Piyush Sharma:

Restaurant API Application with Angular UI:

1.First Creates an Models related to your requirement and add migration:

namespace RestaurantAPI.Models

{

public class Item

{

public int MenuID { get; set; }

public int ItemID { get; set; }

public string Name { get; set; }

public int Price { get; set; }

}

}

2.Create Controllers to acreate your APIs operations:

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using RestaurantAPI.Data;

using RestaurantAPI.Models;

namespace RestaurantAPI.Controllers

{

[ApiController]

[Route("[controller]")]

public class ItemsController : Controller

{

private readonly RestaurantDbContext restaurantDbContext;

public ItemsController(RestaurantDbContext restaurantDbContext)

{

this.restaurantDbContext = restaurantDbContext;

}

[HttpGet]

[Route("{id}")]

public async Task<IActionResult> GetAllItems(int id)

{

var item = await (from i in restaurantDbContext.items

where i.MenuID == id

select new Item()

{

ItemID = i.ItemID,

MenuID = i.MenuID,

Name = i.Name,

Price = i.Price,

}

).ToListAsync();

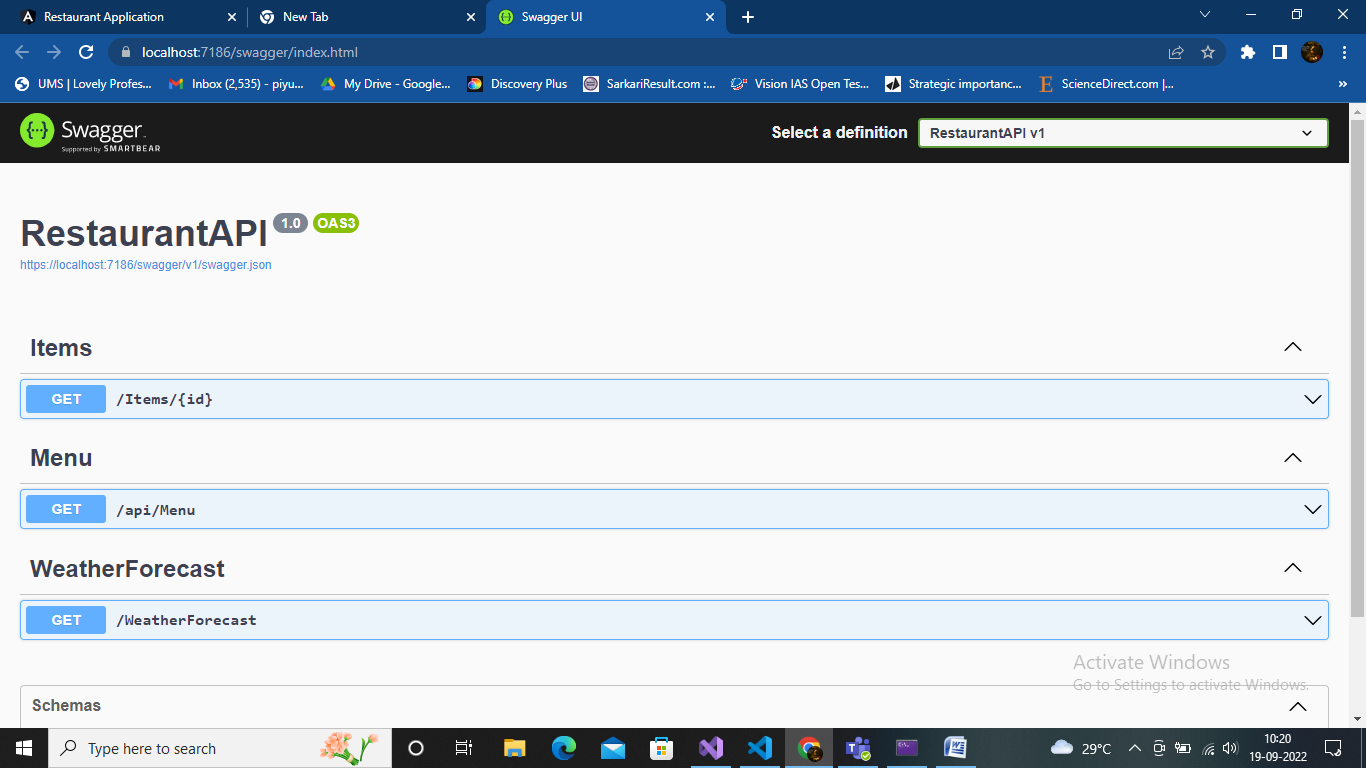
return Ok(item);

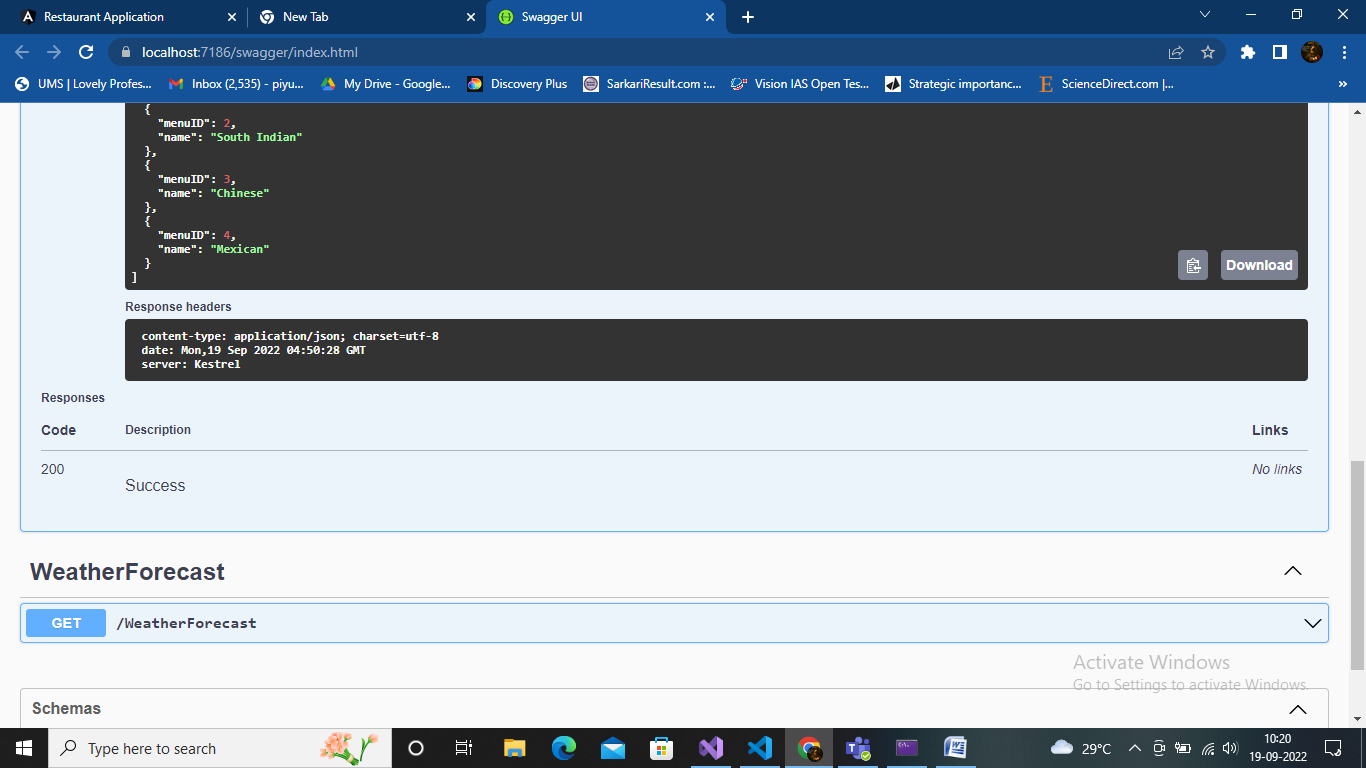
}

}

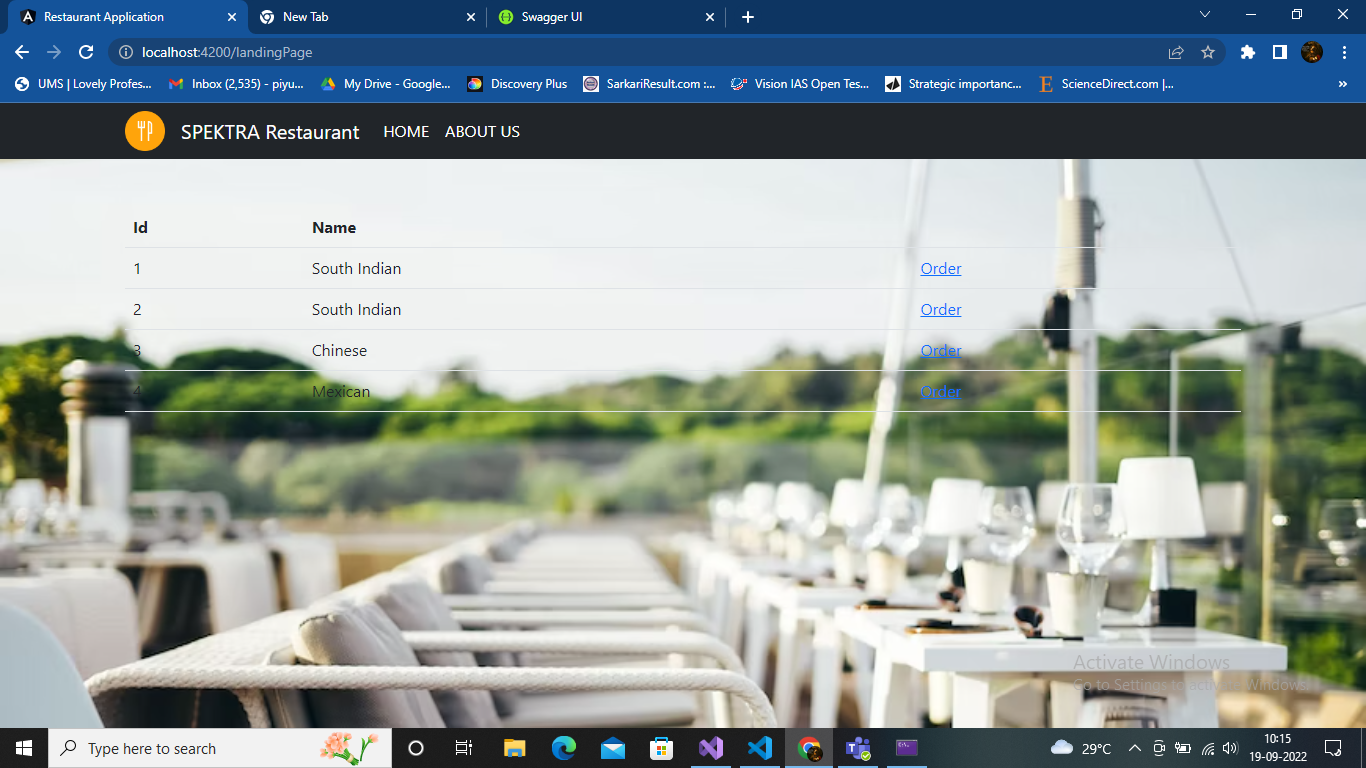
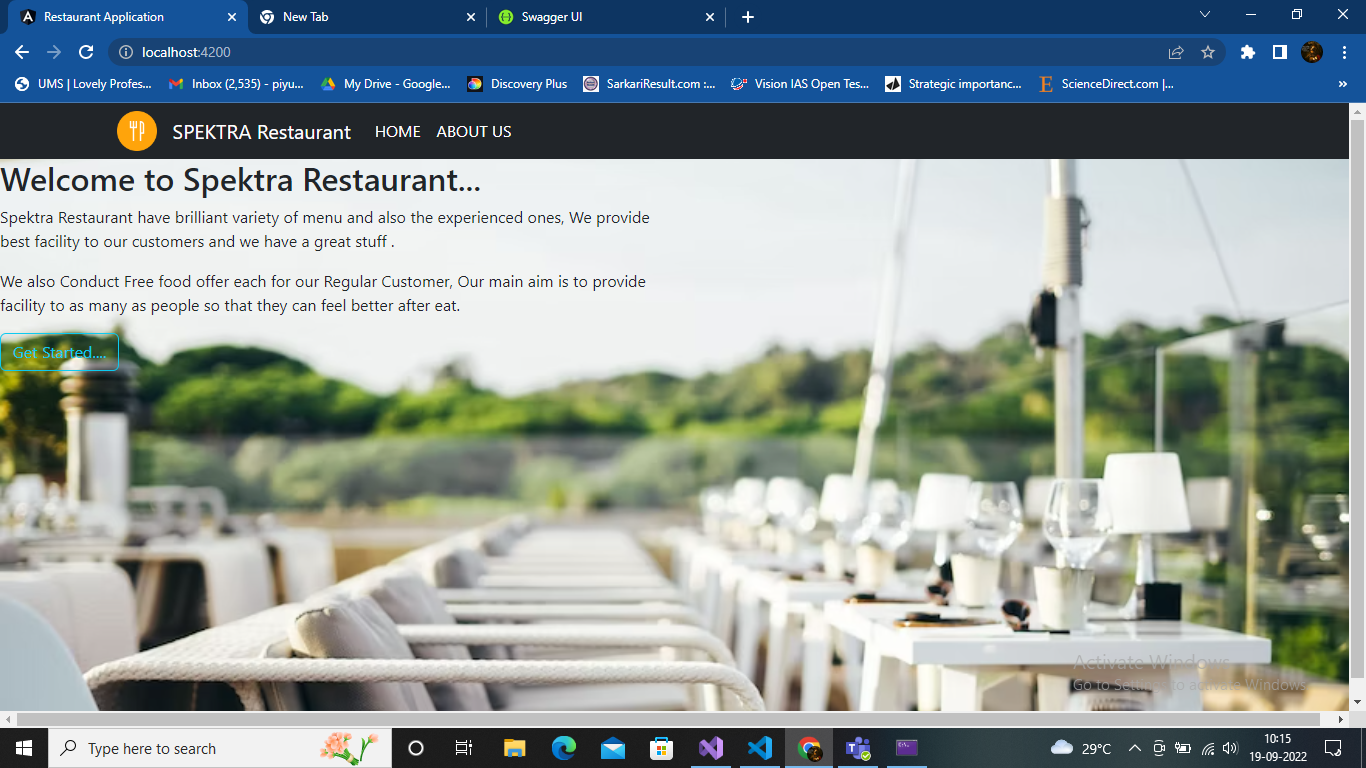
}

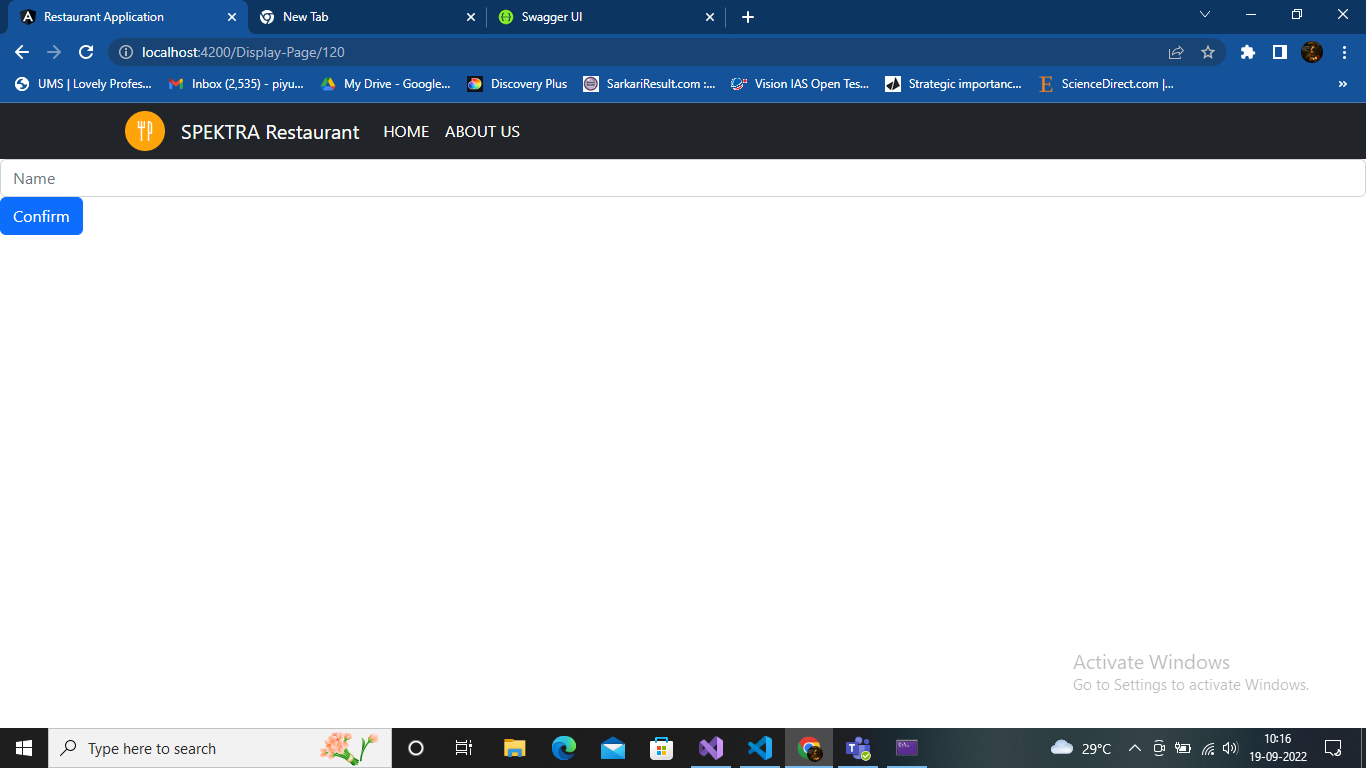
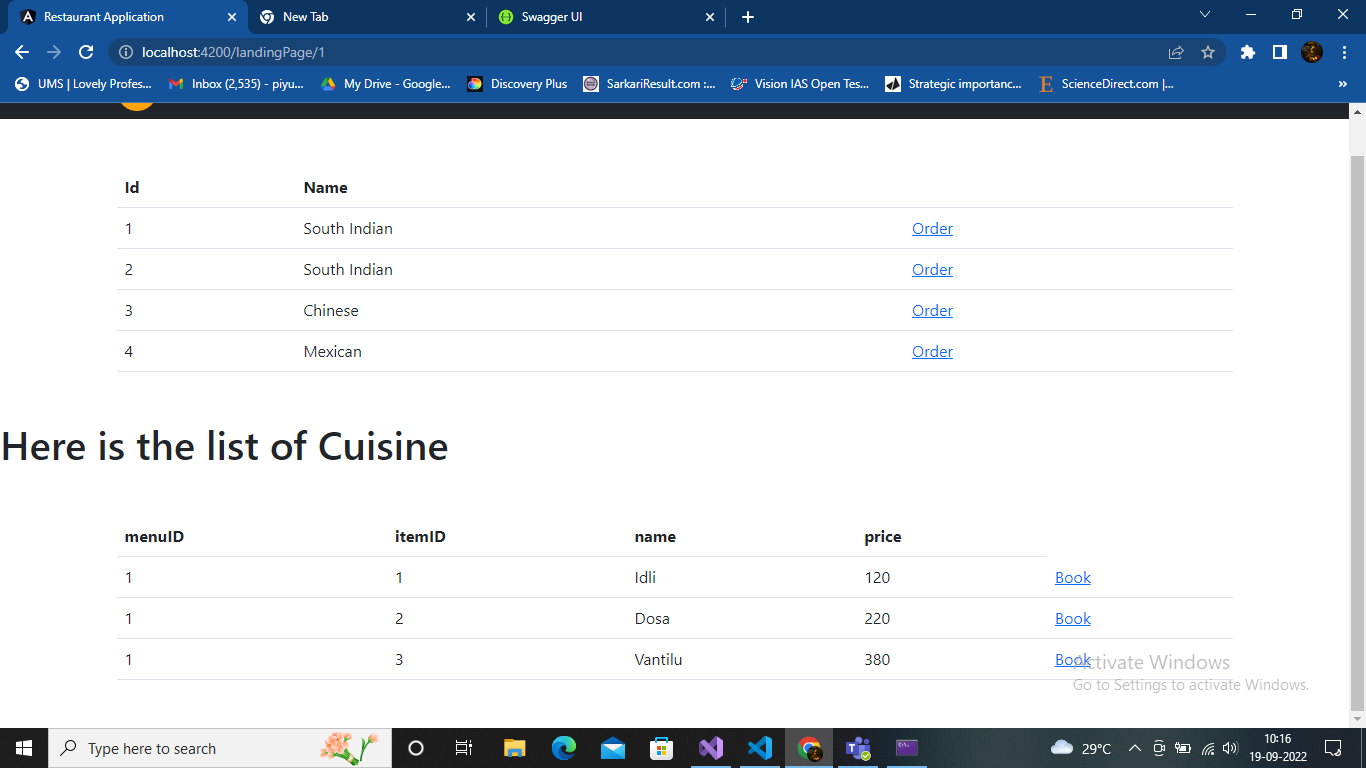
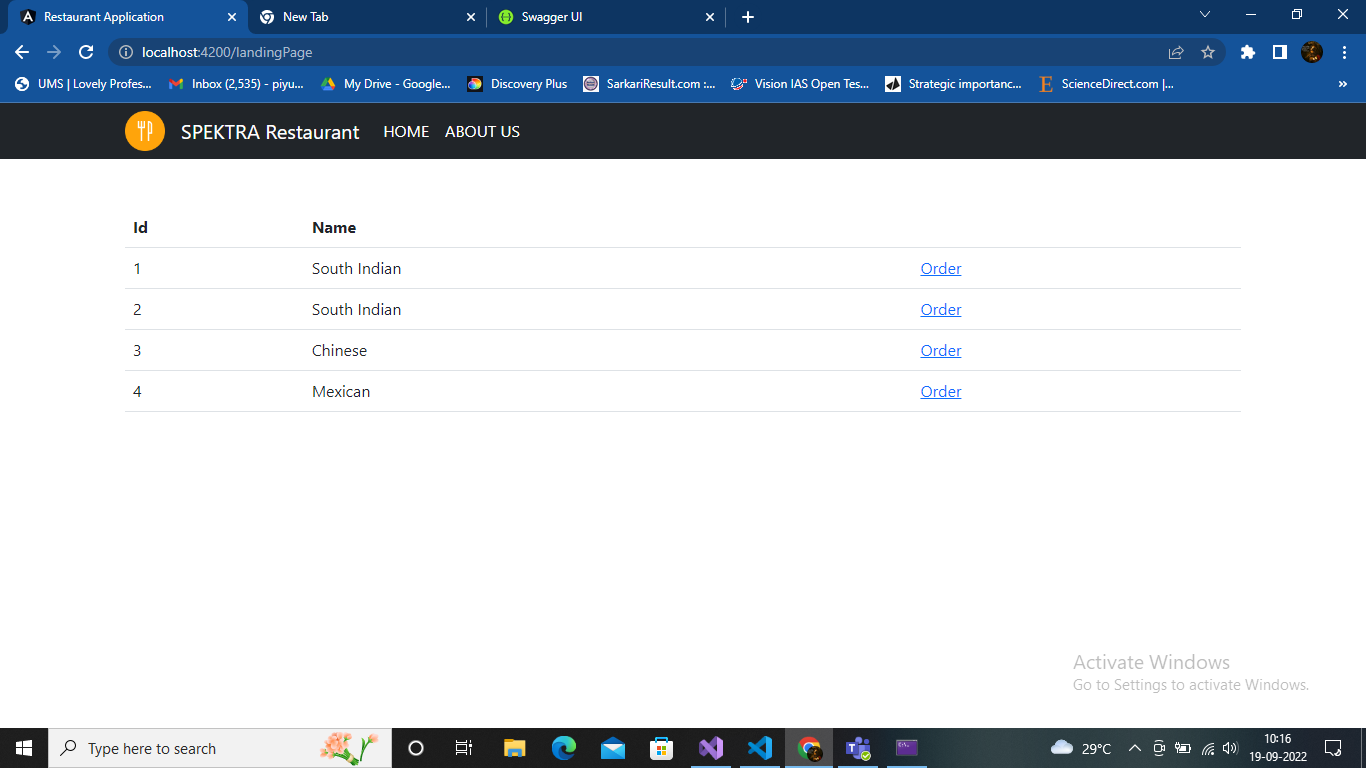
Screenshot of your APIs





Screenshots of your Angular UI





Angular Code:

Create Models

export interface Items{

    menuID:number;

    itemID:number;

    name : string;

    price :number;

}

And creates components according to your requirement and

This is the one of the component code of html:

h1>Here is the list of Cuisine</h1>

<div class="container my-5">

    <h1 class="mb-3"></h1>

    <table class="table">

        <thead>

            <tr>

                <th>menuID</th>

                <th>itemID</th>

                <th> name</th>

                <th>price</th>

            </tr>

        </thead>

        <tbody>

             <tr \*ngFor="let m of item">

                 <td>{{m.menuID}}</td>

                 <td>{{m.itemID}}</td>

                 <td>{{m.name}}</td>

                 <td>{{m.price}}</td>

                 <td><a [routerLink]="['/Display-Page/'+ m.price]" >Book</a></td>

                 </tr>

        <!-- </tbody> -->

    </table>

  </div>

  <router-outlet></router-outlet>

And creates Services and makes required changes:

Thank you;