|  |
| --- |
| Day22 Morning Assignment  By  Anusha Bellala  22-02-2022 |

|  |
| --- |
| Write a C# Code for Employee Management Application. For Adding Employee, Search Employee & Display All Employees. |
| **Code:** |
| Creating a Solution name as **“AnushaFinalProject”** Which Consists of 3 Layers of  UI/Presentation Layer**(AnuClientApp** {ConsoleApp}**).**  Business Logic Layer **(BusinessLogicLibrary** {Class Library}**).**  Data Access Layer **(DataAccessLibrary** {Class Library}**).** |
| **DataAccessLibrary:** |
| using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DataAccessLibrary  {  public class EmployeeDAL  {  public static string filepath = "F:\\c# programs\\EmployeeData\\Employee.txt";  public static bool AddEmployee(int empid, string empname, int empsalary, int empage)  {  try  {  string textContent = String.Concat(empid, ",", empname, ",", empsalary, ",", empage);  File.AppendAllText(filepath, textContent + Environment.NewLine);  return true;  }  catch (Exception)  {  return false;  }  }  public static List<string> GetEmployeeById(int empid)  {  var allEmployees = File.ReadAllLines(filepath);  // bool isFound = false;  List<string> employeeFound = new List<string>();  foreach (string employee in allEmployees)  {  var empDetails = employee.Split(',');  if (Convert.ToInt32(empDetails[0]) == empid)  {  // isFound = true;  employeeFound.Add(employee);  break;  }  }  return employeeFound;  }  public static List<string> GetEmployeeByName(string empname)  {  var allEmployees = File.ReadAllLines(filepath);  //bool isFound = false;  List<string> employeeFound = new List<string>();  foreach (string employee in allEmployees)  {  var empDetails = employee.Split(',');  if (empDetails[1].Contains(empname))  {  employeeFound.Add(employee);  }  }  return employeeFound;  }  public static string[] GetAllEmployees()  {  var allEmployees = File.ReadAllLines(filepath);  return allEmployees;  }  }  } |

|  |
| --- |
| **BusinessLogicLibrary:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAccessLibrary;  namespace BusinessLogicLibrary  {  public class EmployeeBLL  {  public static bool AddEmployee(int empid, string empname, int empsalary, int empage)  {  //To Do Validations  var result = EmployeeDAL.AddEmployee(empid, empname, empsalary, empage);  return result;  }  public static List<string> GetEmployeeById(int empid)  {  var result = EmployeeDAL.GetEmployeeById(empid);  return result;  }  public static List<string> GetEmployeeByName(string empname)  {  var result = EmployeeDAL.GetEmployeeByName(empname);  return result;  }  public static string[] GetAllEmployees()  {  var result = EmployeeDAL.GetAllEmployees();  return result;  }  }  } |

|  |
| --- |
| **AnuClientApp:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BusinessLogicLibrary;  namespace AnuClientApp  {  internal class Program  {  static void Main(string[] args)  {  int ch;  string choice;  do  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("Employee Management Application");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  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:  AddEmployee();  break;  case 2:  SearchEmployeeById();  break;  case 3:  SearchEmployeeByName();  break;  case 4:  DisplayAllEmployees();  break;  default:  Console.WriteLine("Invalid Option");  break;  }  Console.WriteLine("Do you want to continue(Y/N):");  choice = Console.ReadLine();  } while (choice.Equals("Y"));  }  public static void AddEmployee()  {  int id, salary, age;  string name;  Console.WriteLine("Enter id:");  id = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter salary:");  salary = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter age:");  age = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter name:");  name = Console.ReadLine();  var result = EmployeeBLL.AddEmployee(id, name, salary, age);  if (result)  Console.WriteLine("Employee Details Saved Successfully");  else  Console.WriteLine("Some error occured");  }  public static void SearchEmployeeById()  {  int id;  Console.WriteLine("Enter Id:");  id = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.GetEmployeeById(id);  if (result.Count == 0)  Console.WriteLine("No records exists with this id");  else  result.ForEach(p => Console.WriteLine(p));  }  public static void SearchEmployeeByName()  {  string name;  Console.WriteLine("Enter name:");  name =Console.ReadLine();  var result = EmployeeBLL.GetEmployeeByName(name);  if(result.Count == 0)  Console.WriteLine("No Employee Details, Found with given name ");  else  result.ForEach(p => Console.WriteLine(p));  }  public static void DisplayAllEmployees()  {  var employees = EmployeeBLL.GetAllEmployees();  foreach (var employee in employees)  {  Console.WriteLine(employee);  }  }  }  } |
| Output: |