COLLEGE OF APPLIED BUSINESS AND TECHNOLOGY

Kathmandu, Nepal

Report of partial fulfillment of

CSC 367: NET Centric Computing

PRACTICAL EXAM-2080

Submitted To:

Narayan Adhikari

Department of Computer Science and Information Technology

College of Applied Business and Technology

Submitted By:

Rabindra Adhikari

Roll No.: 23571

September 10, 2023

Write a program to show class, constructor, properties and method

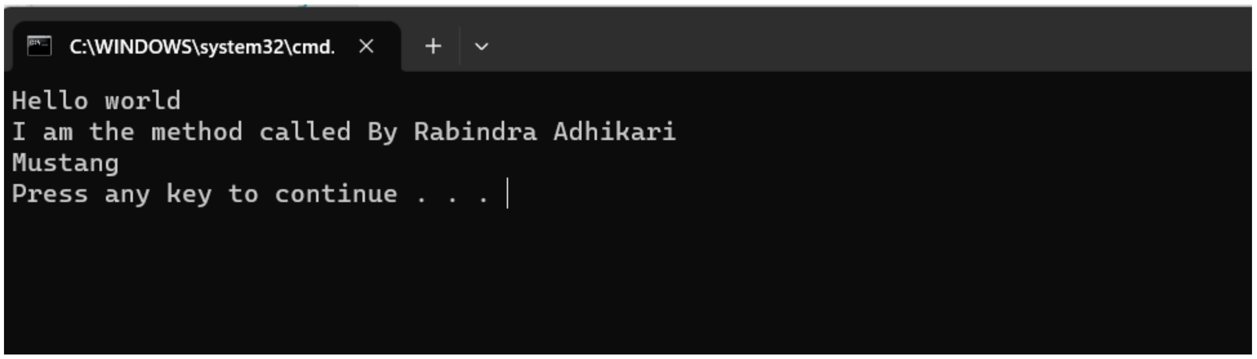
Car.cs

```
namespace Practical
{
internal class Car
{
public string model;  // Create a field
// Create a class constructor for the Car class
public Car()
{
model = "Mustang"; // Set the initial value for model
}
}
}
```

Program.cs

```
using System;
namespace Practical
{
internal class Program
{
static void Main(string[] args)
{
Console.WriteLine("Hello world");
//Method Call in main Method
method();
//Object of Car
Car car= new Car();
Console.WriteLine(car.model);
}
//Method
static void method()
{
Console.WriteLine("I am the method called By Rabindra Adhikari");
}
}
}
```

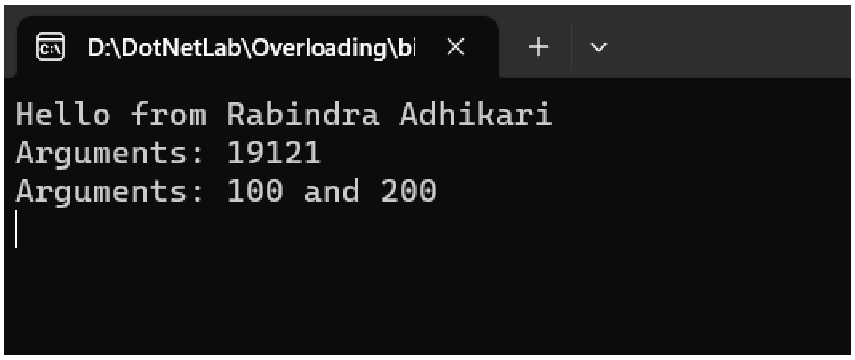
Output



```
Hello world
I am the method called By Rabindra Adhikari
Mustang
Press any key to continue . . .
```

Write a program to demonstrate method overloading

```
using System;

namespace MethodOverload
{
class Program
{
void display()
{
System.Console.WriteLine("Hello from Rabindra Adhikari");
}
// method with one parameter
void display(int a)
{
Console.WriteLine("Arguments: " + a);
}

// method with two parameters
void display(int a, int b)
{
Console.WriteLine("Arguments: " + a + " and " + b);
}
static void Main(string[] args)
{

Program p1 = new Program();
p1.display();
p1.display(19121);
p1.display(100, 200);

Console.ReadLine();
}
}
}
```

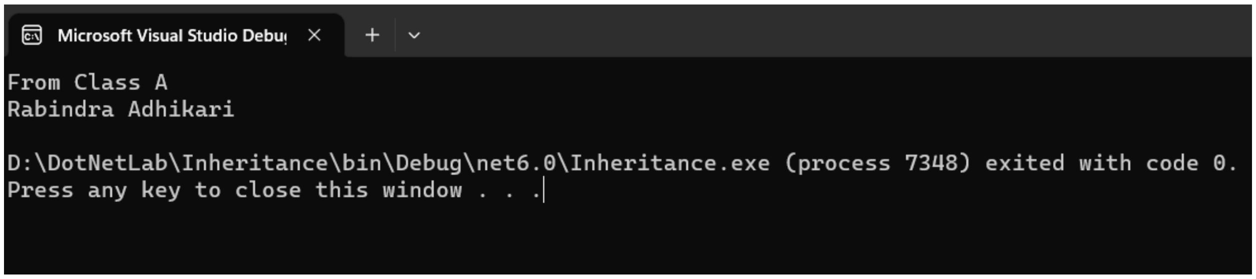


```
Hello from Rabindra Adhikari
Arguments: 19121
Arguments: 100 and 200
```

Write a program to demonstrate single level inheritance and multilevel inheritance

Single level inheritance

```
public class A
{
public void Method1()
{
Console.WriteLine("From Class A ");
Console.WriteLine("Rabindra Adhikari");
}
}

public class B : A
{ }

public class Example
{
public static void Main()
{
B b = new();
b.Method1();
}
}
```

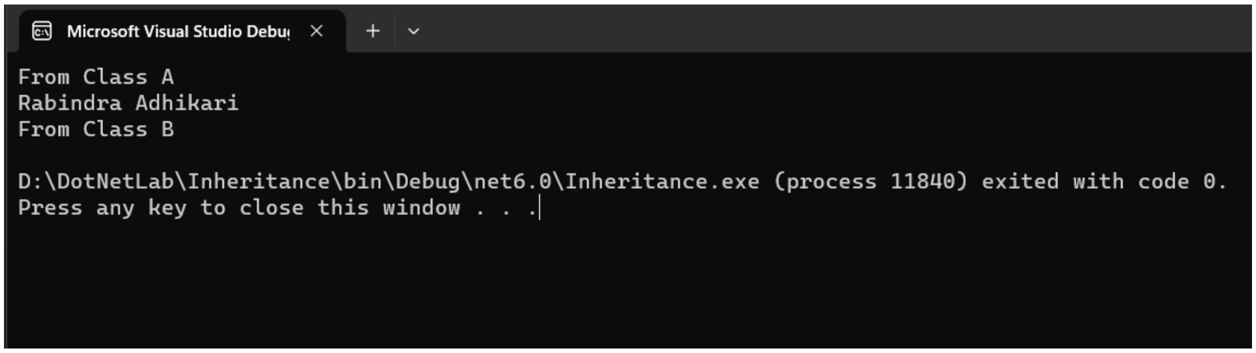


```
From Class A
Rabindra Adhikari

D:\DotNetLab\Inheritance\bin\Debug\net6.0\Inheritance.exe (process 7348) exited with code 0.
Press any key to close this window . . .
```

Multi Level Inheritance

```
public class A
{
public void Method1()
{
Console.WriteLine("From Class A ");
Console.WriteLine("Rabindra Adhikari");
}
}

public class B : A
{
public void Method2()
{
Console.WriteLine("From Class B");
}

}

public class Example
{
public static void Main()
{
```

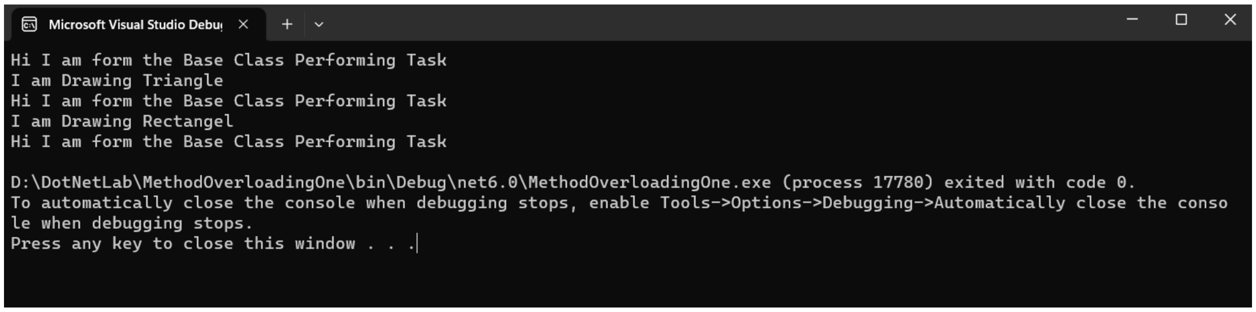
```
B b = new();

b.Method1();

b.Method2();

}

}
```



```
From Class A
Rabindra Adhikari
From Class B

D:\DotNetLab\Inheritance\bin\Debug\net6.0\Inheritance.exe (process 11840) exited with code 0.
Press any key to close this window . . .
```

Write a program to demonstrate method overriding condition

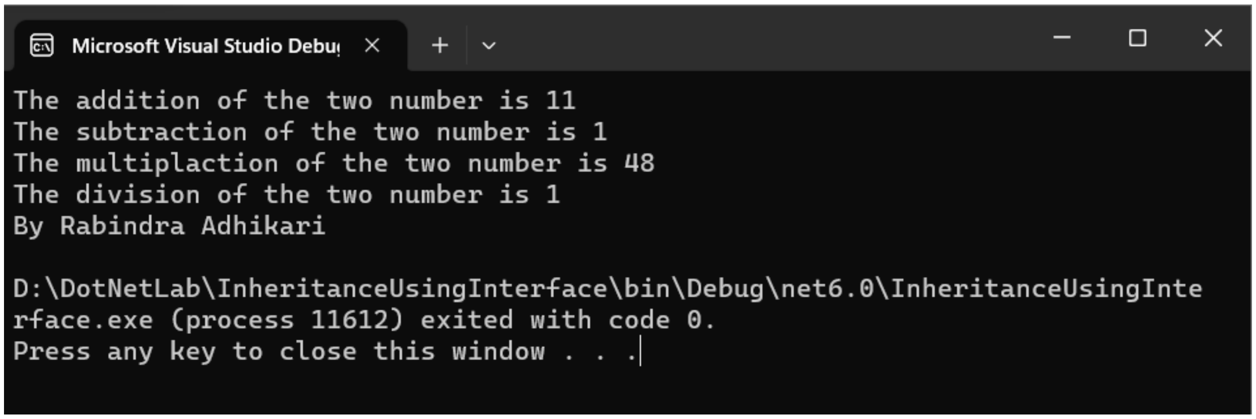
```
public class MethodOverloadingOne {
public class Shape
{
public virtual void Draw()
{
Console.WriteLine("Hi I am form the Base Class Performing Task");
}
}
public class Triange : Shape
{
public override void Draw()
{
Console.WriteLine("I am Drawing Triangle");
base.Draw();
}
}
public class Rectangle : Shape
{
public override void Draw()
{
Console.WriteLine("I am Drawing Rectangel");
base.Draw();
}
}
public static void Main(string[] args)
{
Shape s = new Shape();
s.Draw();
Triange triange = new Triange();
triange.Draw();
Rectangle rect = new Rectangle();
rect.Draw();


}

}
```



```
Hi I am form the Base Class Performing Task
I am Drawing Triangle
Hi I am form the Base Class Performing Task
I am Drawing Rectangel
Hi I am form the Base Class Performing Task

D:\DotNetLab\MethodOverloadingOne\bin\Debug\net6.0\MethodOverloadingOne.exe (process 17780) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Write a program to demonstrate multiple inheritance using interface

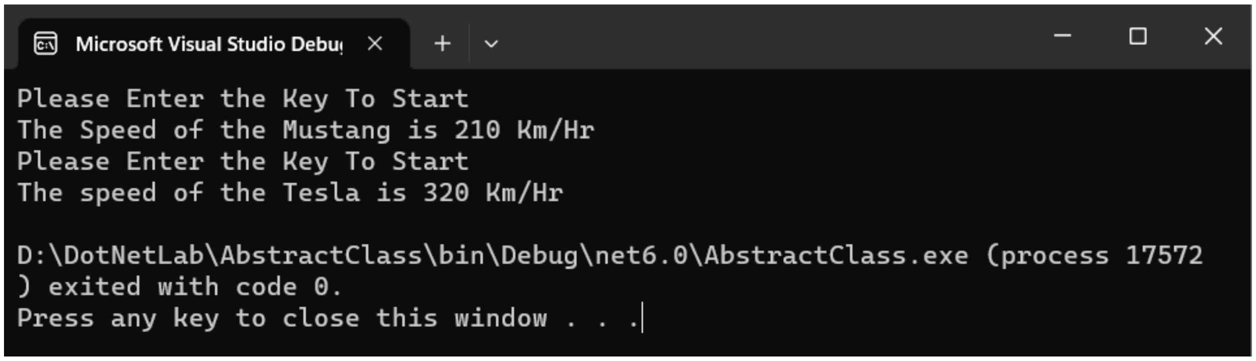
```
public interface IAdd
{
void Add(int a, int b);
}

public interface Isub
{
void Sub(int a, int b);
}

public interface Imul
{
void Mul(int a, int b);
}

public interface Idiv
{
void Div(int a, int b);
}
public class Calculator : IAdd, Isub, Imul, Idiv
{
public int resultAdd;
public int resultSub;
public int resultMul;
public int resultDiv;

public void Add(int a, int b)
{
resultAdd = a + b;
}

public void Sub(int a, int b)
{
resultSub = a - b;
}
public void Mul(int a, int b)
{
resultMul = a * b;
}

public void Div(int a, int b)
{
resultDiv = a / b;
}


public class MainClass
{
static void Main()
{
Calculator calculator = new Calculator();
calculator.Add(5, 6);
calculator.Sub(7, 6);
calculator.Mul(8, 6);
calculator.Div(9, 6);

Console.WriteLine("The addition of the two number is " + calculator.resultAdd);
Console.WriteLine("The subtraction of the two number is " + calculator.resultSub);
Console.WriteLine("The multiplaction of the two number is " + calculator.resultMul);
Console.WriteLine("The division of the two number is " + calculator.resultDiv);
Console.WriteLine("By Rabindra Adhikari");



}
}
}
```



```
The addition of the two number is 11
The subtraction of the two number is 1
The multiplaction of the two number is 48
The division of the two number is 1
By Rabindra Adhikari

D:\DotNetLab\InheritanceUsingInterface\bin\Debug\net6.0\InheritanceUsingInte
rface.exe (process 11612) exited with code 0.
Press any key to close this window . . .
```

Write a program to demonstrate abstract class

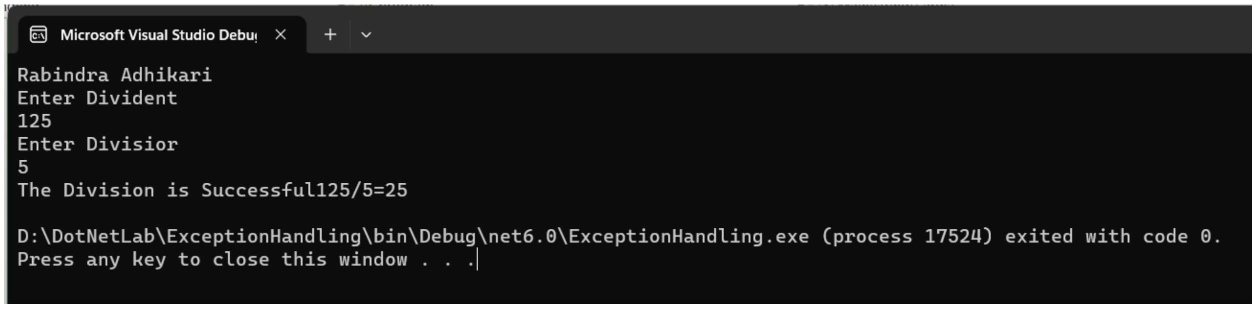
```
abstract class Vehical
{
public abstract void Speed();
public void Start()
{
Console.WriteLine("Please Enter the Key To Start");
}


}
class Mustang : Vehical
{
public override void Speed()
{
Console.WriteLine("The Speed of the Mustang is 210 Km/Hr");
}
}
class Tesla : Vehical
{
public override void Speed()
{
Console.WriteLine("The speed of the Tesla is 320 Km/Hr");
}
}
class App {
public static void Main(string[] args)
{
Mustang m = new Mustang();
m.Start();
m.Speed();


Tesla tesla = new Tesla();
tesla.Start();
tesla.Speed();

}
}
```


Microsoft Visual Studio Debu

```
Please Enter the Key To Start
The Speed of the Mustang is 210 Km/Hr
Please Enter the Key To Start
The speed of the Tesla is 320 Km/Hr

D:\DotNetLab\AbstractClass\bin\Debug\net6.0\AbstractClass.exe (process 17572
) exited with code 0.
Press any key to close this window . . .
```

Write a program to demonstrate exception handline (try, catch, throw throws)

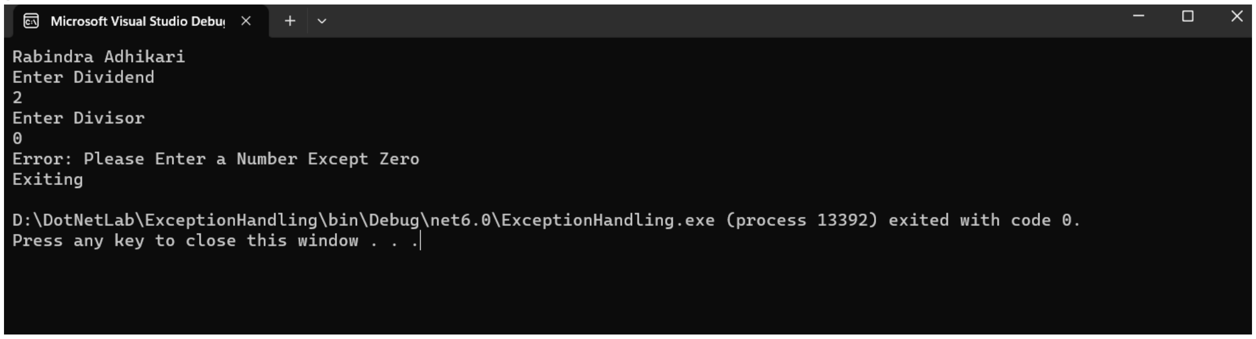
```
class program
{
public static void Main(string[] args)
{
Console.WriteLine("Rabindra Adhikari");
Console.WriteLine("Enter Divident");
int num = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter Divisior");
int num1=Convert.ToInt32(Console.ReadLine());


try
{
var result = num / num1;
Console.WriteLine("The Division is Successful{0}/{1}={2}",num,num1,result);
}
catch (Exception e) {
Console.WriteLine("The Division Can't Be Successful");
}

}
}
```



```
Rabindra Adhikari
Enter Divident
125
Enter Divisior
5
The Division is Successful125/5=25

D:\DotNetLab\ExceptionHandling\bin\Debug\net6.0\ExceptionHandling.exe (process 17524) exited with code 0.
Press any key to close this window . . .
```

Throw

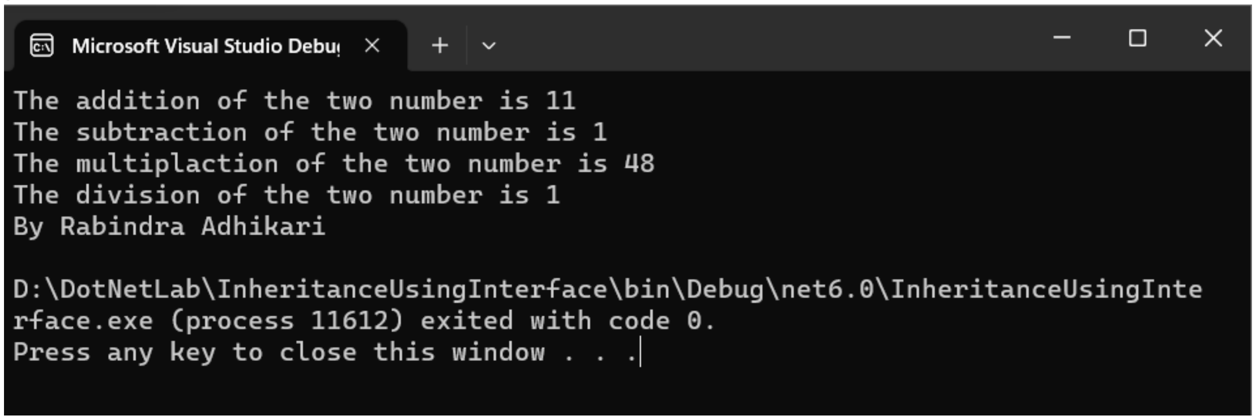
```
using System;

class Program
{
public static void Main(string[] args)
{
Console.WriteLine("Rabindra Adhikari");
Console.WriteLine("Enter Dividend");
int num = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter Divisor");
int num1 = Convert.ToInt32(Console.ReadLine());

try
{
if (num1 == 0)
{
throw new IOException("Please Enter a Number Except Zero");
}
else
{
var result = num / num1;
Console.WriteLine("The Division is Successful: {0}/{1} = {2}", num, num1, result);
}
}
catch (IOException ex)
{
Console.WriteLine("Error: " + ex.Message);
}
finally
{
Console.WriteLine("Exiting");
}
}
}
```



```
Rabindra Adhikari
Enter Dividend
2
Enter Divisor
0
Error: Please Enter a Number Except Zero
Exiting

D:\DotNetLab\ExceptionHandling\bin\Debug\net6.0\ExceptionHandling.exe (process 13392) exited with code 0.
Press any key to close this window . . .
```

Write a program to demonstrate interface

```
public interface IAdd
{
void Add(int a, int b);
}

public interface Isub
{
void Sub(int a, int b);
}

public interface Imul
{
void Mul(int a, int b);
}

public interface Idiv
{
void Div(int a, int b);
}
public class Calculator : IAdd, Isub, Imul, Idiv
{
public int resultAdd;
public int resultSub;
public int resultMul;
public int resultDiv;

public void Add(int a, int b)
{
resultAdd = a + b;
}

public void Sub(int a, int b)
{
resultSub = a - b;
}
public void Mul(int a, int b)
{
resultMul = a * b;
}

public void Div(int a, int b)
{
resultDiv = a / b;
}
public class MainClass
{
static void Main()
{
Calculator calculator = new Calculator();
calculator.Add(5, 6);
calculator.Sub(7, 6);
calculator.Mul(8, 6);
calculator.Div(9, 6);

Console.WriteLine("The addition of the two number is " + calculator.resultAdd);
Console.WriteLine("The subtraction of the two number is " + calculator.resultSub);
Console.WriteLine("The multiplaction of the two number is " + calculator.resultMul);
Console.WriteLine("The division of the two number is " + calculator.resultDiv);
Console.WriteLine("By Rabindra Adhikari");



}
}
}
```



```
The addition of the two number is 11
The subtraction of the two number is 1
The multiplaction of the two number is 48
The division of the two number is 1
By Rabindra Adhikari

D:\DotNetLab\InheritanceUsingInterface\bin\Debug\net6.0\InheritanceUsingInte
rface.exe (process 11612) exited with code 0.
Press any key to close this window . . .
```

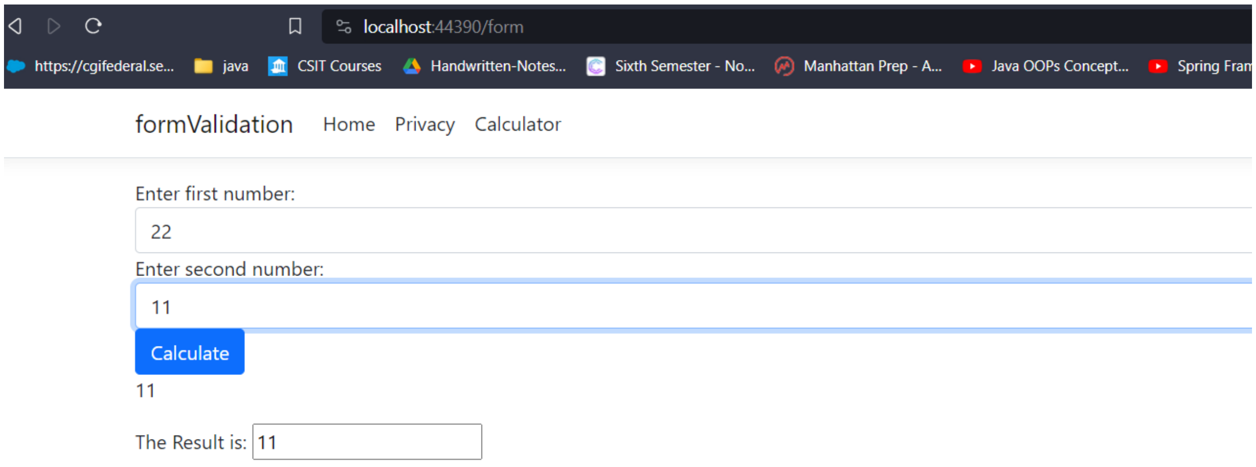
Create a web form to find subtraction of two integer numbers. Use two label that display Enter first number and enter second number, two text box for taking an input and third text box for output. When the submit button is clicked check for emptiness of form's fields, check whether the first input is greater than second or not. If all the condition are matched display the output in third text box.

Form.cshtml.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using System.ComponentModel.DataAnnotations;

namespace formValidation.Pages
{
public class formModel : PageModel
{

public int? firstNumber {  get; set; }

public int? secondNumber { get; set; }

public string? Result { get; set; }
public void OnGet()
{
}
public void OnPost()
{
if (int.TryParse(Request.Form["firstNumber"], out int first) &&
int.TryParse(Request.Form["secondNumber"], out int second))
{
if (first < second)
{
Result = "First Number must be greater than or equal to the second number.";
}
else
{
int result = first - second;
Result = result.ToString();
}
}
else
{
Result = "Please enter valid numbers.";
}
}
}
}
```

Form.cshtml

```
@page
@model formValidation.Pages.formModel

<!DOCTYPE html>
<html>
<head>
<title>Subtraction Form</title>
</head>
<body>
<div>
<form method="post">
<div class="form-group">
<label asp-for="firstNumber">Enter first number:</label>
<input asp-for="firstNumber" class="form-control" name="firstNumber" />
<span asp-validation-for="firstNumber" class="text-danger"></span>
</div>
<div class="form-group">
<label asp-for="secondNumber">Enter second number:</label>
<input asp-for="secondNumber" class="form-control" name="secondNumber" />
<span asp-validation-for="secondNumber" class="text-danger"></span>
</div>
<button type="submit" class="btn btn-primary">Calculate</button>
</form>


@if (!string.IsNullOrEmpty(Model.Result))
{
<p>@Model.Result</p>
<div>
<label>The Result is:</label>
<input type="text" id="txtResult" name="txtResult" value="@Model.Result" readonly />
</div>
}
</div>
</body>
</html>
```


localhost:44390/form

https://cgifederal.se... java CSIT Courses Handwritten-Notes... Sixth Semester - No... Manhattan Prep - A... Java OOPs Concept... Spring Fram

formValidation Home Privacy Calculator

Enter first number:

22

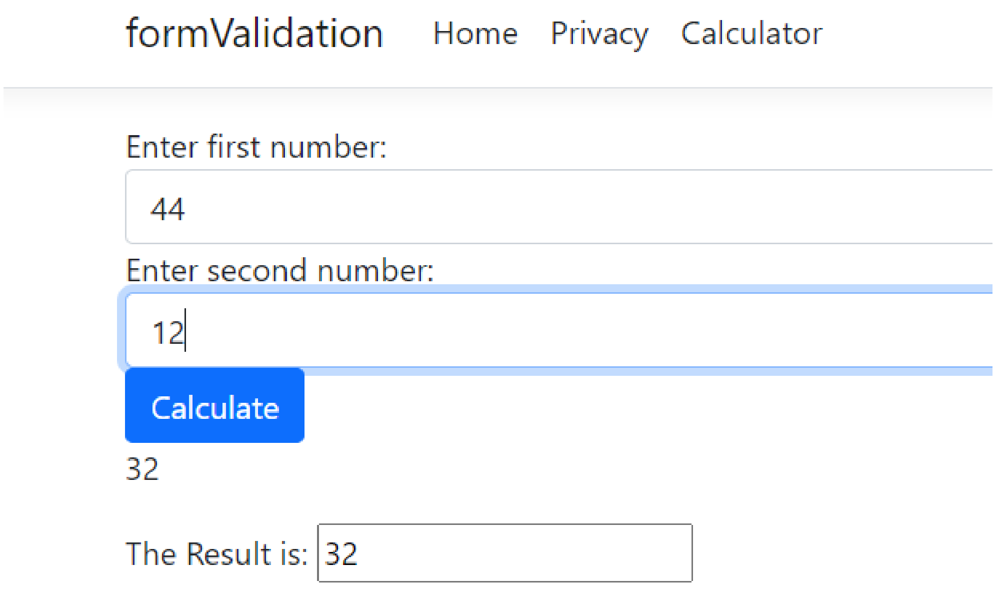
Enter second number:

11

Calculate

11

The Result is: 11


formValidation Home Privacy Calculator

Enter first number:

44

Enter second number:

12

Calculate

32

The Result is: 32

Create a web form for registration which should contains username, password, repassword, gender (radio button), course (checkbox) and country (dropdown) and submit button. After a submit, button is pressed display the entered data in table format.

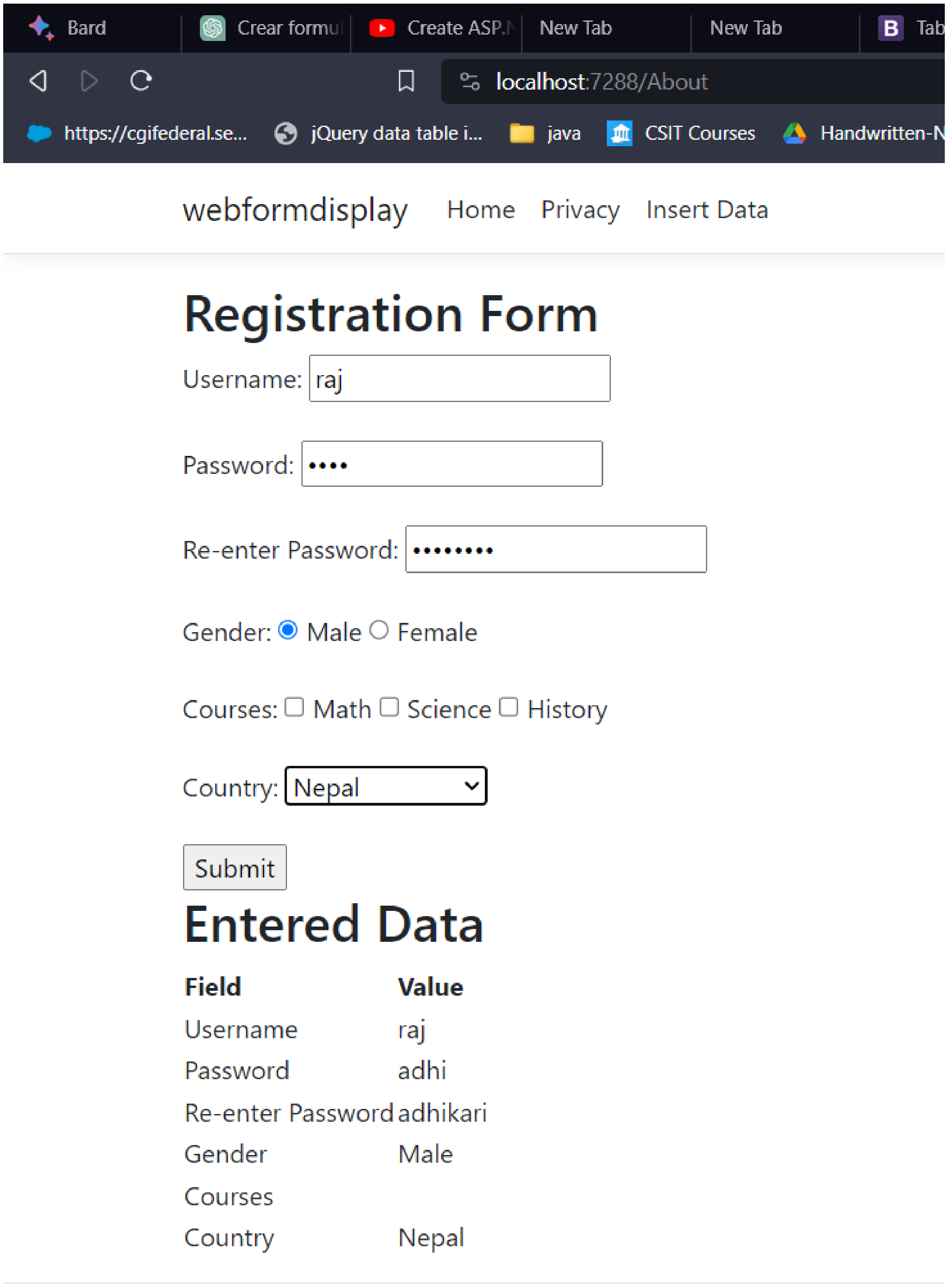
About.cshtml

```
@page
@model webformdisplay.Pages.AboutModel

<h2>Registration Form</h2>
<form method="post">
<label for="username">Username:</label>
<input type="text" id="username" name="username"><br><br>

<label for="password">Password:</label>
<input type="password" id="password" name="password"><br><br>

<label for="repassword">Re-enter Password:</label>
<input type="password" id="repassword" name="repassword"><br><br>

<label>Gender:</label>
<input type="radio" id="male" name="gender" value="Male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="Female">
<label for="female">Female</label><br><br>

<label>Courses:</label>
<input type="checkbox" id="math" name="course" value="Math">
<label for="math">Math</label>
<input type="checkbox" id="science" name="course" value="Science">
<label for="science">Science</label>
<input type="checkbox" id="history" name="course" value="History">
<label for="history">History</label><br><br>

<label for="country">Country:</label>
<select id="country" name="country">
<option value="" disabled selected>Select Country</option>
<option value="Nepal">Nepal</option>
<option value="India">India</option>
<!-- Add more countries as needed -->
</select><br><br>

<input type="submit" value="Submit">
</form>
```

```
@if (Model.hasData)
{
<h2>Entered Data</h2>
<table>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
<tr>
<td>Username</td>
<td>@Model.username</td>
</tr>
<tr>
<td>Password</td>
<td>@Model.password</td>
</tr>
<tr>
<td>Re-enter Password</td>
<td>@Model.repassword</td>
</tr>
<tr>
<td>Gender</td>
<td>@Model.gender</td>
</tr>
<tr>
<td>Courses</td>
<td>@string.Join(", ", Model.course)</td>
</tr>
<tr>
<td>Country</td>
<td>@Model.country</td>
</tr>
</table>
}
```

About.cshtml.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace webformdisplay.Pages
{
public class AboutModel : PageModel
{
[BindProperty]
public bool hasData { get; set; }
[BindProperty]
public string username { get; set; }
[BindProperty]
public string password { get; set; }
[BindProperty]
public string repassword { get; set; }
[BindProperty]
public string gender { get; set; }
[BindProperty]
public string[] course { get; set; }
[BindProperty]
public string country { get; set; }

public void OnGet()
{
}

public void OnPost()
{
hasData = true;
username = Request.Form["username"];
password = Request.Form["password"];
repassword = Request.Form["repassword"];
gender = Request.Form["gender"];
course = Request.Form["course[]"];
country = Request.Form["country"];
}
}
}
```



Registration Form

Username:	raj
Password:	••••
Re-enter Password:	••••••••
Gender:	[✓ Male] Female
Courses:	☐ Math ☐ Science ☐ History
Country:	Nepal

Entered Data

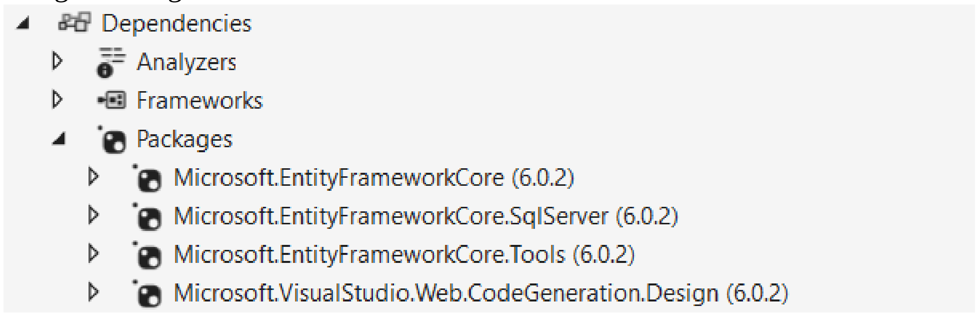
Field	Value
Username	raj
Password	adhi
Re-enter Password	adhikari
Gender	Male
Courses	
Country	Nepal

Using Entity framework create a table tbl_officer having field (id, name, gender, phone, department and position) after this perform complete CRUDE operation (insert, update, display and delete)

Officer.CS

```
namespace OfficersApp.Models
{
public class Officer
{

[Key]
public int Id { get; set; }

public string Name { get; set; }
[Required]
public string Gender { get; set; }
[Required]
public string Phone { get; set; }
[Required]
public string Department { get; set; }
[Required]
public string Position { get; set; }


}
}
```

Nudget Packages



```
Dependencies
  Analyzers
  Frameworks
  Packages
    Microsoft.EntityFrameworkCore (6.0.2)
    Microsoft.EntityFrameworkCore.SqlServer (6.0.2)
    Microsoft.EntityFrameworkCore.Tools (6.0.2)
    Microsoft.VisualStudio.Web.CodeGeneration.Design (6.0.2)
```

AppSetting.Json

```
"ConnectionStrings": {
"DevConnection":
"server=THOMASRAJ\\SQLEXPRESS;database=tbl_officer;Trusted_Connection=True;MultipleActiveResultSets=True;"
}
```

Program.cs

```
builder.Services.AddDbContext<OfficerDbContext>(options=>options.UseSqlServer(builder.Co
nfiguration.GetConnectionString("DevConnection")));
```

OfficeController

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using OfficersApp.Models;

namespace OfficersApp.Controllers
{
public class OfficerController : Controller
{
private readonly OfficerDbContext context;

public OfficerController(OfficerDbContext officedbcontext)
{
context = officedbcontext;

}

[HttpGet]
public IActionResult Index()
{
var officer = context.Officers.ToList();
return View(officer);
}
[HttpGet]
public IActionResult Add()
{
return View();
}

[HttpPost]
public IActionResult Add(AddOfficerViewModel addOfficerRequest) {

var officer = new Officer()
{
Id = addOfficerRequest.Id,
Name = addOfficerRequest.Name,
Gender = addOfficerRequest.Gender,
Phone = addOfficerRequest.Phone,
Department = addOfficerRequest.Department,
Position = addOfficerRequest.Position


};
```

```
context.Add(officer);
context.SaveChanges();
return RedirectToAction("Index");



}
[HttpGet]
public async Task<IActionResult> View(int id) {
var officer = await context.Officers.FirstOrDefaultAsync(x => x.Id == id);

if (officer != null)
{
var viewModel = new UpdateOfficerViewModel()
{
Id = officer.Id,
Name = officer.Name,
Gender = officer.Gender,
Phone = officer.Phone,
Department = officer.Department,
Position = officer.Position


};
return await Task.Run(()=>View("View",viewModel));


}

return RedirectToAction("Index");

}

[HttpPost]
public async Task<IActionResult> View(UpdateOfficerViewModel model)
{
var officer = await context.Officers.FindAsync(model.Id);
if (officer != null)
{
officer.Name = model.Name;
officer.Gender = model.Gender;
officer.Phone = model.Phone;
officer.Department = model.Department;
officer.Position = model.Position;

await context.SaveChangesAsync();
return RedirectToAction("Index");
}

return RedirectToAction("Index");


}

[HttpPost]
public async Task<IActionResult> Delete(UpdateOfficerViewModel model)
{
var officer=await context.Officers.FindAsync(model.Id);
if (officer != null)
{
context.Officers.Remove(officer);
await context.SaveChangesAsync();

return RedirectToAction("Index");
}

return RedirectToAction("Index");

}
}
}
```

Model
AddOfficerViewModel

```
using System.ComponentModel.DataAnnotations;

namespace OfficersApp.Models
{
public class AddOfficerViewModel
{
[Key]
public int Id { get; set; }
public string Name { get; set; }

public string Gender { get; set; }

public string Phone { get; set; }

public string Department { get; set; }

public string Position { get; set; }
}
}
```

UpdateEmployeeView Model

```
namespace OfficersApp.Models
{
public class UpdateOfficerViewModel
{

public int Id { get; set; }
public string Name { get; set; }

public string Gender { get; set; }

public string Phone { get; set; }

public string Department { get; set; }

public string Position { get; set; }
}
}
```

View

Add.cshtml

```
@model OfficersApp.Models.AddOfficerViewModel
@{
}
<h1>Add Employee</h1>
<form method="post" action="Add" class="m-5">
<div class="md-3">
<label for="">Id</label>
<input type="hidden" class="form-control" asp-for="Id">
</div>

<div class="md-3">
<label for="">Name</label>
<input type="text" class="form-control" asp-for="Name">
</div>
<div class="md-3">
<label for="">Gender</label>
<input type="text" class="form-control" asp-for="Gender">
</div>
<div class="md-3">
<label for="">Phone</label>
<input type="number" class="form-control" asp-for="Phone">
</div>
<div class="md-3">
<label for="">Department</label>
<input type="text" class="form-control" asp-for="Department">
</div>
```

```
<div class="md-3">
<label for="">Position</label>
<input type="text" class="form-control" asp-for="Position">
</div>

<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Index.cshtml

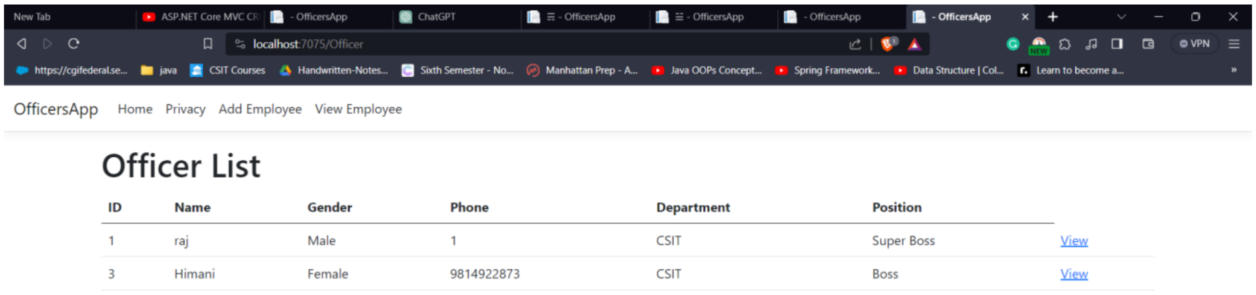
```
@model List<OfficersApp.Models.Officer>
@{
}

<h1>Officer List</h1>

<table class="table">

<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Gender</th>
<th>Phone</th>
<th>Department</th>
<th>Position</th>
</tr>
</thead>
<tbody>
@foreach(var officer in Model)
{
<tr>
<td>@officer.Id</td>
<td>@officer.Name</td>
<td>@officer.Gender</td>
<td>@officer.Phone</td>
<td>@officer.Department</td>
<td>@officer.Position</td>
<td><a href="Officer/View/@officer.Id">View</a></td>

</tr>
}
</tbody>
</table>
```

View.cshtml

```
@model OfficersApp.Models.UpdateOfficerViewModel
```

```
@{
}



<h1>Edit Employee</h1>
<form method="post" action="View" class="m-5">
<div class="md-3">
<label for="">Id</label>
<input type="text" class="form-control" asp-for="Id" readonly>
</div>

<div class="md-3">
<label for="">Name</label>
<input type="text" class="form-control" asp-for="Name">
</div>
<div class="md-3">
<label for="">Gender</label>
<input type="text" class="form-control" asp-for="Gender">
</div>
<div class="md-3">
<label for="">Phone</label>
<input type="number" class="form-control" asp-for="Phone">
</div>
<div class="md-3">
<label for="">Department</label>
<input type="text" class="form-control" asp-for="Department">
</div>
<div class="md-3">
<label for="">Position</label>
<input type="text" class="form-control" asp-for="Position">
</div>

<button type="submit" class="btn btn-primary">Submit</button>
<button type="submit" class="btn btn-denger"
asp-action="Delete"
asp-controller="Officer">Delete</button>


</form>
```

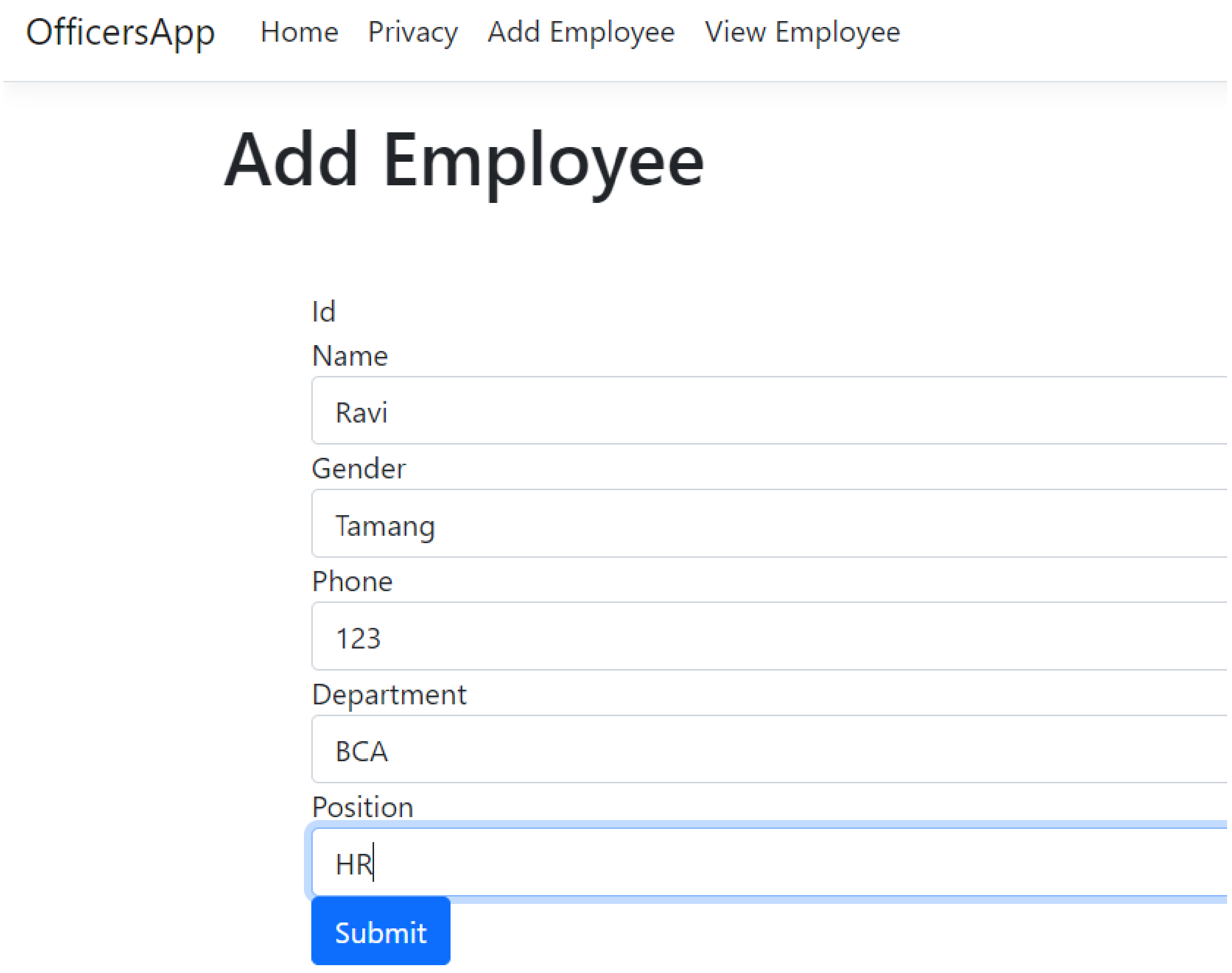
Index.cshtml



Officer List

ID	Name	Gender	Phone	Department	Position	
1	raj	Male	1	CSIT	Super Boss	View
3	Himani	Female	9814922873	CSIT	Boss	View

Add Emp



Add Employee

Id

Name

Ravi

Gender

Tamang

Phone

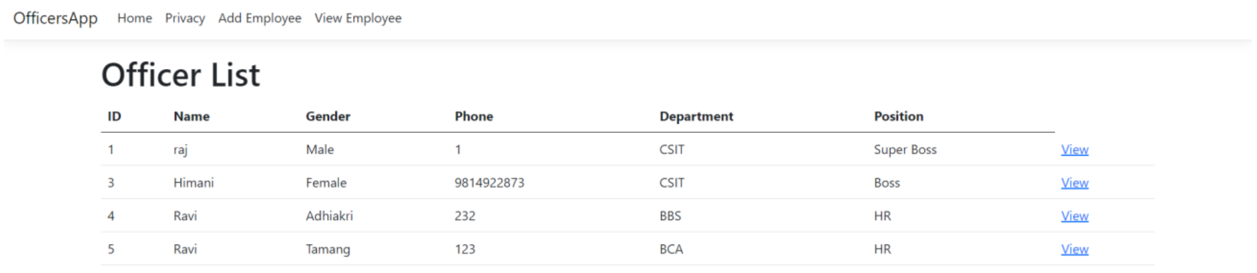
123

Department

BCA

Position

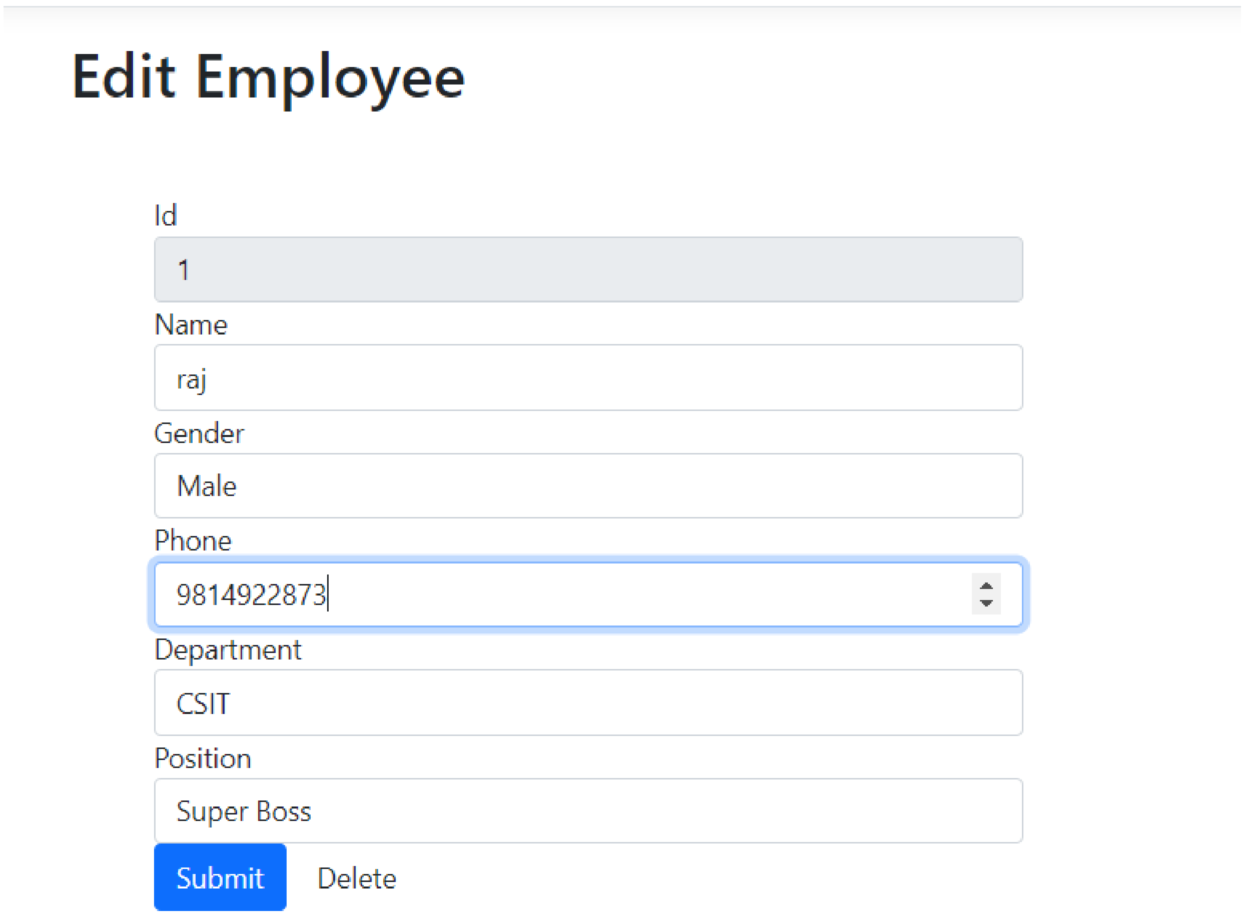
HR



Officer List

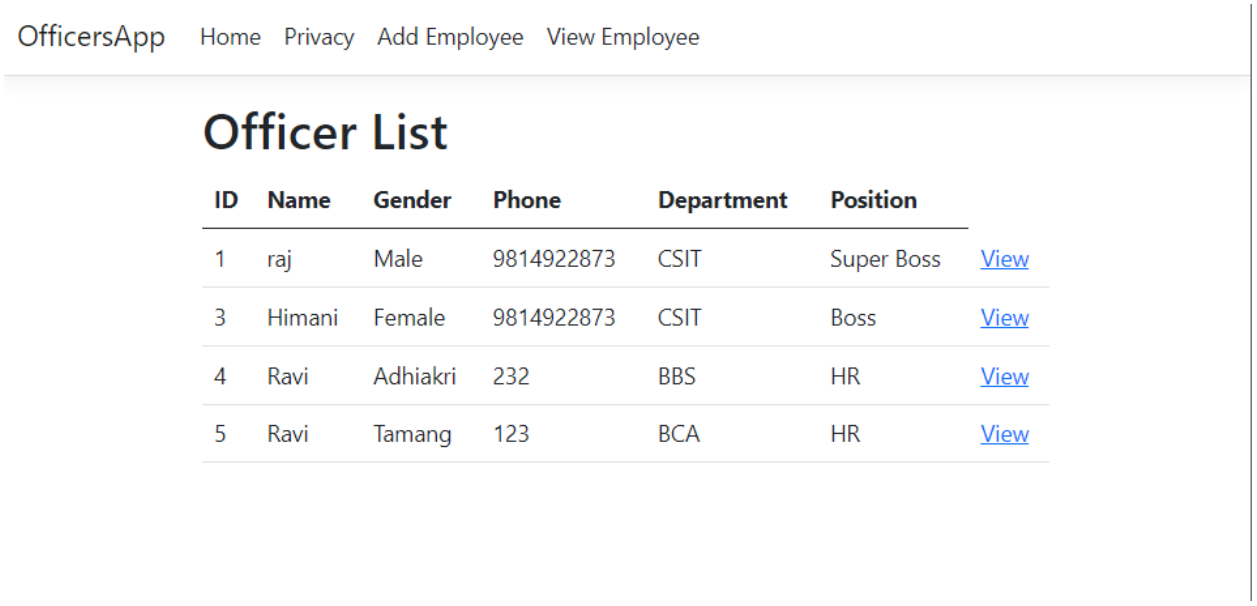
ID	Name	Gender	Phone	Department	Position	
1	raj	Male	1	CSIT	Super Boss	View
3	Himani	Female	9814922873	CSIT	Boss	View
4	Ravi	Adhiakri	232	BBS	HR	View
5	Ravi	Tamang	123	BCA	HR	View

Edit



Edit Employee

Id	1
Name	raj
Gender	Male
Phone	9814922873
Department	CSIT
Position	Super Boss



Officer List

ID	Name	Gender	Phone	Department	Position	
1	raj	Male	9814922873	CSIT	Super Boss	View
3	Himani	Female	9814922873	CSIT	Boss	View
4	Ravi	Adhiakri	232	BBS	HR	View
5	Ravi	Tamang	123	BCA	HR	View

Using ADO.net perform crude operation. Assume you have table tbl_student (id, name, gender, faculty and grade) and database db_prime. All the input should be taken from user for data insertion.

StudentController.cs

```
using Microsoft.AspNetCore.Mvc;

using SecondCrudOperation.Models;

using System.Data;

using System.Data.SqlClient;


namespace SecondCrudOperation.Controllers

{

public class StudentController : Controller

{

private readonly string connectionString =
"Server=THOMASRAJ\\SQLEXPRESS;Database=tbl_student;Integrated
Security=True;TrustServerCertificate=True";

public IActionResult DisplayAll()

{

SqlConnection conn = new SqlConnection(connectionString);


try

{

// Opening the connection

conn.Open();


// Creating a list to store all the retrieved students

List<Student> students = new List<Student>();


string readCmd = "SELECT * FROM [tbl_student];";
```

```
using (SqlCommand cmd = new SqlCommand(readCmd, conn))
{
// Retrieving the data
using (SqlDataReader reader = cmd.ExecuteReader())
{
// Reading all the data
while (reader.Read())
{
// Adding the data read to the student object
Student std = new Student();
std.Id = (int)reader["Id"];
std.Name = (string)reader["Name"];
std.Gender = (string)reader["Gender"];
std.Faculty = (string)reader["Faculty"];
std.Grade = (String)reader["Grade"];
// Adding the student to the list of students.
students.Add(std);
}
}
}

// Closing the connection
conn.Close();

return View(students);
}

finally
{
// Ensure that the connection is closed in case of exceptions
```

```
if (conn.State == ConnectionState.Open)
{
conn.Close();
}
}
}
}
}
```

DisplayAll.cshtml

```
@model List<SecondCrudOperation.Models.Student>

@{
}

<style>
body {
font-family: Arial, sans-serif;
}

h1 {
text-align: center;
color: #333;
}

table {
width: 100%;
border-collapse: collapse;
margin: 20px 0;
}
```

```
th, td {
padding: 12px;
text-align: left;
border-bottom: 1px solid #ddd;
}

th {
background-color: #f2f2f2;
color: #333;
}

tr:nth-child(even) {
background-color: #f2f2f2;
}

tr:hover {
background-color: #ddd;
}

.action-buttons {
display: flex;
justify-content: center;
}

.action-buttons a {
margin: 5px;
text-decoration: none;
padding: 8px 15px;
background-color: #007bff;
color: #fff;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s;
}

.action-buttons a:hover {
background-color: #0056b3;
}

.error-message {
color: red;
text-align: center;
font-weight: bold;
margin-top: 10px;
}
</style>

<h1>Students Data</h1>
<div class="action-buttons">
<a asp-controller="Add" asp-action="AddStudents"><button>Add Students</button></a>
</div>

<p class="error-message">@ViewBag.ErrorMsg</p>

<table>
<thead>
<tr>
<th>Id</th>
```

```
<th>Name</th>
<th>Gender</th>
<th>Faculty</th>
<th>Grade</th>
<th>Action</th>
</tr>
</thead>
<tbody>
@foreach (var std in Model)
{
<tr>
<td>@std.Id</td>
<td>@std.Name</td>
<td>@std.Gender</td>
<td>@std.Faculty</td>
<td>@std.Grade</td>
<td class="action-buttons">
<a asp-controller="Update" asp-action="Edit" asp-route-id="@std.Id"><button>Update</button></a>
<a asp-controller="Delete" asp-action="Delete" asp-route-id="@std.Id"><button>Delete</button></a>
</td>
</tr>
}
</tbody>
</table>
```

AddController.cs

```
using Microsoft.AspNetCore.Mvc;
using SecondCrudOperation.Models;
using System.Data.SqlClient;

namespace SecondCrudOperation.Controllers
```

```
{

public class AddController : Controller

{

private readonly string connectionString =
"Server=THOMASRAJ\\SQLEXPRESS;Database=tbl_student;Integrated
Security=True;TrustServerCertificate=True";

[HttpGet]

public IActionResult AddStudents()

{

return View();

}


[HttpPost]

public IActionResult AddStudents(Student student)


{


if (ModelState.IsValid)

{

try

{

string connectionString = "Server=THOMASRAJ\\SQLEXPRESS;Database=tbl_student;Integrated
Security=True;TrustServerCertificate=True"; // Replace with your actual connection string


using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();


string addSql = "INSERT INTO tbl_student ( Name, Gender, Faculty, Grade) VALUES ( @name,
@gender, @faculty, @grade)";
using (SqlCommand cmd = new SqlCommand(addSql, connection))
{

cmd.Parameters.AddWithValue("@name", student.Name);
cmd.Parameters.AddWithValue("@gender", student.Gender);
cmd.Parameters.AddWithValue("@faculty", student.Faculty);
cmd.Parameters.AddWithValue("@grade", student.Grade);

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)
{
// Data inserted successfully
// You can handle success or redirect to another page here
}
else
{
ViewBag.ErrorMsg = "No rows were inserted. Please check your data and SQL statement.";
// You may want to handle the error or redirect to another page
}
}
}
}
catch (SqlException ex)
{
ViewBag.ErrorMsg = "Connection Failed: " + ex.Message;
// Handle the error, show a message, or redirect to an error page
}

}
```

```
return RedirectToAction("DisplayAll", "Student");
}

}
}
```

AddStudent.cshtml

```
@model SecondCrudOperation.Models.Student

@{
ViewData["Title"] = "Add Student";
}

<h1>Add Student</h1>

<form asp-controller="Add" asp-action="AddStudents" method="post">
<div class="form-group">
<label asp-for="Name">Name:</label>
<input asp-for="Name" class="form-control" />
</div>

<div class="form-group">
<label asp-for="Gender">Gender:</label>
<input asp-for="Gender" class="form-control" />
</div>

<div class="form-group">
<label asp-for="Faculty">Faculty:</label>
<input asp-for="Faculty" class="form-control" />
</div>
```

```
<div class="form-group">
<label asp-for="Grade">Grade:</label>
<input asp-for="Grade" class="form-control" />
</div>

<button type="submit" class="btn btn-primary">Add Student</button>
</form>
```

UpdateController.cs

```
using Microsoft.AspNetCore.Mvc;
using SecondCrudOperation.Models;
using System.Data.SqlClient;
using System.Data.SqlTypes;

namespace SecondCrudOperation.Controllers
{

public class UpdateController : Controller
{
private string connectionString =
"Server=THOMASRAJ\\SQLEXPRESS;Database=tbl_student;Integrated
Security=True;TrustServerCertificate=True";
[HttpGet]
public IActionResult Edit(int id)
{

SqlConnection connection = new SqlConnection(connectionString);
connection.Open();

string sqlcmd = "SELECT * FROM tbl_student WHERE id=@id";
SqlCommand command = new SqlCommand(sqlcmd, connection);
// Add the student ID to the command parameters.
command.Parameters.AddWithValue("@id", id);

// Execute the command and get the results.
SqlDataReader reader = command.ExecuteReader();

// Create a new student object to store the results.
Student student = new Student();

// Read the results and populate the student object.
if (reader.Read())
{
student.Id = reader.GetInt32(0);
student.Name = reader.GetString(1);
student.Gender = reader.GetString(2);
student.Faculty = reader.GetString(3);
student.Grade = reader.GetString(4);
}

// Close the reader and the connection.
reader.Close();
connection.Close();

// Return the student object to the view.
return View(student);
}

public IActionResult Edit(Student student)
```

```
{
//Creating the connection
SqlConnection sqlConnection = new SqlConnection(connectionString);
//Opening the connection
sqlConnection.Open();

//Creating the update query
string updateQuery = @"Update tbl_student SET
name = @name,
faculty = @faculty,
gender = @gender,
grade = @grade
WHERE id = @id";

using (SqlCommand cmd = new SqlCommand(@updateQuery, sqlConnection))
{
cmd.Parameters.AddWithValue("name", student.Name);
cmd.Parameters.AddWithValue("faculty", student.Faculty);
cmd.Parameters.AddWithValue("gender", student.Gender);
cmd.Parameters.AddWithValue("grade", student.Grade);
cmd.Parameters.AddWithValue("id", student.Id);

cmd.ExecuteNonQuery();
}

//Closing the connection
sqlConnection.Close();
return RedirectToAction("DisplayAll","Student");
}
}
```

```
}
```

Edit.cshtml

```
<style>
label, input {
display: block;
}
</style>
@model Student

<h1>Update Student Form</h1>

<form asp-controller="Update" asp-action="Edit" method="post">

<div>
<label asp-for="Id">Id</label>
<input type="text" asp-for="Id" />
<span asp-validation-for="Id"></span>
</div>

<div>
<label asp-for="Name">Name</label>
<input type="text" asp-for="Name" />
<span asp-validation-for="Name"></span>
</div>

<div>
<label asp-for="Gender">Gender</label>
<input type="text" asp-for="Gender" />
<span asp-validation-for="Gender"></span>
```

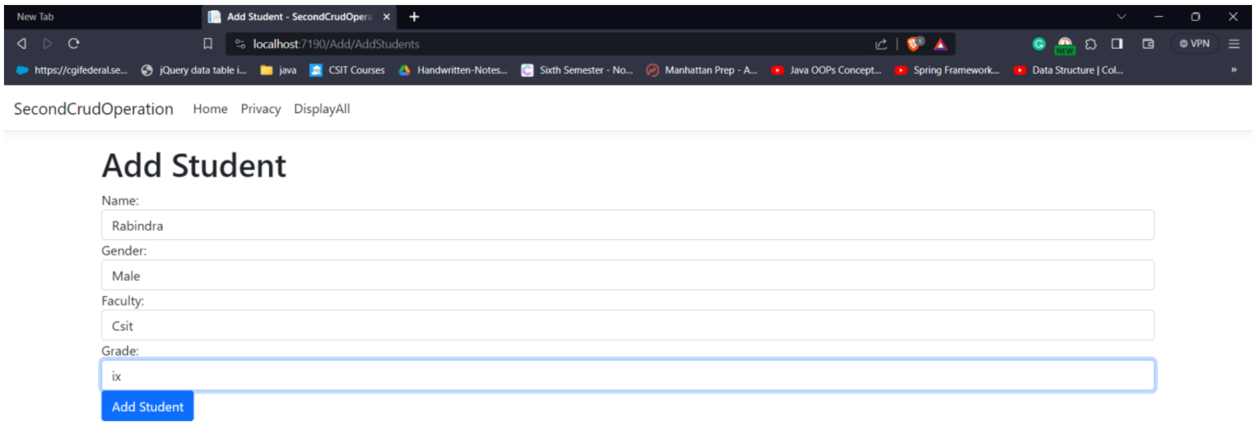
```
</div>


<div>
<label asp-for="Faculty">Faculty</label>
<input type="text" asp-for="Faculty" />
<span asp-validation-for="Faculty"></span>
</div>

<div>
<label asp-for="Grade">Grade</label>
<input type="text" asp-for="Grade" />
<span asp-validation-for="Grade"></span>
</div>

<input type="submit" />
</form>
```

Delete.Controller

```
using Microsoft.AspNetCore.Mvc;
using System.Data.SqlClient;

namespace SecondCrudOperation.Controllers
{
public class DeleteController : Controller
{
private string connectionString =
"Server=THOMASRAJ\\SQLEXPRESS;Database=tbl_student;Integrated
Security=True;TrustServerCertificate=True";
public IActionResult Delete(int id)
{
```

```
try
{
//Creating the connection
SqlConnection conn = new SqlConnection(connectionString);
//Opening the connection
conn.Open();

//Creating the query
string deleteQuery = @"DELETE FROM tbl_student WHERE id=@id";

//Executing the query
SqlCommand cmd = new SqlCommand(@deleteQuery, conn);
cmd.Parameters.AddWithValue("id", id);
cmd.ExecuteNonQuery();



//Closing the connection
conn.Close();
}
catch (Exception ex)
{
Console.WriteLine(ex.Message);
}

return RedirectToAction("DisplayAll","Student");
}


}
```

```
}
```



New Tab | Add Student - SecondCrudOpera

localhost:7190/Add/AddStudents

SecondCrudOperation Home Privacy DisplayAll

Add Student

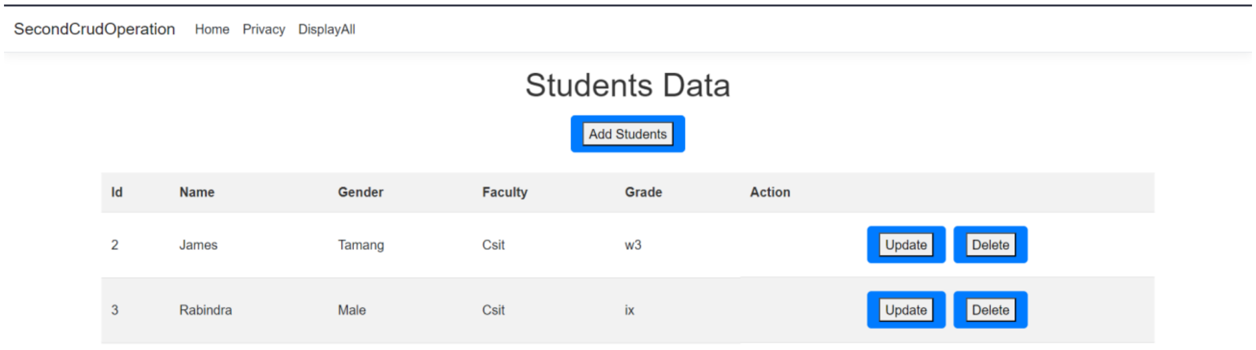
Name: Rabindra

Gender: Male

Faculty: Csit

Grade: ix

Add Student

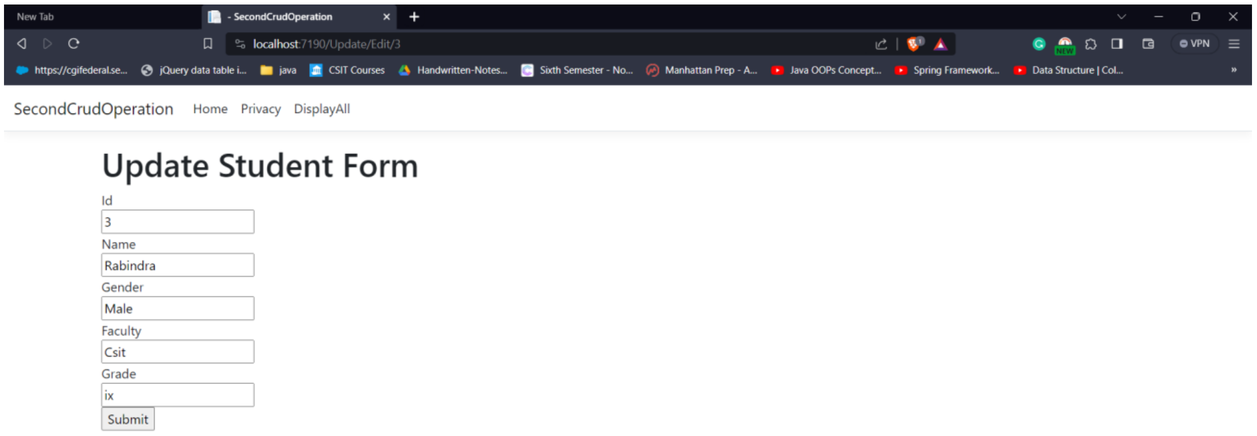


SecondCrudOperation Home Privacy DisplayAll

Students Data

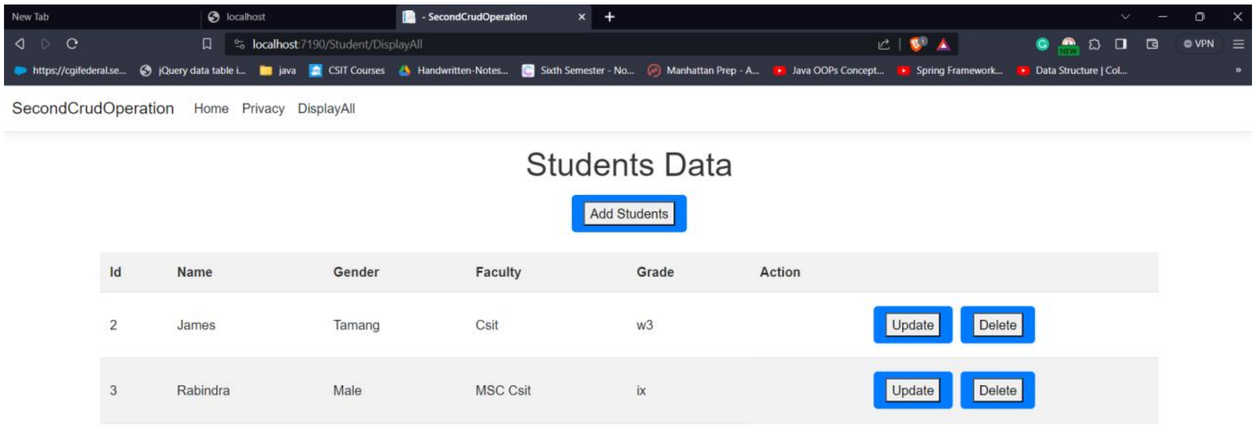
Add Students

Id	Name	Gender	Faculty	Grade	Action
2	James	Tamang	Csit	w3	Update Delete
3	Rabindra	Male	Csit	ix	Update Delete


localhost:7190/Update/Edit/3
SecondCrudOperation Home Privacy DisplayAll

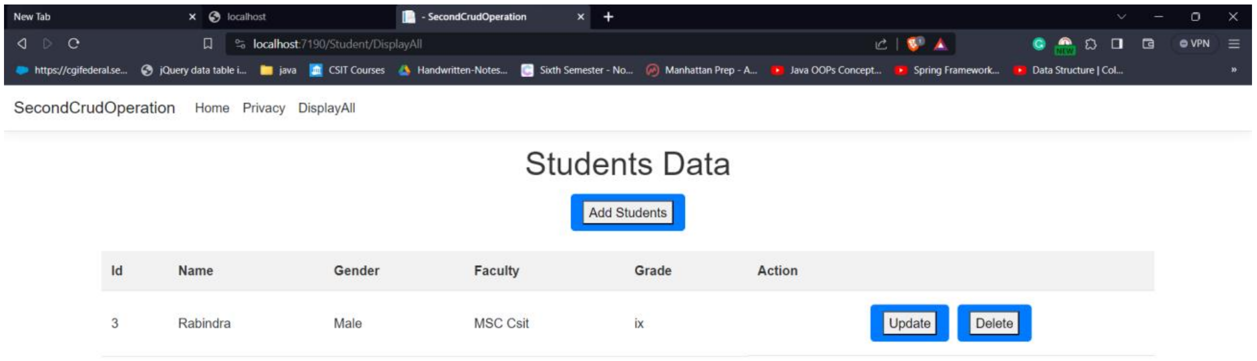
Update Student Form

Id	3
Name	Rabindra
Gender	Male
Faculty	Csit
Grade	ix


localhost:7190/Student/DisplayAll
SecondCrudOperation Home Privacy DisplayAll
Add Students

Students Data

Id	Name	Gender	Faculty	Grade	Action
2	James	Tamang	Csit	w3	Update Delete
3	Rabindra	Male	MSC Csit	ix	Update Delete


localhost:7190/Student/DisplayAll
SecondCrudOperation Home Privacy DisplayAll
Add Students

Students Data

Id	Name	Gender	Faculty	Grade	Action
3	Rabindra	Male	MSC Csit	ix	Update Delete