

FoodController.cs

```
using FoodOrderApi.Services;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using FoodOrderApi.Models;
namespace FoodOrderApi.Controllers
    [Route("api/[controller]")]
    [ApiController]
    public class FoodController : ControllerBase
        private readonly DataStore _store;
        public FoodController(DataStore store) {
            _store = store;
        [HttpGet]
        public ActionResult<IEnumerable<FoodItem>> GetAllFoods()
            return Ok(_store.FoodItems);
        }
    }
}
```

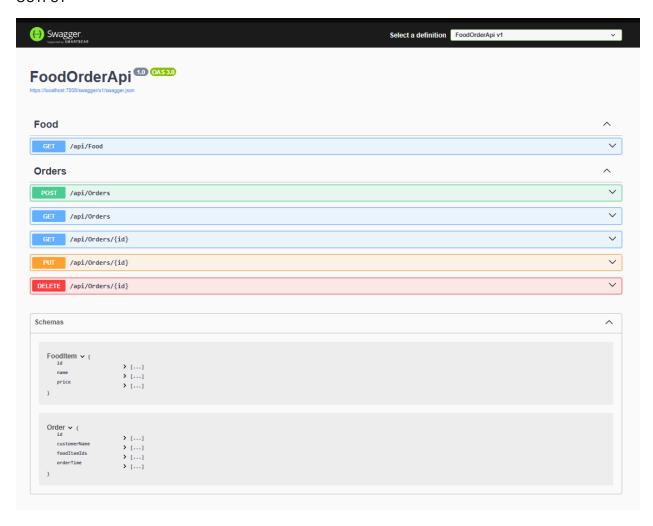
OrdersController.cs

```
using FoodOrderApi.Services;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using FoodOrderApi.Models;
namespace FoodOrderApi.Controllers
    [Route("api/[controller]")]
    [ApiController]
    public class OrdersController : ControllerBase
        private readonly DataStore _store;
        public OrdersController(DataStore store)
            _store = store;
        }
        [HttpPost]
        public ActionResult<Order> PlaceOrder(Order order)
            order.Id = _store.Orders.Count + 1;
            _store.Orders.Add(order);
            return CreatedAtAction(nameof(GetOrder), new { id = order.Id }, order);
        }
        [HttpGet]
        public ActionResult<IEnumerable<Order>> GetAllOrders()
            return Ok(_store.Orders);
        [HttpGet("{id}")]
        public ActionResult<Order> GetOrder(int id)
            var order = _store.Orders.FirstOrDefault(o => o.Id == id);
            return order == null ? NotFound() : Ok(order);
        }
        [HttpPut("{id}")]
        public IActionResult UpdateOrder(int id,Order updatedOrder)
            var order = _store.Orders.FirstOrDefault(o => o.Id == id);
            if (order == null) return NotFound();
            order.CustomerName = updatedOrder.CustomerName;
            order.FoodItemIds = updatedOrder.FoodItemIds;
            return NoContent();
        }
        [HttpDelete("{id}")]
        public IActionResult DeleteOrder (int id)
```

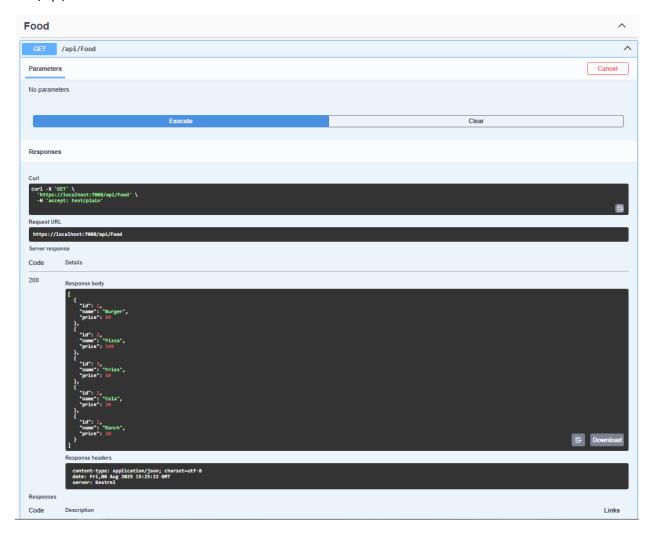
```
{
             var order = _store.Orders.FirstOrDefault(o => o.Id == id);
             if(order == null) return NotFound();
             _store.Orders.Remove(order);
             return NoContent();
         }
    }
}
FoodItem.cs
namespace FoodOrderApi.Models
    public class FoodItem
         public int Id { get; set; }
         public string Name { get; set; } = string.Empty;
         public decimal Price { get; set; }
}
Order.cs
namespace FoodOrderApi.Models
    public class Order
         public int Id { get; set; }
         public string CustomerName { get; set; } = string.Empty;
         public List<int> FoodItemIds { get; set; } = new();
         public DateTime OrderTime { get; set; } = DateTime.Now;
    }
}
DataStore.cs
using FoodOrderApi.Models;
namespace FoodOrderApi.Services
    public class DataStore
         public List<FoodItem> FoodItems { get; set; } = new()
             new FoodItem {Id = 1, Name = "Burger", Price = 99},
             new FoodItem {Id = 2, Name = "Pizza", Price = 199},
new FoodItem {Id = 3, Name = "Fries", Price = 49},
new FoodItem {Id = 2, Name = "Cola", Price = 30},
             new FoodItem {Id = 2, Name = "Ranch", Price = 20}
         };
         public List<Order> Orders { get; set; } = new();
    }
}
```

```
Program.cs
var builder = WebApplication.CreateBuilder(args);
// Add services to the container.
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddSingleton<FoodOrderApi.Services.DataStore>();
var app = builder.Build();
// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
    app.UseSwagger();
    app.UseSwaggerUI();
}
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

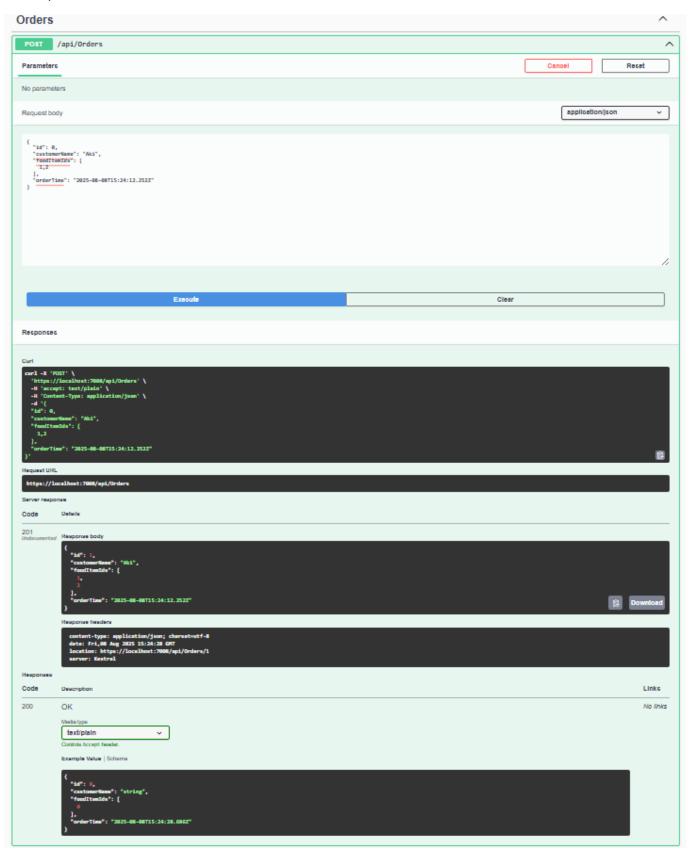
OUTPUT



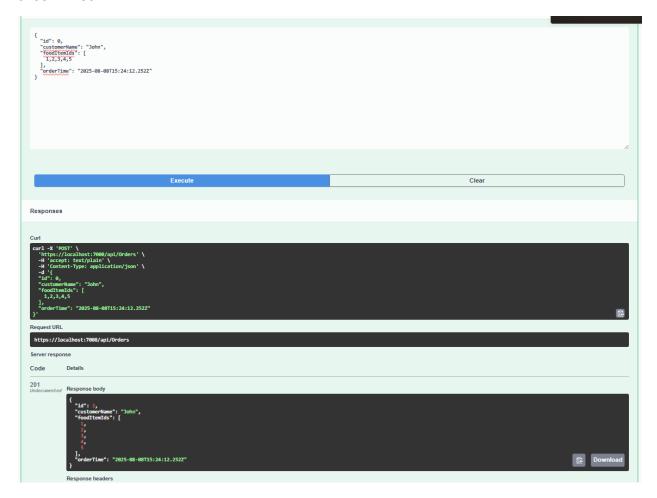
GET/api/Food



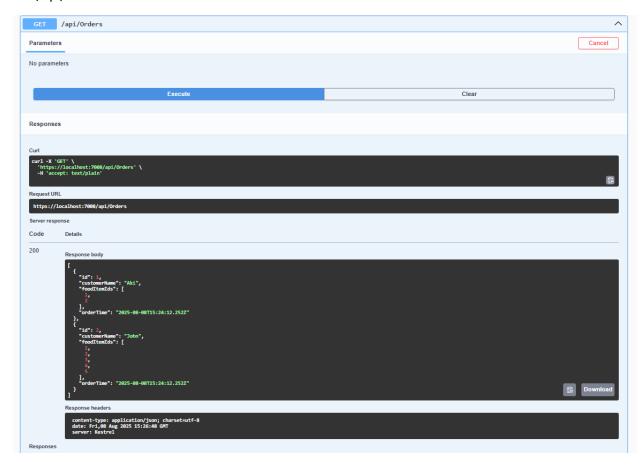
POST/api/Orders



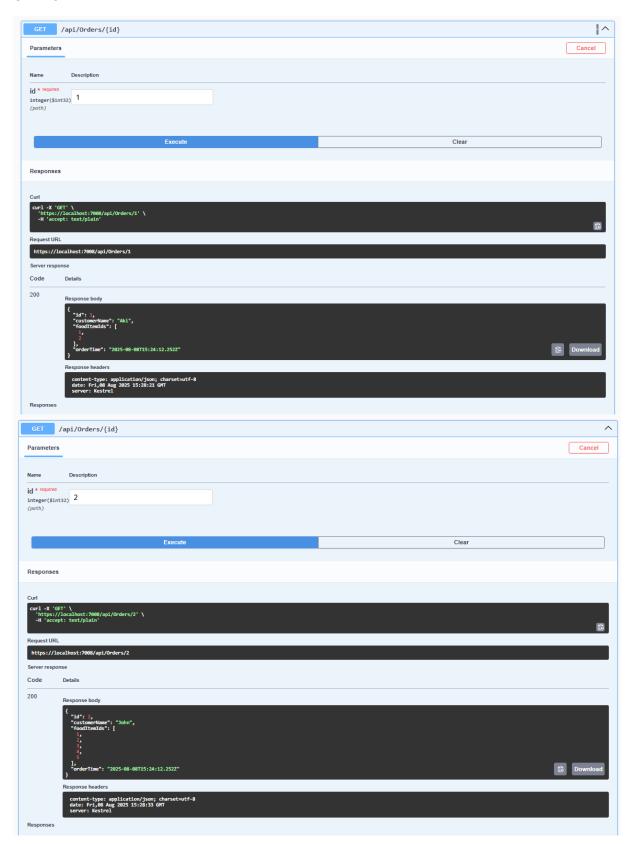
SECOND POST



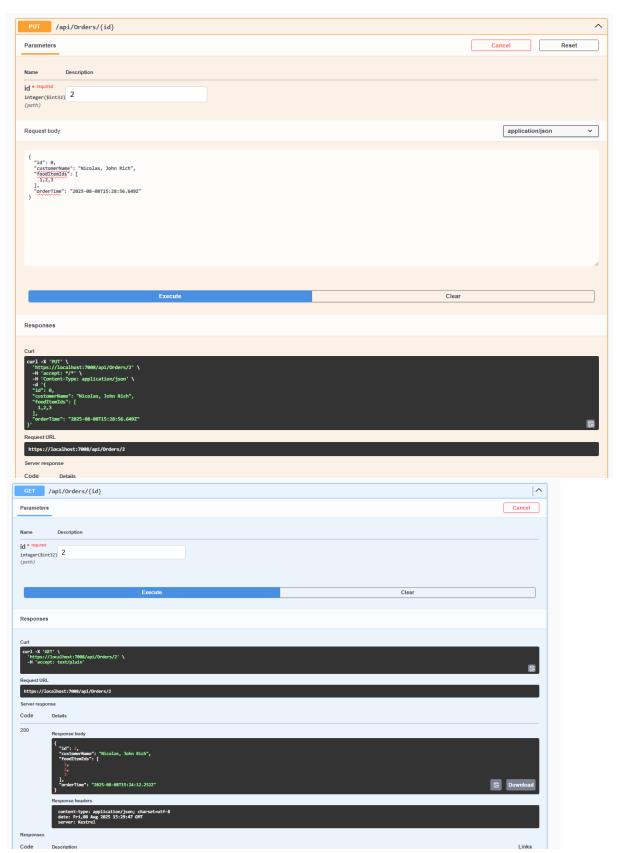
GET/api/Orders



GET IDS



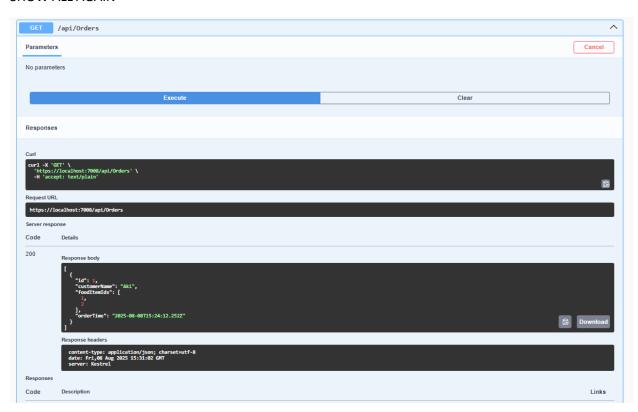
PUT



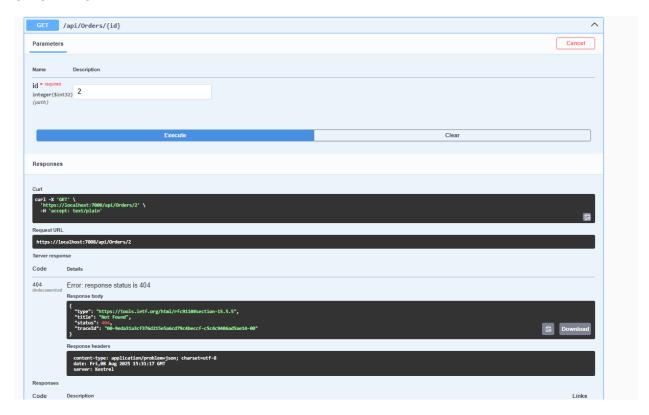
DELETE



SHOW ALL AGAIN



CHECK ID NG NADELETE



REFLECTION

In my Food Order API project, I followed the instructions given by our professor and used the provided files to build the system. The code was organized into different parts using functions that handled tasks like showing the menu, placing and updating orders, checking if the order is valid, and viewing or deleting existing orders. The FoodController is used to show the list of available food items, while the OrdersController handles everything related to customer orders. This made the code easier to read and understand because each part had its own role.

The API works by sending and receiving data through HTTP requests. I used GET requests to view the food list and orders, and POST requests to add new orders. The data is sent in JSON format and automatically turned into C# objects by the API. I tested the endpoints using Swagger, which made it easier to check if the requests were working and see the results without needing a separate website.

One thing I noticed is that whenever I restart the application, all the orders are gone. That's because the data is stored only in memory, and not saved permanently in a database. So every time the app runs again, it starts fresh.

A challenge I faced while doing this activity was getting build errors because of missing modules. The program showed CS0246 errors saying it couldn't find things like Order, FoodItem, and DataStore. I realized I forgot to add the correct using statements at the top of my files. Once I added lines like using FoodOrderApi.Models; and using FoodOrderApi.Services;, the errors were gone and the project worked properly. This taught me how important it is to double-check all references when working with multiple files.