

ASSIGNMENT 02

CLOUD COMPUTING



**AIMAN ZIA SATTI**

**02-131212-028**

**BSE-6B**

**BAHRIA UNIVERSITY (KARACHI CAMPUS**)

Assignment 2

Cloud Computing

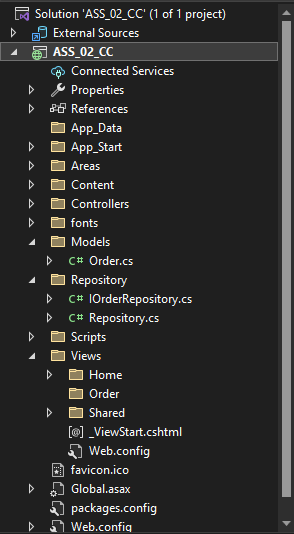
Class: **(BSE 6B)**

Course Instructor: **(Engr. Muhammad Faisal)**

Due date: **13/5/2024**

Student’s Name: **AIMAN ZIA SATTI**  Reg. No: **79301**

**Question:** In this assignment you are supposed to develop a web API that will contain the following.



**Order.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace ASS\_02\_CC.Models

{

public class Order

{

public int OrdId { get; set; }

public string CusName { get; set; }

public int Laptop { get; set; }

public int LCD { get; set; }

public int Phone { get; set; }

}

}

**Repository.cs:**using ASS\_02\_CC.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using static ASS\_02\_CC.Repository.Repository;

namespace ASS\_02\_CC.Repository

{

public class Repository

{

public class OrderRepository : IOrderRepository

{

private List<Order> \_orders = new List<Order>();

private int \_nextId = 1;

public OrderRepository()

{

Add(new Order { CusName = "Aiman Zia Satti", Laptop = 90, LCD = 2, Phone = 7 });

Add(new Order { CusName = "Sana", Laptop = 24, LCD = 87, Phone = 20 });

Add(new Order { CusName = "Hassan", Laptop = 0, LCD = 26, Phone = 78 });

}

public IEnumerable<Order> GetAll()

{

return \_orders;

}

public Order Get(int id)

{

return \_orders.Find(o => o.OrdId == id);

}

public Order Add(Order item)

{

if (item == null)

{

throw new ArgumentNullException("item");

}

item.OrdId = \_nextId++;

\_orders.Add(item);

return item;

}

public void Remove(int id)

{

\_orders.RemoveAll(o => o.OrdId == id);

}

public bool Update(Order item)

{

if (item == null)

{

throw new ArgumentNullException("item");

}

int index = \_orders.FindIndex(o => o.OrdId == item.OrdId);

if (index == -1)

{

return false;

}

\_orders.RemoveAt(index);

\_orders.Add(item);

return true;

}

}

}

}

**IOrderRepository:**

using ASS\_02\_CC.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace ASS\_02\_CC.Repository

{

public interface IOrderRepository

{

IEnumerable<Order> GetAll();

Order Get(int id);

Order Add(Order item);

void Remove(int id);

bool Update(Order item);

}

}

**ORDER CONTROLLER:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using ASS\_02\_CC.Models;

using ASS\_02\_CC.Repository;

using static ASS\_02\_CC.Repository.Repository;

namespace ASS\_02\_CC.Controllers

{

public class OrderController : ApiController

{

static readonly IOrderRepository repository = new OrderRepository();

public IEnumerable<Order> Get()

{

return repository.GetAll();

}

public Order Get(int id)

{

Order item = repository.Get(id);

if (item == null)

{

throw new HttpResponseException(HttpStatusCode.NotFound);

}

return item;

}

public IEnumerable<Order> GetOrderByCusName(string CName)

{

return repository.GetAll().Where(

p => string.Equals(p.CusName, CName, StringComparison.OrdinalIgnoreCase));

}

public HttpResponseMessage Post(Order item)

{

item = repository.Add(item);

var response = Request.CreateResponse<Order>(HttpStatusCode.Created, item);

string uri = Url.Link("DefaultApi", new { id = item.OrdId });

response.Headers.Location = new Uri(uri);

return response;

}

public void Put(int id, Order order)

{

order.OrdId = id;

if (!repository.Update(order))

{

throw new HttpResponseException(HttpStatusCode.NotFound);

}

}

public void Delete(int id)

{

Order item = repository.Get(id);

if (item == null)

{

throw new HttpResponseException(HttpStatusCode.NotFound);

}

repository.Remove(id);

}

}

}

**WEBAPI CONFIG:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace ASS\_02\_CC

{

public static class WebApiConfig

{

public static void Register(HttpConfiguration config)

{

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{id}",

defaults: new { id = RouteParameter.Optional }

);

config.Routes.MapHttpRoute(

name: "OrderByName",

routeTemplate: "api/{controller}/{action}/{CName}",

defaults: new { action = "GetOrderByCusName", CName = RouteParameter.Optional }

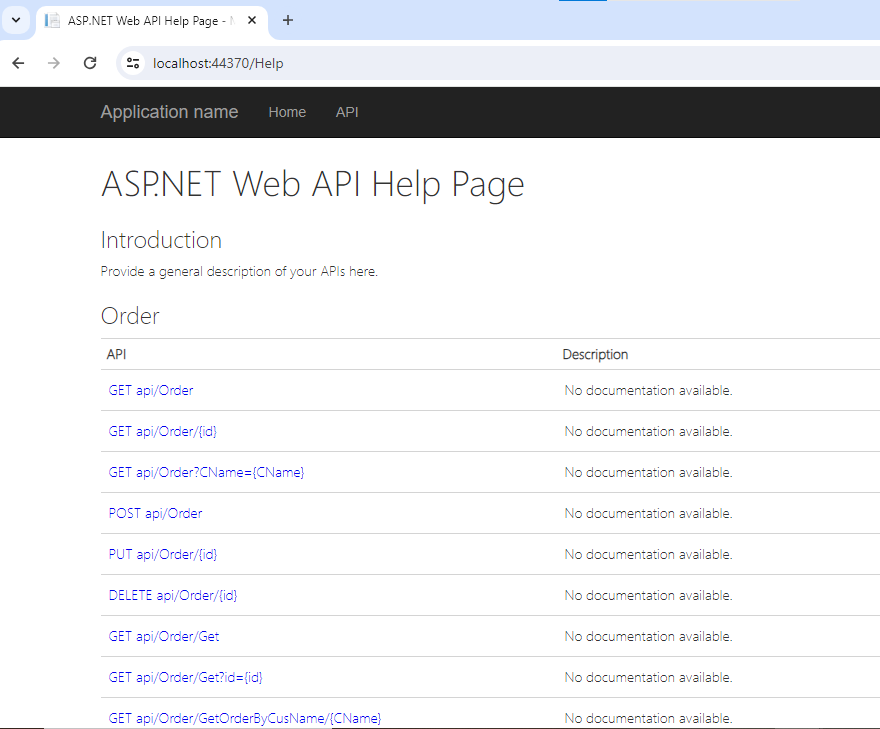
);

}

}

}

**OUTPUTS:**



A screenshot of a computer

Description automatically generated

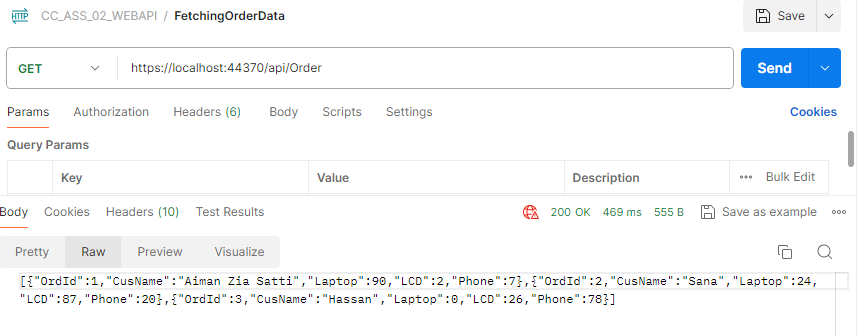
A screenshot of a computer

Description automatically generated

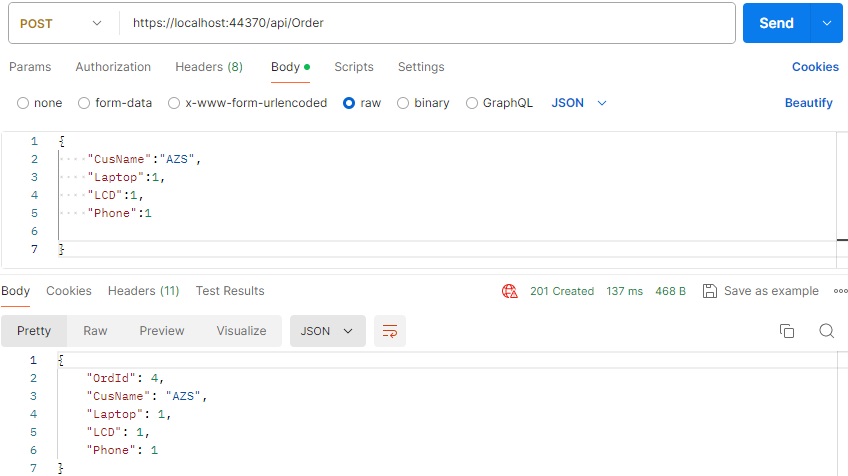
**GET ORDER:**

A screenshot of a computer code

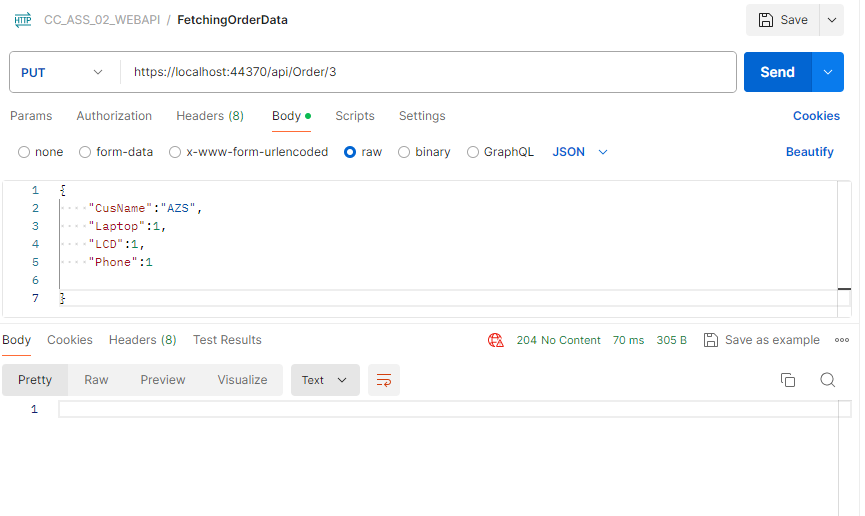
Description automatically generated

**GET ALL:**

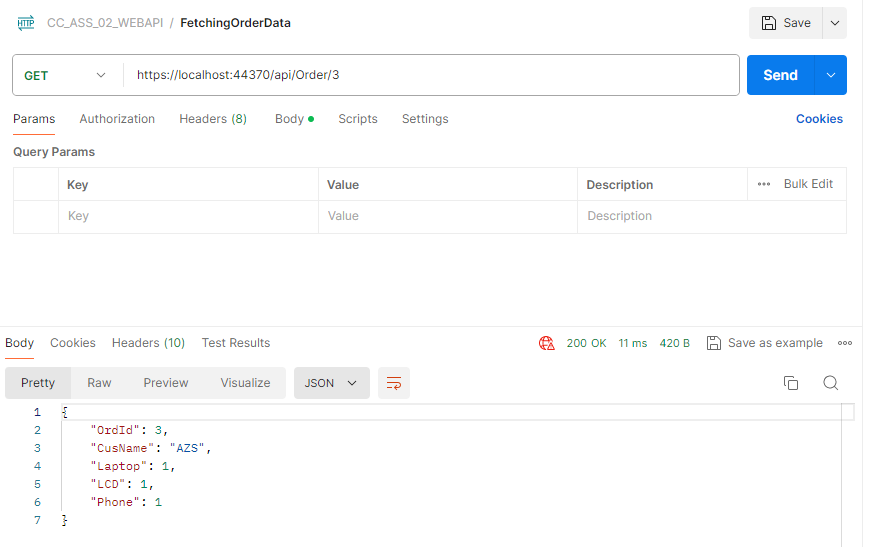
**POST:**



**PUT:**



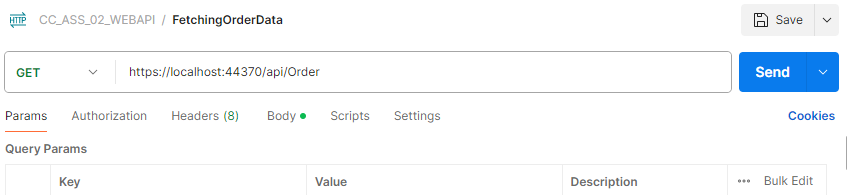
**GET ORDER BY ID:**

****

**DELETE:  
A screenshot of a computer

Description automatically generated**

**AFTER DELETION AND UPDATION:**

****

**OrdId:3 HAS BEEN DELETED AND REMAINING RECORD HAS BEEN DISPLAYED:**

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**