointStyle.Dash;

foreach (Tuple<string, string, double> row in rows)

{

table.Rows.Add(row.Item1, row.Item2, row.Item3);

dataset.DataPoints.Add(row.Item1, row.Item3);

}

dataGridViewPressure.DataSource = table;

chartPressure.Datasets.Add(dataset);

chartPressure.Update();

}

private void tabHumidity()

{

var api = new db();

List<Tuple<string, string, double>> rows = api.ReadDB(5);

DataTable table = new DataTable();

table.Columns.Add("Date", typeof(string));

table.Columns.Add("Time", typeof(string));

table.Columns.Add("Value", typeof(double));

var dataset = new Guna.Charts.WinForms.GunaAreaDataset();

dataset.PointStyle = PointStyle.Dash;

foreach (Tuple<string, string, double> row in rows)

{

table.Rows.Add(row.Item1, row.Item2, row.Item3);

dataset.DataPoints.Add(row.Item1, row.Item3);

}

dataGridViewHumidity.DataSource = table;

chartHumidity.Datasets.Add(dataset);

chartHumidity.Update();

}

private void guna2TabControl1\_Selected(object sender, TabControlEventArgs e)

{

if (tabControl.SelectedTab == tabPage1){tabDashboard();}

else if (tabControl.SelectedTab == tabPage2){tabTemp1();}

else if (tabControl.SelectedTab == tabPage3){tabTemp2();}

else if (tabControl.SelectedTab == tabPage4){tabTemp3();}

else if (tabControl.SelectedTab == tabPage5){tabPressure();}

else if (tabControl.SelectedTab == tabPage6){tabHumidity();}

}

private void radialGaugeFiller()

{

try

{

JObject json = JObject.Parse(data);

JArray sensors = (JArray)json["sensors"];

radialGaugeTemp1.Value = (int)sensors[0]["value"];

radialGaugeTemp2.Value = (int)sensors[1]["value"];

radialGaugeTemp3.Value = (int)sensors[2]["value"];

radialGaugePressure.Value = (int)sensors[3]["value"];

radialGaugeHumadity.Value = (int)sensors[4]["value"];

}

catch (Exception)

{

radialGaugeFiller();

}

}

private void gunaChart1\_Load(object sender, EventArgs e)

{

}

}

}