**SOURCE CODE**

1. Program.cs

using System;

namespace Assessment\_\_Phase\_1

{

class Program

{

/// <summary>

/// This is the main method where the program execution starts

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

int g = 1;

TeacherBo tbo = new TeacherBo();

//int cnt = 0;

while (g == 1)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.WriteLine("Hi, welcome to Little Flower School!!\nI am your friendly bot.\nHow can I help you with?");

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("Please select from the menu below, to perform action required :");

Console.WriteLine("1: Display all the Teachers ");

Console.WriteLine("2: Display the teachers by ID");

Console.WriteLine("3: Add a Teacher details ");

Console.WriteLine("4: Delete a Teacher details");

Console.WriteLine("5: Edit Teacher details");

Console.WriteLine("6: Exit\n");

int op = Convert.ToInt32(Console.ReadLine());

switch (op)

{

case 1:

tbo.Display();

break;

case 2:

Console.WriteLine("Enter the Id of Teacher:");

int id = Convert.ToInt32(Console.ReadLine());

Teachers it = tbo.getTeacherById(id);

Console.WriteLine(it.ToString());

break;

case 3:

Console.WriteLine("Enter Teacher's ID");

int tID = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the name: ");

string name = Console.ReadLine();

Console.WriteLine("Enter the class: ");

string clas = Console.ReadLine();

Console.WriteLine("Enter the section");

string section = Console.ReadLine();

tbo.addTeacher(tID, name, clas, section);

//tbo.Display();

break;

case 4:

Console.WriteLine("Enter the Id of Teacher to Delete it :");

int idToDelete = Convert.ToInt32(Console.ReadLine());

tbo.teacherDelete(idToDelete);

break;

case 5:

Console.WriteLine("Enter the ID of Teacher to edit the record:");

int idToEdit = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the updated name: ");

string ed\_name = Console.ReadLine();

Console.WriteLine("Enter the updated class: ");

string ed\_clas = Console.ReadLine();

Console.WriteLine("Enter the updated section");

string ed\_section = Console.ReadLine();

tbo.teacherEdit(idToEdit, ed\_name, ed\_clas, ed\_section);

break;

case 6:

g = 0;

break;

default:

Console.WriteLine("Please select the valid value.");

break;

}

}

FileOperations fp = new FileOperations();

//Console.WriteLine(t.ToString());

}

}

}

1. Teachers.cs

using System.Linq;

namespace Assessment\_\_Phase\_1

{

/// <summary>

/// this is the model class for data type Teacher

/// </summary>

class Teachers

{

public int id { get; set; }

public string name { get; set; }

public string clas { get; set; }

public string section { get; set; }

public Teachers(int id, string name, string clas, string section)

{

this.id = id;

this.name = name;

this.clas = clas;

this.section = section;

}

public override string ToString()

{

return $"Id : {id}, Name : {name}, Class : {clas}, Section : {section}";

}

}

}

1. TeacherBO.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace Assessment\_\_Phase\_1

{

/// <summary>

/// This class handles all the operations related to teacher in it

/// </summary>

class TeacherBo

{

public static List<Teachers> teacherList;

public TeacherBo()

{

teacherList = new List<Teachers>();

}

public Boolean addTeacher(int id, string name, string clas, string section)

{

Teachers t = new Teachers(id, name, clas, section);

teacherList.Add(t);

return true;

}

public void Display()

{

foreach (var item in teacherList)

{

Console.WriteLine(item.ToString());

}

}

public Teachers getTeacherById(int id)

{

return teacherList.Find(x => x.id == id);

}

public string teacherDelete(int id)

{

Teachers t = teacherList.Find(x => x.id == id);

teacherList.Remove(t);

return "Item Deleted!!";

}

public string teacherEdit(int id, string name, string clas, string section)

{

var index = teacherList.FindIndex(x => x.id == id);

if (index>-1)

{

teacherList[index].name = name;

teacherList[index].clas = clas;

teacherList[index].section = section;

}

else

{

Console.WriteLine("Record Not Found. Please enter it using the menu shown!!");

}

return "Item updated!!";

}

}

}

1. FileOperations.cs

using System;

using System.Collections.Generic;

using System.Text;

using System.IO;

namespace Assessment\_\_Phase\_1

{

/// <summary>

/// This class perform the write operation in the text file.

/// </summary>

class FileOperations

{

public FileOperations()

{

string fpath = @"C:\Users\11035915\source\repos\ProjectSolution\ProjectSolutionPhase1\Collection.txt";

FileStream fname = new FileStream(fpath, FileMode.Create, FileAccess.Write);

StreamWriter sw = new StreamWriter(fname);

sw.WriteLine("Teachers Records are as shown below!!\n");

List<Teachers> list = TeacherBo.teacherList;

foreach (var item in list)

{

sw.WriteLine($"{item.ToString()}");

sw.WriteLine("==========================================================================");

}

sw.Close();

}

}

}