|  |
| --- |
| DAY 22 ASSIGNMENTS  BY  LOKESH NADELLA  22-02-2022 |

|  |
| --- |
| EmployeeBll::: |
| CODE: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAcessLayer;  namespace BusinessLogicLibrary  {  public class EmployeeBll  {  public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  var result = EmployeeDAL1.AddEmployee(empId, empName, empSalary, empAge);  return result;  }  public static List<string> GetEmployeeById(int id)  {  var result = EmployeeDAL1.GetEmployeeById(id);  return result;  }  public static List<string> GetEmployeeByname(string name)  {  var result = EmployeeDAL1.GetEmployeeByname(name);  return result;  }  public static string[] GetallEmployees()  {  var result = EmployeeDAL1.GetallEmployees();  return result;  }  }  } |
|  |
| EmployeeDAL1:: |
| CODE:: |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DataAcessLayer  {  public class EmployeeDAL1  {  public static string filepath = " C:\\EmployeeData\\Employees.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> GetEmployeeById(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> GetEmployeeByname(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;  }  }  } |
|  |
| MyClientApp:: |
| Code::: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BusinessLogicLibrary;  namespace MyClientApp  {  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 Employee:");  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 records exists with this name");  else  {  result.ForEach(p => Console.WriteLine(p));  }  }  public static void DisplayAllEmployees()  {  var result = EmployeeBll.GetallEmployees();  result.ToList().ForEach(p => Console.WriteLine(p));  }  }  } |

|  |
| --- |
| OUTPUT::: |
| TEXTFILE:: |
|  |
| Add Employee::: |
|  |
|  |
| Search Employee by id:: |
|  |
|  |
|  |
| Search employee by name:: |

|  |
| --- |
|  |
| DISPLAY ALL EMPLOYEE:: |
|  |
|  |