|  |
| --- |
| **DAY 22 ASSIGNMENT**  **By**  **ARUN KUMAR YADLAPALLI**  **@**  **NB Healthcare Technologies PVT LTD.** |

|  |
| --- |
| **Q) Project of Employee Management App** |
| **Code:**  **DAL Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.IO;    namespace Data\_Access\_Library  {  //Author: Arun  //Purpose: To create a employee management app  public class Employee\_DAL  {  public static string filepath = "C:\\Users\\91832\\Desktop\\Arun\_Final\_Project\\File.txt.txt";  /// <summary>  /// This method is used to add employees  /// </summary>  /// <param name="empid"></param>  /// <param name="empname"></param>  /// <param name="empsalary"></param>  /// <param name="empage"></param>  /// <returns> Either True or False </returns>  public static bool Add\_Employee(int empid, string empname, int empsalary, int empage)  {  //Code which appends data is as follows:  try  {  string textcontent= string.Concat (empid, ",", empname, "," , empsalary, "," , empage);  File.AppendAllText(filepath, textcontent+Environment.NewLine );  return true;  }  catch (Exception ex)  {  return false;  }  }  /// <summary>  /// method searches for employee through id  /// </summary>  /// <param name="empid"></param>  /// <returns></returns>  public static List<string> Get\_Emp\_By\_Id(int empid)  {  var allEmployees=File.ReadAllLines (filepath);  bool isFound = false;  List<string> employees = new List<string> ();  foreach (string emp in allEmployees)  {  var employeeDetails = emp.Split(',');  if (Convert.ToInt32(employeeDetails[0]) == empid)  {  isFound = true;  employees.Add(emp);  break;  }    }  return employees;  }  /// <summary>  /// searching employees through name  /// </summary>  /// <param name="name"></param>  /// <returns>employees with the given name</returns>  public static List<string> Get\_Emp\_Byname(string name)  {  var allEmployees = File.ReadAllLines(filepath);  List<string> employees = new List<string>();  foreach (string emp in allEmployees)  {  var employeeDetails = emp.Split(',');  if (employeeDetails[1].Contains(name))  employees.Add(emp);      }  return employees;  }    public static String[] Display\_All\_Employees()  {  var allEmployees = File.ReadAllLines (filepath);  return allEmployees;  }  }  }  **BLL Code:**  using System;  using System.IO;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Data\_Access\_Library;      namespace Business\_Logic\_Layer  {  public static class Employee\_BLL  {  public static bool AddEmployee(int empid, string empname, int empsalary, int empage)  {  IsnegoRrepeat:  var st = Employee\_DAL.filepath;  bool verify = false;  int emp\_id;  var allEmployees = File.ReadAllLines(st);  foreach (var employee in allEmployees)  {  var emp\_split = employee.Split(',');  if (emp\_split[0] == empid.ToString())  {  verify = true;  break;  }  }  if (empid<=0||verify)  {  Console.WriteLine("Check the enter value greater than zero and number should not be repeated");  goto IsnegoRrepeat;  }    else  {  emp\_id = empid;  }  string emp\_name;  NameCheck:  if(empname.Length<3)  {  Console.WriteLine("Enter minimum # characters");  goto NameCheck;  }  else  {  emp\_name = empname;  }  int emp\_salary;  SalaryCheck:  if(empsalary<12000)  {  Console.WriteLine("Enter salary Above 12000");  goto SalaryCheck;  }  else  {  emp\_salary = empsalary;  }  int emp\_age;  AgeCheck:  if(empage<18||empage>58)  {  Console.WriteLine("Enter age above 18 and below 58");  goto AgeCheck;  }  else  {  emp\_age = empage;  }  // to do things  var result = Employee\_DAL.Add\_Employee(emp\_id, emp\_name, emp\_salary, emp\_age);  return result;        //all success then call Data access layer  }    public static List<string> Get\_Emp\_By\_Id(int id)  {  var result = Employee\_DAL.Get\_Emp\_By\_Id(id);  return result;  }    public static List<string> Get\_Emp\_Byname(string name)  {  var result= Employee\_DAL .Get\_Emp\_Byname(name);  return result;  }    public static string [] Display\_All\_Employees()  {  var result = Employee\_DAL .Display\_All\_Employees();  return result;  }  }  }  **ClientAPP Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using Business\_Logic\_Layer;    namespace My\_Client\_App  {  public class Program  {  public static void Add\_Employee()  {  int id, salary, age;  string name;  Console.WriteLine("enter id");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("enter name");  name = Console.ReadLine();  Console.WriteLine("enter salary");  salary = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("enter age ");  age = Convert.ToInt32(Console.ReadLine());    //calling BLL    var result = Employee\_BLL.AddEmployee(id, name, salary, age);  if(result)  {  Console.WriteLine("employee details saved");    }  else  {  Console.WriteLine("error occured");  }  }  public static void GetEmpById()  {  int id;  Console.WriteLine("Enter id");  id=Convert.ToInt32(Console.ReadLine());  var result=Employee\_BLL.Get\_Emp\_By\_Id(id);  if (result.Count == 0)  Console.WriteLine("No records");  else  result.ForEach(p => Console.WriteLine(p));  }  public static void GetEmpByName()  {  string name;  Console.WriteLine("Enter name");  name=Console.ReadLine();  var result=Employee\_BLL.Get\_Emp\_Byname(name);  if(result.Count==0)  Console.WriteLine("No name Found");    else  result.ForEach(p => Console.WriteLine(p));    }  public static void DisplayAllEmployees()  {  var result = Employee\_BLL.Display\_All\_Employees();  result.ToList().ForEach(r => Console.WriteLine(r));  }  static void Main(string[] args)  {  int ch;  string choice;  do  {  Console.WriteLine("Employee Management");  Console.WriteLine("1. Add Employee");  Console.WriteLine("2. search employee by id");  Console.WriteLine("3. search employee by name");  Console.WriteLine("4. display all employees");  Console.WriteLine("Enter your choice");  ch = Convert.ToInt32(Console.ReadLine());    switch (ch)  {  case 1:  Add\_Employee();  break;  case 2:  GetEmpById();  break;  case 3:  GetEmpByName();  break;  case 4:  DisplayAllEmployees();  break;  default:  Console.WriteLine("Invalid Input");  break;  }  Console.WriteLine("Do you want to Continue (y/n)");  choice = Console.ReadLine();  } while (choice.Equals("y"));  }  }  } |
| **Output Screenshots :** |