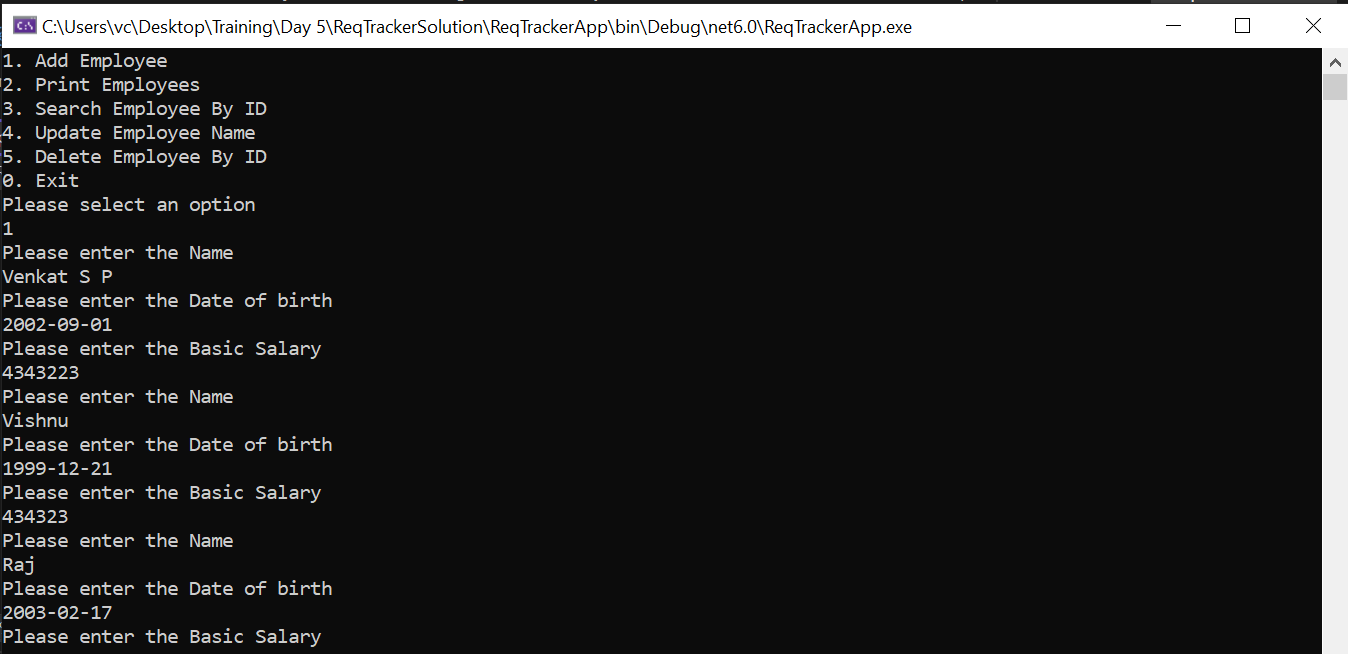
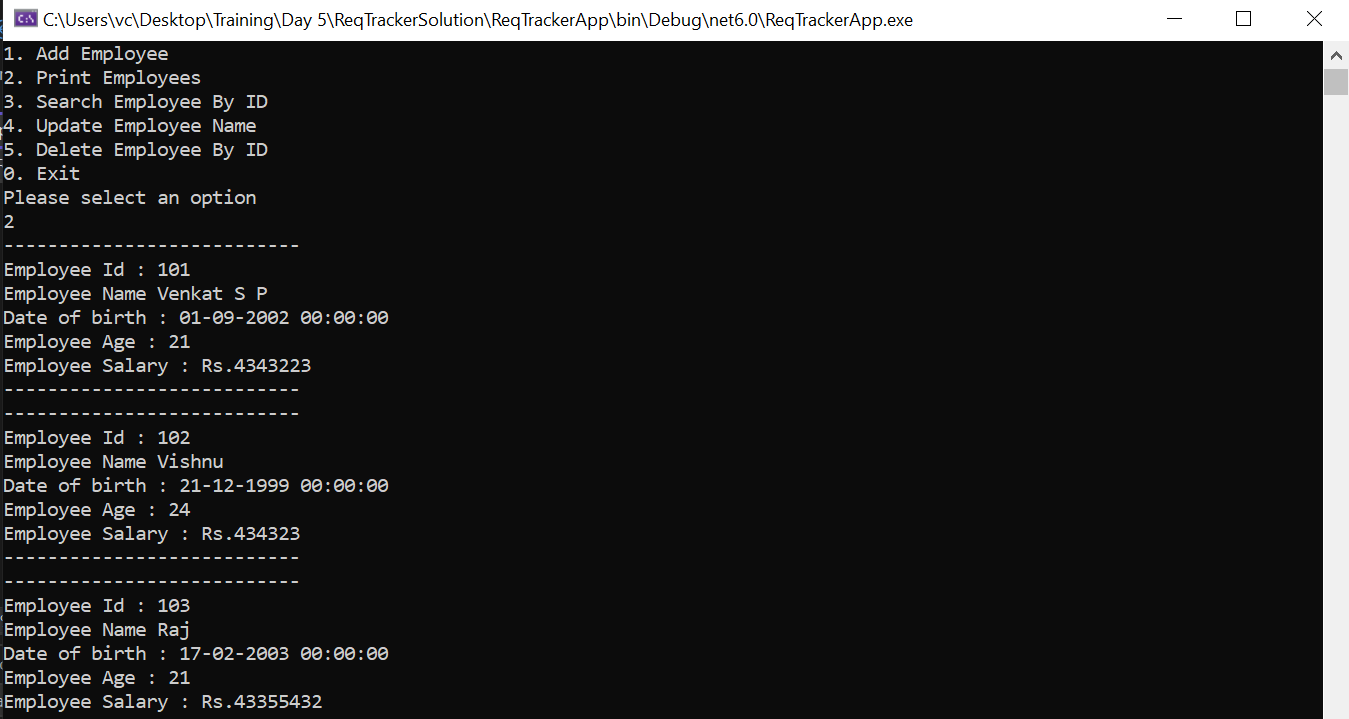
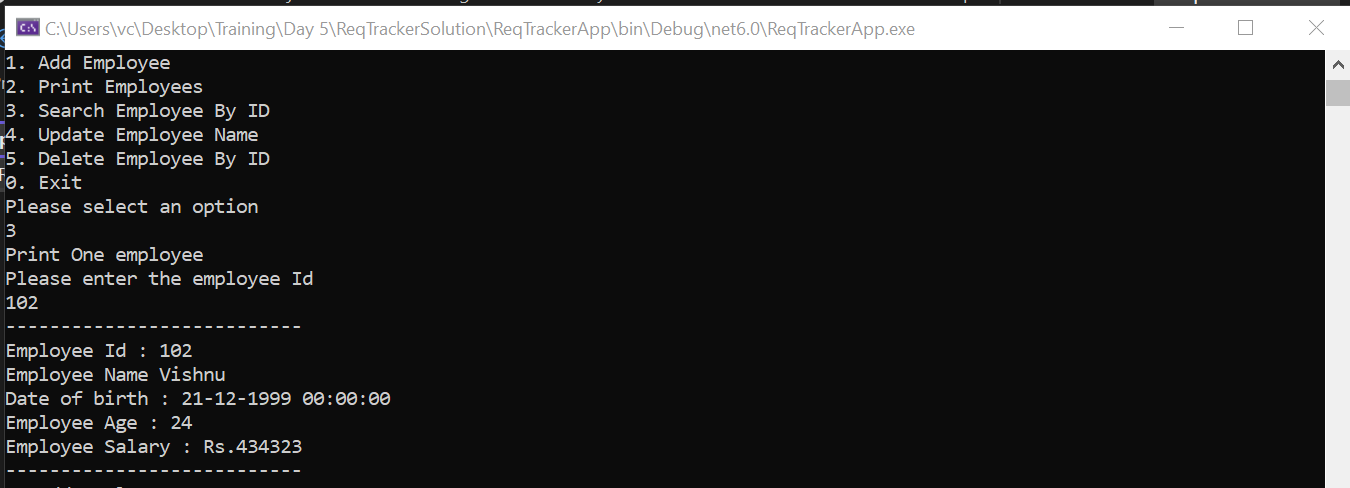
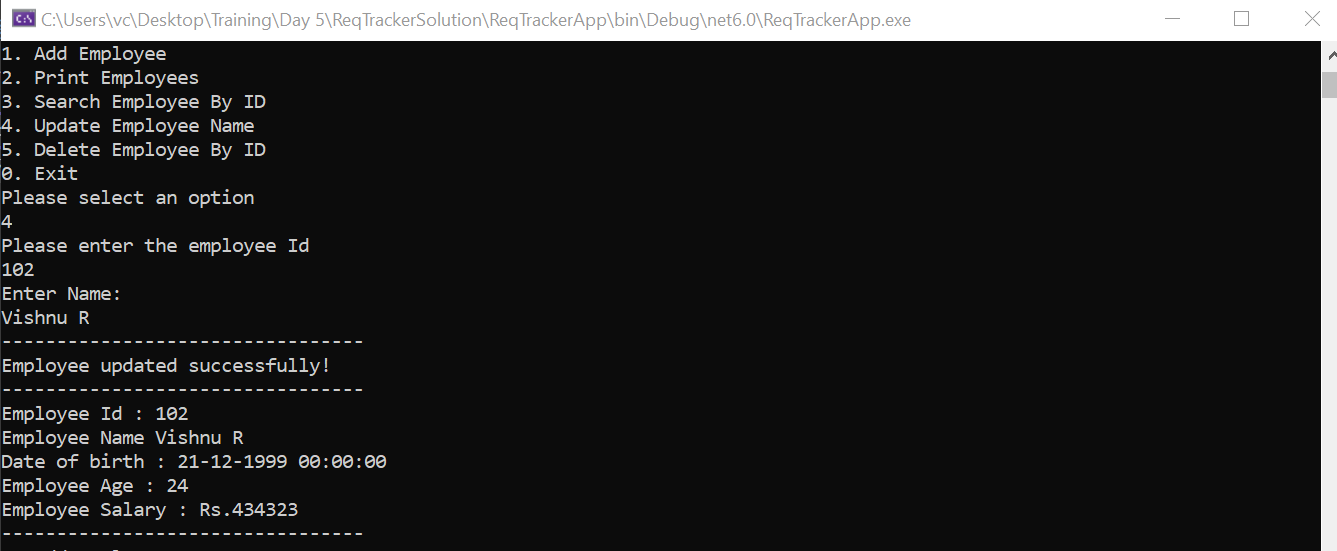
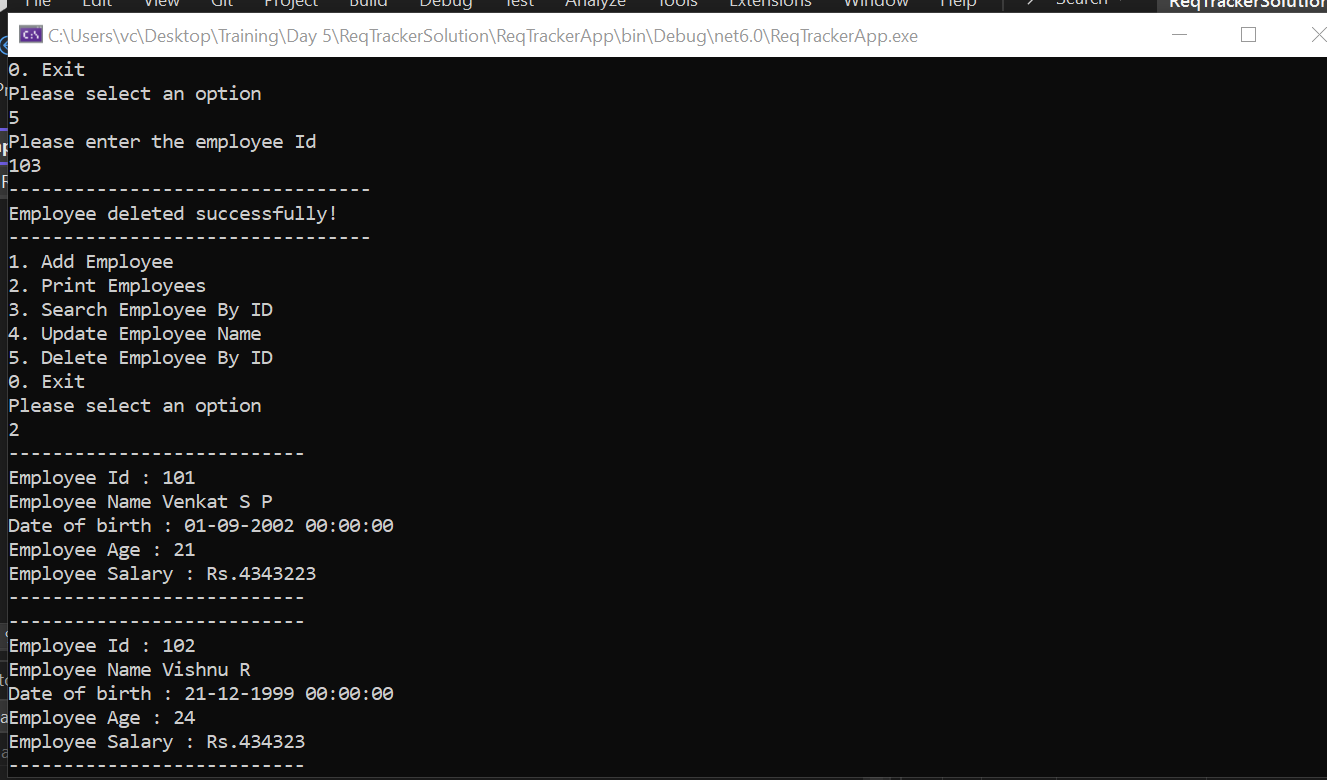
**OUTPUT:**











**CODE:**

using ReqTrackerClass;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ReqTrackerApp

{

internal class EmployeeApp

{

Employee[] employees;

/// <summary>

/// Contructor to create a array of 3 Employees

/// </summary>

public EmployeeApp()

{

employees = new Employee[3];

}

/// <summary>

/// Function to print Menu for Operations

/// </summary>

void PrintMenu()

{

Console.WriteLine("1. Add Employee");

Console.WriteLine("2. Print Employees");

Console.WriteLine("3. Search Employee By ID");

Console.WriteLine("4. Update Employee Name");

Console.WriteLine("5. Delete Employee By ID");

Console.WriteLine("0. Exit");

}

/// <summary>

/// Switch Case to Decide the Operation to be done

/// </summary>

void EmployeeInteraction()

{

int choice = 0;

do

{

PrintMenu();

Console.WriteLine("Please select an option");

choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 0:

Console.WriteLine("Bye.....");

break;

case 1:

AddEmployee();

break;

case 2:

PrintAllEmployees();

break;

case 3:

SearchAndPrintEmployee();

break;

case 4:

UpdateEmployeeName();

break;

case 5:

DeleteEmployee();

break;

default:

Console.WriteLine("Invalid choice. Try again");

break;

}

} while (choice != 0);

}

/// <summary>

/// Function to add employee

/// </summary>

void AddEmployee()

{

if (employees[employees.Length - 1] != null)

{

Console.WriteLine("Sorry we have reached the maximum number of employees");

return;

}

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] == null)

{

employees[i] = CreateEmployee(i);

}

}

}

/// <summary>

/// Function to print all employees

/// </summary>

void PrintAllEmployees()

{

if (employees[0] == null)

{

Console.WriteLine("No Employees available");

return;

}

for (int i = 0; i < employees.Length; i++)

{

if (employees[i] != null)

PrintEmployee(employees[i]);

}

}

/// <summary>

/// Function to get input string

/// </summary>

/// <param name="field">Field params as (string)</param>

/// <returns></returns>

string GetStringInp(string field)

{

Console.WriteLine($"Enter {field}:");

string inp;

do

{

inp = Console.ReadLine();

if (string.IsNullOrEmpty(inp))

{

Console.WriteLine($"Invalid {field} entry");

}

} while (string.IsNullOrEmpty(inp));

return inp;

}

/// <summary>

/// Function to create employee

/// </summary>

/// <param name="id">Id as integer</param>

/// <returns></returns>

Employee CreateEmployee(int id)

{

Employee employee = new Employee();

employee.Id = 101 + id;

employee.BuildEmployeeFromConsole();

return employee;

}

/// <summary>

/// Print a particular employee detail

/// </summary>

/// <param name="employee">Employee obj as params</param>

void PrintEmployee(Employee employee)

{

Console.WriteLine("---------------------------");

employee.PrintEmployeeDetails();

Console.WriteLine("---------------------------");

}

/// <summary>

/// Get Id from console

/// </summary>

/// <returns></returns>

int GetIdFromConsole()

{

int id = 0;

Console.WriteLine("Please enter the employee Id");

while (!int.TryParse(Console.ReadLine(), out id))

{

Console.WriteLine("Invalid entry. Please try again");

}

return id;

}

/// <summary>

/// Function to search and print employee

/// </summary>

void SearchAndPrintEmployee()

{

Console.WriteLine("Print One employee");

int id = GetIdFromConsole();

Employee employee = SearchEmployeeById(id);

if (employee == null)

{

Console.WriteLine("No such Employee is present");

return;

}

PrintEmployee(employee);

}

/// <summary>

/// Function to search employee By Id

/// </summary>

/// <param name="id"></param>

/// <returns></returns>

Employee SearchEmployeeById(int id)

{

Employee employee = null;

return employees[id-101] != null ? employees[id-101] : employee;

}

/// <summary>

/// Function to Update employee Name

/// </summary>

void UpdateEmployeeName()

{

int id = GetIdFromConsole ();

Employee emp = SearchEmployeeById(id);

string Name = GetStringInp("Name");

emp.Name = Name;

Console.WriteLine("---------------------------------");

Console.WriteLine("Employee updated successfully!");

Console.WriteLine("---------------------------------");

emp.PrintEmployeeDetails ();

Console.WriteLine("---------------------------------");

}

/// <summary>

/// Function to Delete employee

/// </summary>

void DeleteEmployee()

{

int id = GetIdFromConsole();

employees[id - 101] = null;

Console.WriteLine("---------------------------------");

Console.WriteLine("Employee deleted successfully!");

Console.WriteLine("---------------------------------");

}

static void Main(string[] args)

{

EmployeeApp program = new EmployeeApp();

program.EmployeeInteraction();

}

}

}