|  |
| --- |
| *Day 22 Assignment*  *By*  *M.Pallavi* |

|  |
| --- |
| Program: program to create Employee Management Application to  1.AddEmployee  2.search Employee.  3.Display all employee. |
| **Created solution as Final project which consists 3 layers.**  **clientApp,BusinesslogicLibrary, DLLlibrary**. |
| **Data Link Library:**  Code:  using System;  using System.Collections.Generic;  using System.Linq;  using System.IO;  namespace DataAcessLayer  {  public static class EmployeeDAL  {  public static string filepath = "D:\\Day1 Assessment by Pallavi Mechineni\\Day 22 project\\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 ex)  {  return false;  }  }  public static List<string> GetEmployeeId(int id)  {  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]) == id)  {  //isFound = true;  employeeFound.Add(employee);  break;  }  }  return employeeFound;  }  public static List<string> GetEmployeeName(string name)  {  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(name))  {  employeeFound.Add(employee);  }  }  return employeeFound;  }  public static string[] GetAllEmployees()  {    var allemployees = File.ReadAllLines(filepath);  return allemployees;  }  }  } |
| Output: |
| Business logic Library:  using System;  using DataAcessLayer;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace BusinessLogicLibrary  {  public class EmployeeBLL  {  public static bool AddEmployee(int empid, string empname, int empsalary, int empage)  {  var result = EmployeeDAL.AddEmployee(empid, empname, empsalary, empage);  return result;  }  public static List<string> GetEmployeeId(int id)  {  var result = EmployeeDAL.GetEmployeeId(id);  return result;  }  public static List<string> GetEmployeeName(string name)  {  var result = EmployeeDAL.GetEmployeeName(name);  return result;  }  public static string[] GetAllEmployees()  {  var result = EmployeeDAL.GetAllEmployees();  return result;  }  }  } |
|  |
| ConsoleApplication Code:  using System;  using BusinessLogicLibrary;  namespace ClientApp  {  public class  {  public static void Main(string[] args)  {  string choice;  int ch;  do  {  Console.WriteLine("Employee Management Application");  Console.WriteLine("\t 1. Add Employee Details");  Console.WriteLine("\t 2. get all employee details");  Console.WriteLine("\t 3. Search Employee By ID");  Console.WriteLine("\t 4. Search Employee by Name");  Console.Write("\n Enter Your Choice : ");  ch = Convert.ToInt32(Console.ReadLine());  switch (ch)  {  case 1: AddEmployee();  break;  case 2: GetAllEmployees();  break;  case 3: GetEmployeeById();  break;  case 4: GetEmployeeByName();  break;  default:Console.WriteLine("Invalid Option");  break;  }  Console.Write("\nDo You Wish to Continue to Main Menu (y/n): ");    choice = Console.ReadLine();    } while (choice == "y" || choice == "Y");  Console.ReadLine();  }  public static void AddEmployee()  {  int id, salary, age;  string name;  Console.Write("\nEnter the Employee Id : ");  id = Convert.ToInt32(Console.ReadLine());  Console.Write("\nEnter the Employee Name : ");  name = Console.ReadLine();  Console.Write("\nEnter the Employee Salary : ");  salary = Convert.ToInt32(Console.ReadLine());  Console.Write("\nEnter the Employee Age : ");  age = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.AddEmployee(id, name, salary, age);  if (result)  {  Console.WriteLine("\n Employee Details Saved Succesfully");  }  else  {  Console.WriteLine("\n Error occured During Saving The Given Data, Please Try Again");  }  }  public static void GetEmployeeById()  {  int id;  Console.Write("\nEnter any ID to Search Employee\_Details\_By\_ID : ");  id = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.GetEmployeeId(id);  if (result.Count == 0)  {    Console.WriteLine("No Records, Found With This ID");    }  else  {  Console.WriteLine("The Employee Details For Given Id :");    result.ForEach(e => Console.WriteLine("\t{0}", e));    }  }  public static void GetEmployeeByName()  {  string name;  Console.Write("\nEnter any NAME, To Search\_Employee\_Details\_By\_Name : ");  name = Console.ReadLine();  var result = EmployeeBLL.GetEmployeeName(name);  if (result.Count == 0)  {    Console.WriteLine("No Employee Details, Found with Given Name '{0}' ", name);  }  else  {  Console.WriteLine("The Employees, whose names Consists '{0}' are : \n", name);    result.ForEach(emp => Console.WriteLine("\t{0}", emp));    }  }  public static void GetAllEmployees()  {  var employees = EmployeeBLL.GetAllEmployees();    foreach (var employee in employees)  {  Console.WriteLine(employee);  }    }  }  } |
| Output: |