

```

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 System.Net;
using LOLTeamCounterPick.Classes;
using System.Text.RegularExpressions;
using System.IO;

namespace LOLTeamCounterPick.Components
{
    public partial class ProgressFRM : Form
    {
        List<string> HTMLSources;
        List<ChampionData> ChampionsData;
        public ProgressFRM()
        {
            InitializeComponent();
            HTMLSources = new List<string>();
            ChampionsData = new List<ChampionData>();
            this.Visible = false;
        }

        public void DownloadFromNet()
        {
            PB.Maximum = GVs.championNames.Count() * 3;
            LBL.Text = "Downloading Data From LOLCounter.com";
            //Application.DoEvents();
            this.Visible = true;
            Download();
            ImportHTMLSources();
            GVs.ChampionsData = ChampionsData;
            Export();
        }

        public void Import()
        {
            PB.Maximum = GVs.championNames.Count();
            LBL.Text = "Importing Data From Local Files";
            //Application.DoEvents();
            this.Visible = true;
            ImportLocalData();
            GVs.ChampionsData = ChampionsData;
        }

        private void Download()
        {
            if (HTMLSources != null) { HTMLSources.Clear(); }
            using (WebClient client = new WebClient()) // WebClient class inherits IDisposable
            {
                foreach (string name in GVs.championNames)
                {
                    PB.PerformStep();
                    //Application.DoEvents();
                    string nameFixed = Regex.Replace(name, @"^[^a-zA-Z]+", "");
                    string HTMLSourceStr = client.DownloadString(string.Format("http://www.lolcounter.com/
champions/{0}/strong", nameFixed));
                    HTMLSources.Add(HTMLSourceStr);
                }
            }
        }

        private void ImportHTMLSources()
        {
            foreach (string HTMLSource in HTMLSources)
            {
                PB.PerformStep();
                //Application.DoEvents();
                ChampionsData.Add(new ChampionData(HTMLSource, 1));
            }
        }
    }
}

```

```

private void Export()
{
    Microsoft.Office.Interop.Excel.Application xlApp = new Microsoft.Office.Interop.Excel.
Application();
    if (xlApp == null)
    {
        throw new Exception("EXCEL could not be started. Check that your office installation and
project references are correct.");
    }
    xlApp.DisplayAlerts = false;

    string currentWorkingPath = Directory.GetCurrentDirectory() + "\\Data";
    if (!Directory.Exists(currentWorkingPath)) { Directory.CreateDirectory(currentWorkingPath); }
    string workBookName;
    string workSheetName;
    Microsoft.Office.Interop.Excel.Workbook xlWorkBook;
    Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;

    foreach (ChampionData champ in ChampionsData)
    {
        PB.PerformStep();
        //Application.DoEvents();
        workBookName = champ.name;
        object misValue = System.Reflection.Missing.Value;
        xlWorkBook = xlApp.Workbooks.Add(misValue);
        xlWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
        workSheetName = "StrongAgainst";
        xlWorkSheet.Name = workSheetName;
        xlWorkSheet.Cells[1, 1] = "Name";
        xlWorkSheet.Cells[1, 2] = "VotedYes";
        xlWorkSheet.Cells[1, 3] = "VotedNo";
        xlWorkSheet.Cells[1, 4] = "Supportness";
        xlWorkSheet.Cells[1, 5] = "Correctness";
        xlWorkSheet.Cells[1, 6] = "Value";
        for (int i = 0; i < champ.Records.Count; i++)
        {
            xlWorkSheet.Cells[i + 2, 1] = champ.Records[i].name;
            xlWorkSheet.Cells[i + 2, 2] = champ.Records[i].votedYes;
            xlWorkSheet.Cells[i + 2, 3] = champ.Records[i].votedNo;
            xlWorkSheet.Cells[i + 2, 4] = champ.Records[i].supportness;
            xlWorkSheet.Cells[i + 2, 5] = champ.Records[i].correctness;
            xlWorkSheet.Cells[i + 2, 6] = champ.Records[i].value;
        }
        xlWorkBook.SaveAs(currentWorkingPath + string.Format("\\{0}.xls", workBookName),
Microsoft.Office.Interop.Excel.XlFileFormat.xlWorkbookNormal, misValue
, misValue, misValue, misValue,
Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive,
misValue, misValue, misValue, misValue, misValue);
        xlWorkBook.Close(true, misValue, misValue);

        releaseObject(xlWorkSheet);
        releaseObject(xlWorkBook);
    }
    xlApp.Quit();
    releaseObject(xlApp);
}
private void ImportLocalData()
{
    Microsoft.Office.Interop.Excel.Application xlApp = new Microsoft.Office.Interop.Excel.
Application();
    if (xlApp == null)
    {
        throw new Exception("EXCEL could not be started. Check that your office installation and
project references are correct.");
    }
    xlApp.DisplayAlerts = false;

    string currentWorkingPath = Directory.GetCurrentDirectory() + "\\Data";
    if (!Directory.Exists(currentWorkingPath)) { MessageBox.Show("Error: Local Data Imcomplete!\r\
nPlease Restart Program To Download Data To Local!"); }
    string workBookName;

```

```

    Microsoft.Office.Interop.Excel.Workbook xlWorkbook;
    Microsoft.Office.Interop.Excel.Worksheet xlWorksheet;

    foreach (string championName in GVs.championNames)
    {
        ChampionsData.Add(new ChampionData(championName, 0));
    }
    foreach (ChampionData champ in ChampionsData)
    {
        PB.PerformStep();
        //Application.DoEvents();
        workbookName = champ.name;
        object misValue = System.Reflection.Missing.Value;
        xlWorkbook = xlApp.Workbooks.Open(currentWorkingPath+"\"+workbookName+".xls");
        xlWorksheet = (Microsoft.Office.Interop.Excel.Worksheet)xlWorkbook.Worksheets.get_Item(1);
        Microsoft.Office.Interop.Excel.Range last = xlWorksheet.Cells.SpecialCells(Microsoft.Office.
Interop.Excel.XlCellType.xlCellTypeLastCell, Type.Missing);
        Microsoft.Office.Interop.Excel.Range range = xlWorksheet.get_Range("A1", last);
        Microsoft.Office.Interop.Excel.Range rangeToRead = xlWorksheet.UsedRange;
        int lastUsedRow = last.Row;
        int count = lastUsedRow - 1;

        for (int i = 0; i < count; i++)
        {
            string name = (string)(rangeToRead.Cells[i + 2, 1] as Microsoft.Office.Interop.Excel.
Range).Value2;
            double value = (double)(rangeToRead.Cells[i + 2, 6] as Microsoft.Office.Interop.Excel.
Range).Value2;
            champ.Records.Add(new ChampionDataRecord(name, value));
        }
        xlWorkbook.Close(true, misValue, misValue);

        releaseObject(xlWorksheet);
        releaseObject(xlWorkbook);

    }
    xlApp.Quit();
    releaseObject(xlApp);
}
private void releaseObject(object obj)
{
    try
    {
        System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
        obj = null;
    }
    catch (Exception ex)
    {
        obj = null;
        throw new Exception("Exception Occured while releasing object " + ex.ToString());
    }
    finally
    {
        GC.Collect();
    }
}
}
}

```