namespace customerService.Models

{

public class Product

{

[Key]

public int Product\_Id { get; set; }

public string Product\_Name { get; set; }

public string Product\_Des{ get; set; }

public float Product\_Price { get; set; }

public float Product\_Tax { get; set; }

}

}

namespace customerService.Models

{

public class Order\_Products

{

[ForeignKey("Order\_Id")]

public virtual Order Order\_ref { get; set; }

[ForeignKey("Product\_Id")]

public virtual Product Product\_ref { get; set; }

}

}

namespace customerService.Models

{

public class Order\_Products

{

[ForeignKey("Order\_Id")]

public virtual Order Order\_ref { get; set; }

[ForeignKey("Product\_Id")]

public virtual Product Product\_ref { get; set; }

}

}

[HttpGet("{id}")]

public async Task<IActionResult> GetProduct([FromRoute] int Id)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var product = await \_context.Products.FindAsync(Id);

if (product == null)

{

return NotFound();

}

Order OrderUser = new Order();

OrderUser.Order\_Idd = 1;

OrderUser.Order\_Status = "COMPLETE";

OrderUser.Total\_Items = 1;

OrderUser.Total\_Price = product.Product\_Price;

OrderUser.Total\_Sum\_Tax = product.Product\_Tax;

OrderUser.Total\_Tax = OrderUser.Total\_Sum\_Tax;

await PostOrder(OrderUser);

return Ok(product);

}

POSTING VALUES

// GET: api/Orders/5/////////////for getting product data

[HttpGet("add")]

public async Task<IActionResult> GetProduct([FromBody] List<int> Id)

{

int totalItems = 0;float price = 0;float sumTax = 0;Product PID = new Product() ;

Order\_Products reference = new Order\_Products();

//lastId=(from order in \_context.Orders

// select order.Order\_Idd).Max();

//lastId++;

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

foreach (int item in Id)

{

var product = await \_context.Products.FindAsync(item);

if (product == null)

{

return NotFound();

}

PID = product;

totalItems++;

price += product.Product\_Price;

sumTax += product.Product\_Tax;

}

Order OrderUser = new Order();

OrderUser.Order\_Status = "Inprogress";

OrderUser.Total\_Items=totalItems;

OrderUser.Total\_Price=price;

OrderUser.Total\_Sum\_Tax=sumTax;

OrderUser.Total\_Tax=sumTax;

reference.Order\_ref = OrderUser;

reference.Product\_ref = PID;

await PostOrder(OrderUser);

await PostOrder\_Products(reference);

return Ok();

}

WRITE FUNCTION

[HttpGet("add")]

public async Task<IActionResult> GetProduct([FromBody] List<int> Id)

{

int totalItems = 0;float price = 0;float sumTax = 0;

List<Product> PID = new List<Product>() ;

//lastId=(from order in \_context.Orders

// select order.Order\_Idd).Max();

//lastId++;

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

foreach (int item in Id)

{

var product = await \_context.Products.FindAsync(item);

if (product == null)

{

return NotFound();

}

PID.Add(product);

totalItems++;

price += product.Product\_Price;

sumTax += product.Product\_Tax;

}

Order OrderUser = new Order();

OrderUser.Order\_Status = "Inprogress"+PID.Count;

OrderUser.Total\_Items=totalItems;

OrderUser.Total\_Price=price;

OrderUser.Total\_Sum\_Tax=sumTax;

OrderUser.Total\_Tax=sumTax;

await PostOrder(OrderUser);

foreach (var item in PID)

{

Order\_Products reference = new Order\_Products();

reference.Order\_ref = OrderUser;

reference.Product\_ref = item;

await PostOrder\_Products(reference);

}

return Ok();

}