|  |
| --- |
| **Day -22 Morning Assignments**  **By**  **Manoj Yekolla**  **22-Feb-2022** |

|  |
| --- |
| **1.Write a C# Code for Employee Management Application.** |
| **Code :** |
| * **For Adding Employee ,Searching Employee and Display All Employee Details** |
| * Solution Name : FinalProjectManoj * UI/Presentantion Layer : ConsoleApp1 ( Console App) * Bussiness Logic Layer : BussinessLogicLayer( Class Library1) * Data Access Layer : DataAcessesLayer ( Class Library2) |
| Screenshot (369) |
| **Bussiness Logic Layer** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAcessesLayer;  namespace BussinessLogicLayer  {  public class EmployeeBLL  {  public static bool AddEmployee(int empId, string empName, int empSalary, int  empAge)  {  // Todo Validations  var result = DALEmployee.AddEmployee(empId, empName, empSalary, empAge);  return result;  }  public static List<String> GetEmployeesById(int id)  {  var result = DALEmployee.GetEmployeesById(id);  return result;  }  public static List<String> GetEmployeesByName(string name)  {  var result = DALEmployee.GetEmployeesByName(name);  return result;  }  public static string[] GetAllEmployees()  {  var result = DALEmployee.GetAllEmployees();  return result;  }  }  } |
| **Data Accesses Layer :** |
| Code : |
| using System;  using System.IO;  using System.Collections.Generic;  namespace DataAcessesLayer  {  public static class DALEmployee  {  public static string filePath = @"F:\Nb HealthCare Tech\Day-22 Morning Assignments\EmployeesData\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> GetEmployeesById(int id)  {  var allEmployees = File.ReadAllLines(filePath);  //bool isFound;  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> GetEmployeesByName(string name)  {  var allEmployees = File.ReadAllLines(filePath);  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 allEmplyees = File.ReadAllLines(filePath);  return allEmplyees;  }  }  } |
| **Ui/Presentation Layer:ConsoleApp1** |
| using System;  using BussinessLogicLayer;  namespace ConsoleApp1  {  public class Program  {  public static void Main(string[] args)  {  string choice;  int ch;  do  {  Console.WriteLine("--------------------------------------------");  Console.WriteLine("Welecome to Employement Management Application");  Console.WriteLine("------------------------------------------------");  Console.WriteLine(" 1. Add Employee Details");  Console.WriteLine(" 2. Search Employee By Id");  Console.WriteLine(" 3. Search Employee By Name");  Console.WriteLine(" 4. Display All Employee Details");  Console.Write(" 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.Write("Do You Want to Continue (y/n): ");    choice = Console.ReadLine();    } while (choice == "y" || choice == "Y");      Console.ReadLine();  }  public static void AddEmployee()  {  int id, salary, age;  string name;  Console.Write("Enter employee Id : ");  id = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter employee Name : ");  name = Console.ReadLine();  Console.Write("Enter employee Salary : ");  salary = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter employee Age : ");  age = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.AddEmployee(id, name, salary, age);  if (result)  {    Console.WriteLine(" Employee Details Saved Succesfully");    }  else  {  Console.WriteLine("Something went worng ");  }  }  public static void SearchEmployeeById()  {  int id;  Console.Write("Enter your Id : ");  id = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.GetEmployeesById(id);  if (result.Count == 0)  {    Console.WriteLine(" No Records Found");    }  else  {  Console.WriteLine(" Successfull saved Id details : ");    result.ForEach(e => Console.WriteLine("\t{0}", e));    }  }  public static void SearchEmployeeByName()  {  string name;  Console.Write("Enter your Name : ");  name = Console.ReadLine();  var result = EmployeeBLL.GetEmployeesByName(name);  if (result.Count == 0)  {  Console.WriteLine("No Records are Found");      }  else  {  Console.WriteLine("Display your Name ");      result.ForEach(emp => Console.WriteLine( emp));    }  }  public static void DisplayAllEmployees()  {  var employees = EmployeeBLL.GetAllEmployees();    Console.WriteLine(" Display All Employee Deatials");      foreach (var employee in employees)  {  Console.WriteLine(employee);  }    }  }  } |
| **Output :** |
| Screenshot (377) |
| Screenshot (378) |
| Screenshot (379) |
| Screenshot (380) |
| **The End** |