Troubleshooting-Diary E-Commerce System

Case : Add new endpoints to existing controllers

```
ProductController: Add endpoints to get products by Category or Supplier.

UserController: Add endpoints to get users by Role or Email.
```

1. ProductController: Add GetProductsByCategory and GetProductsBySupplier endpoints

```sql
[HttpGet("ByCategory/{categoryId}")]
    public IActionResult GetProductsByCategory(int categoryId)
    {
      try
      {
        var products = _productService.GetProductsByCategory(categoryId);
        var productDTOs = _mapper.Map<List<ProductDto>>(products);
        return Ok(productDTOs);
      }
      catch (Exception ex)
      {
        return StatusCode(500, $"An error occurred while retrieving products by category. {ex.Message}");
      }
    }
    [HttpGet("BySupplier/{supplierId}")]
    public IActionResult GetProductsBySupplier(int supplierId)
```

```
        {
            try
            {
                var products = _productService.GetProductsBySupplier(supplierId);
                var productDTOs = _mapper.Map<List<ProductDto>>(products);
                return Ok(productDTOs);
            }
            catch (Exception ex)
            {
                return StatusCode(500, $"An error occurred while retrieving products by supplier. {ex.Message}");
            }
        }
```

2. UserController: Add GetUsersByRole and GetUserByEmail endpoints

```sql
[HttpGet("ByRole/{role}")]
        public IActionResult GetUsersByRole(string role)
        {
            try
            {
                var users = _userService.GetUsersByRole(role);
                var userDTOs = _mapper.Map<List<UserDto>>(users);
                return Ok(userDTOs);
            }
            catch (Exception ex)
            {
                return StatusCode(500, $"An error occurred while retrieving users by role. {ex.Message}");
            }
        }
        [HttpGet("ByEmail")]
        public IActionResult GetUserByEmail(string email)
        {
            try
            {
                var user = _userService.GetUserByEmail(email);
                if (user == null)
                {
                    return NotFound($"User with email {email} not found.");
                }
                var userDTO = _mapper.Map<UserDto>(user);
                return Ok(userDTO);
            }
            catch (Exception ex)
            {
                return StatusCode(500, $"An error occurred while retrieving user by email. {ex.Message}");
            }
        }
```

- In above examples, new endpoints are added to the ProductController and UserController to retrieve products by category or supplier, and users by role or email, respectively.

- These endpoints use the service layer to fetch the data and AutoMapper to convert entities to DTOs before returning them in the response.

Implement Pagination

```

- Add pagination parameters (page number, page size) to relevant endpoints.

- Modify service methods to return paginated results.

```

1. Modify GetAllProducts endpoint in ProductController to support pagination

```sql
[HttpGet]
    public IActionResult GetAllProducts(int pageNumber = 1, int pageSize = 10)
    {
        try
        {
            var products = _productService.GetAllProducts(pageNumber, pageSize);
            var productDTOs = _mapper.Map<List<ProductDto>>(products);
            return Ok(productDTOs);
        }
        catch (Exception ex)
        {
            return StatusCode(500, $"An error occurred while retrieving products.
{ex.Message}");
        }
    }
```

2. Modify GetAllProducts method in ProductService to handle pagination

```sql
public List<Product> GetAllProducts(int pageNumber, int pageSize)
```

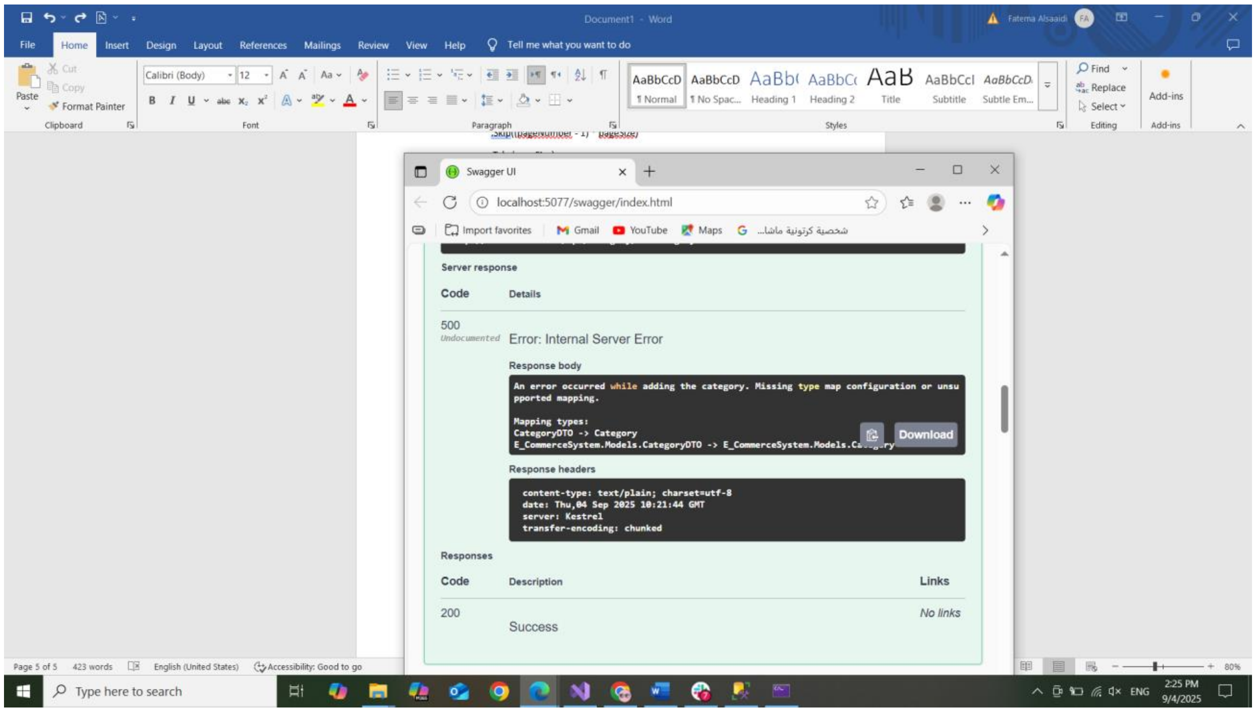
```
    {
        return _productRepo.GetAllProducts()
                            .Skip((pageNumber - 1) * pageSize)
                            .Take(pageSize)
                            .ToList();
    }
```

- In above examples, the GetAllProducts endpoint in the ProductController is modified to accept pagination parameters (pageNumber and pageSize).

- The ProductService's GetAllProducts method is updated to use LINQ's Skip and Take methods to return a paginated list of products based on the provided parameters.

- This approach allows clients to request specific pages of data, improving performance and usability for large datasets.

Error



Server response

Code	Details
500 *Undocumented*	Error: Internal Server Error

Response body

```
An error occurred while adding the category. Missing type map configuration or unsu
pported mapping.

Mapping types:
CategoryDTO -> Category
E_CommerceSystem.Models.CategoryDTO -> E_CommerceSystem.Models.Category
```

Response headers

```
content-type: text/plain; charset=utf-8
date: Thu,04 Sep 2025 10:21:44 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code	Description	Links
200	Success	*No links*

Cuase:

calling AutoMapper to map CategoryDTO → Category, but there’s no mapping profile registered

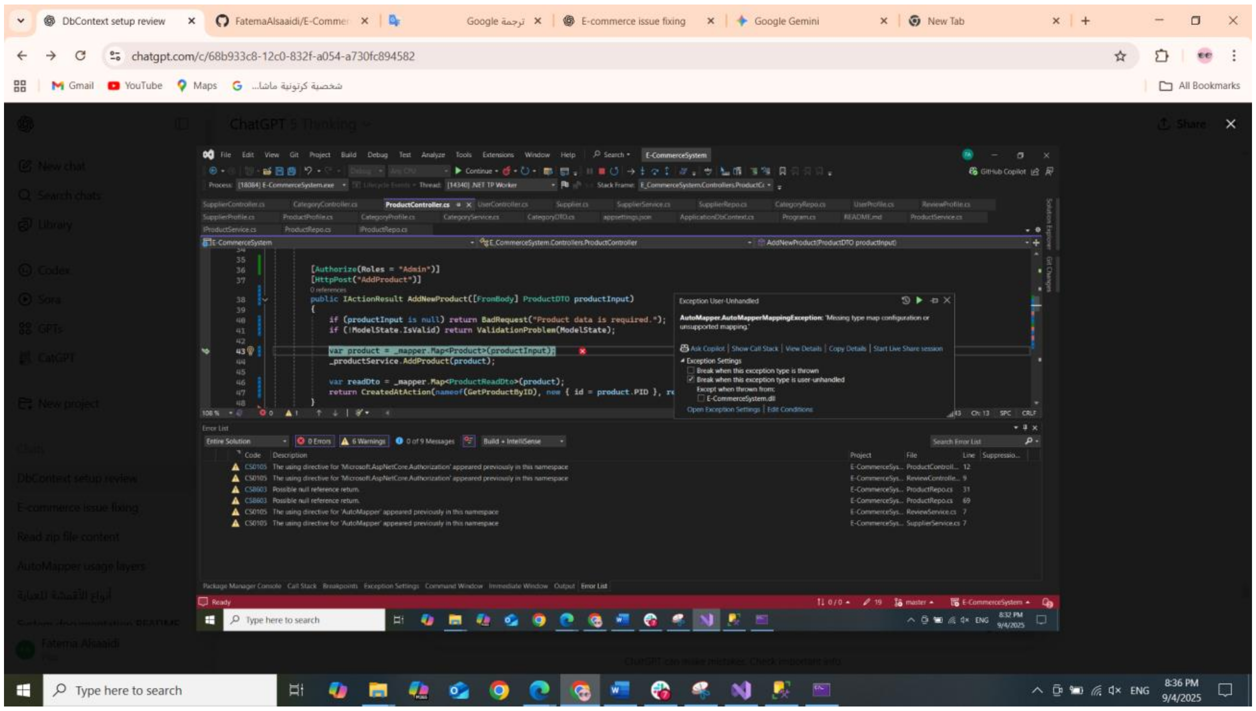
Solve :

```
using AutoMapper;
using E_CommerceSystem.Models;
using static E_CommerceSystem.Models.CategoryDTO;

namespace E_CommerceSystem.Mapping
{
    public class CategoryProfile : Profile
    {
        public CategoryProfile()
        {
            // DTO (class) -> Entity
            CreateMap<CategoryDTO, Category>()
                .ForMember(d => d.CategoryId, opt => opt.Ignore()); // ignore on create/update

            // Entity -> Read DTO (record)
            CreateMap<Category, CategoryReadDto>();
        }
    }
}
```

Error



```
[Authorize(Roles = "Admin")]
[HttpPost("AddProduct")]
0 references
public IActionResult AddNewProduct([FromBody] ProductDTO productInput)
{
    if (productInput is null) return BadRequest("Product data is required.");
    if (!ModelState.IsValid) return ValidationProblem(ModelState);

    var product = _mapper.Map<Product>(productInput);
    _productService.AddProduct(product);

    var readDto = _mapper.Map<ProductReadDto>(product);
    return CreatedAtAction(nameof(GetProductByID), new { id = product.PID }, re
}
```

```
Exception User-Unhandled

AutoMapper.AutoMapperMappingException: 'Missing type map configuration or
unsupported mapping.'
```

Exception Settings
- [ ] Break when this exception type is thrown
- [x] Break when this exception type is user-unhandled

Except when thrown from:
- [ ] E-CommerceSystem.dll

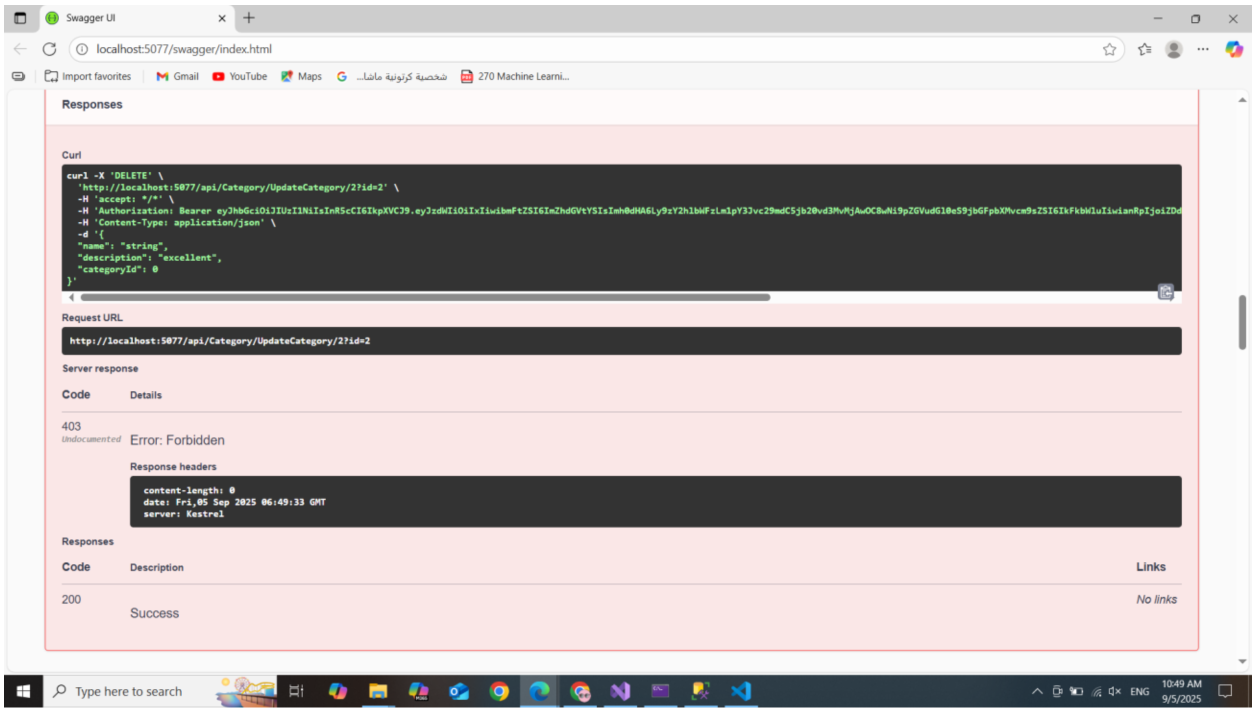
Error List: Entire Solution — 0 Errors — 6 Warnings — 0 of 9 Messages — Build + IntelliSense

```
Code    Description                                                                      Project            File                 Line
CS0105  The using directive for 'Microsoft.AspNetCore.Authorization' appeared previously in this namespace  E-CommerceSys...  ProductControll...  12
CS0105  The using directive for 'Microsoft.AspNetCore.Authorization' appeared previously in this namespace  E-CommerceSys...  ReviewControlle...  9
CS8603  Possible null reference return.                                                  E-CommerceSys...   ProductRepo.cs       31
CS8603  Possible null reference return.                                                  E-CommerceSys...   ProductRepo.cs       69
CS0105  The using directive for 'AutoMapper' appeared previously in this namespace       E-CommerceSys...   ReviewService.cs     7
CS0105  The using directive for 'AutoMapper' appeared previously in this namespace       E-CommerceSys...   SupplierService.cs   7
```

Solve :

Check dto in create part of code if there is missing attribute should add it.

Error :



Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5077/api/Category/UpdateCategory/2?id=2' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxIiwibmFtZSI6ImZhdGVtYSIsImh0dHA6Ly9zY2hlbWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9pZGVudGl0eS9jbGFpbXMvcm9sZSI6IkFkbWluIiwianRpIjoiZDd
  -H 'Content-Type: application/json' \
  -d '{
  "name": "string",
  "description": "excellent",
  "categoryId": 0
}'
```

Request URL

```
http://localhost:5077/api/Category/UpdateCategory/2?id=2
```

Server response

Code	Details
403 *Undocumented*	Error: Forbidden

Response headers

```
content-length: 0
date: Fri,05 Sep 2025 06:49:33 GMT
server: Kestrel
```

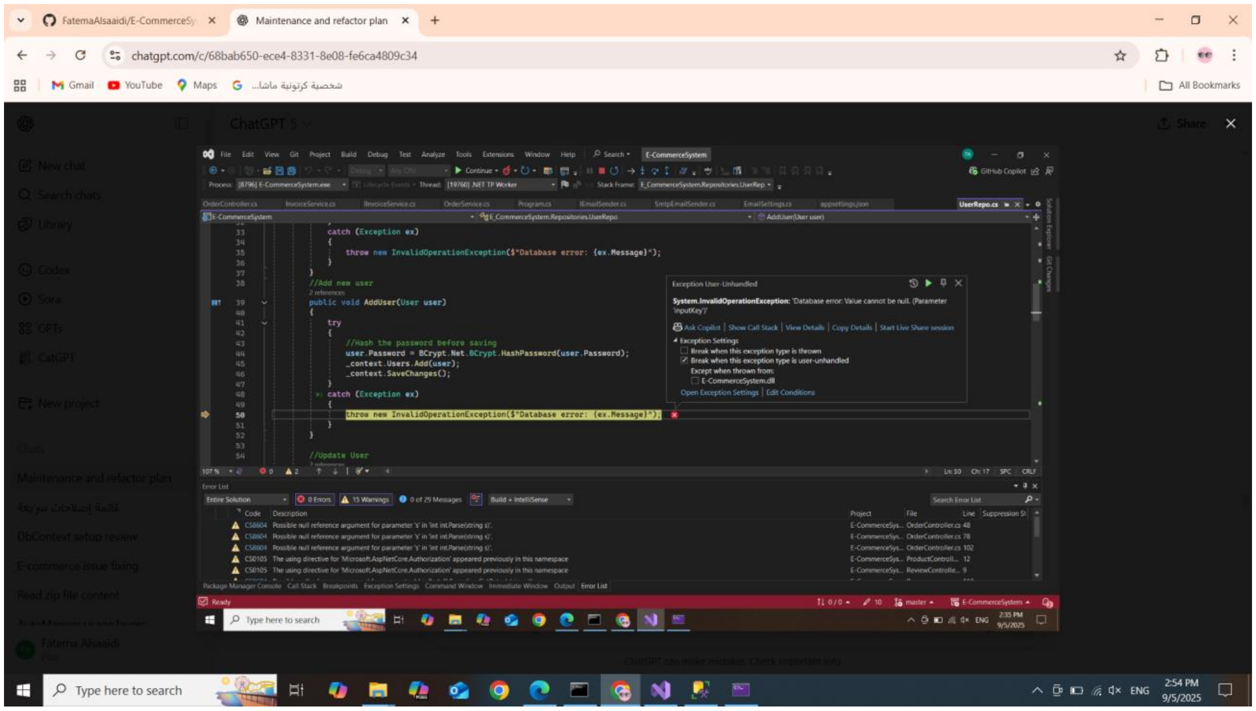
Responses

Code	Description	Links
200	Success	*No links*

Solve :

Check authorize “Admin”not “admin”

Error:



```
            catch (Exception ex)
            {
                throw new InvalidOperationException($"Database error: {ex.Message}");
            }
        }
        //Add new user
        public void AddUser(User user)
        {
            try
            {
                //Hash the password before saving
                user.Password = BCrypt.Net.BCrypt.HashPassword(user.Password);
                _context.Users.Add(user);
                _context.SaveChanges();
            }
            catch (Exception ex)
            {
                throw new InvalidOperationException($"Database error: {ex.Message}");
            }
        }

        //Update User
```

Exception User-Unhandled

System.InvalidOperationException: 'Database error: Value cannot be null. (Parameter 'inputKey')'

Exception Settings
- [ ] Break when this exception type is thrown
- [x] Break when this exception type is user-unhandled
  - Except when thrown from:
  - [ ] E-CommerceSystem.dll

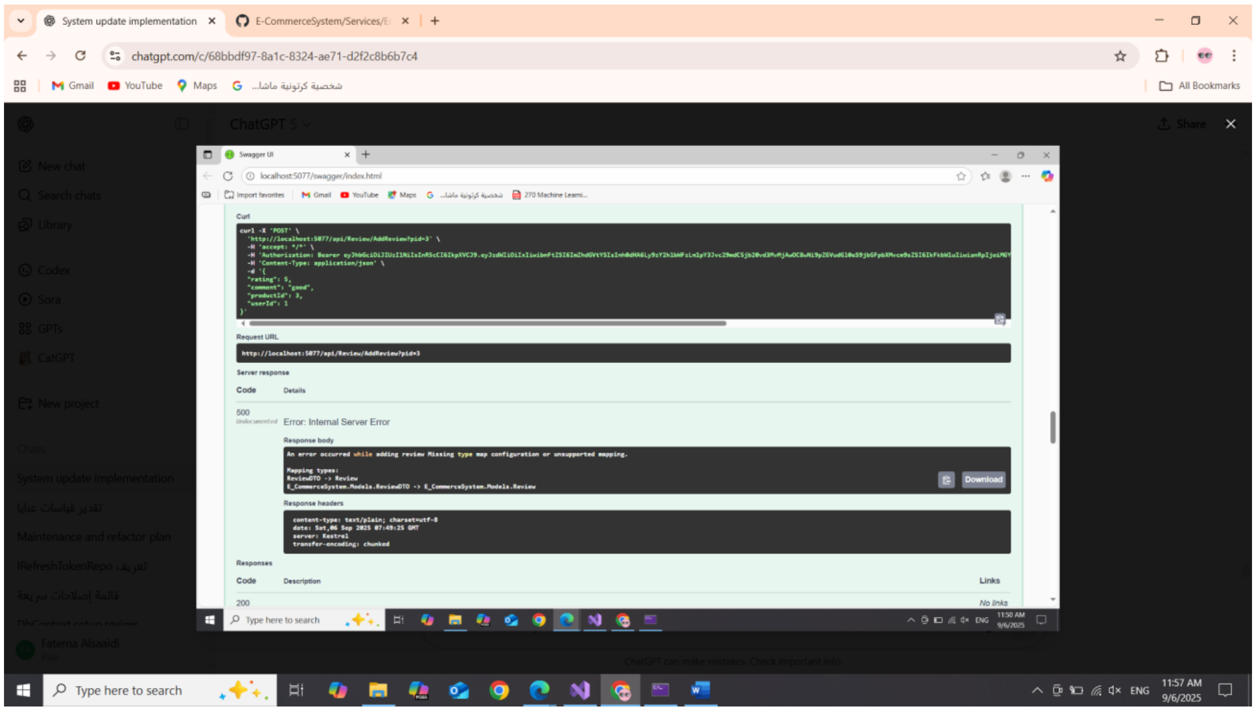
Error List: 0 Errors, 15 Warnings, 0 of 29 Messages

Code	Description	Project	File	Line
CS8604	Possible null reference argument for parameter 's' in 'int int.Parse(string s)'.	E-CommerceSys...	OrderController.cs	48
CS8604	Possible null reference argument for parameter 's' in 'int int.Parse(string s)'.	E-CommerceSys...	OrderController.cs	78
CS8604	Possible null reference argument for parameter 's' in 'int int.Parse(string s)'.	E-CommerceSys...	OrderController.cs	102
CS0105	The using directive for 'Microsoft.AspNetCore.Authorization' appeared previously in this namespace	E-CommerceSys...	ProductControll...	12
CS0105	The using directive for 'Microsoft.AspNetCore.Authorization' appeared previously in this namespace	E-CommerceSys...	ReviewControlle...	9

Solve : sure input role value

public record UserRegisterDto(string UName, string Email, string Password, string? Phone, string role);

Error:



Curl

```
curl -X 'POST' \
  'http://localhost:5077/api/Review/AddReview?pid=3' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxIiwibmFtZSI6ImZhdGVtYSIsImh0dHA6Ly9zY2hlbWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9pZGVudGl0eS9jbGFpbXMvcm9sZSI6IkFkbWluIiwianRpIjoiMGY
  -H 'Content-Type: application/json' \
  -d '{
  "rating": 5,
  "comment": "good",
  "productId": 3,
  "userId": 1
}'
```

Request URL

```
http://localhost:5077/api/Review/AddReview?pid=3
```

Server response

Code: 500 Undocumented

Details: Error: Internal Server Error

Response body

```
An error occurred while adding review Missing type map configuration or unsupported mapping.

Mapping types:
ReviewDTO -> Review
E_CommerceSystem.Models.ReviewDTO -> E_CommerceSystem.Models.Review
```

Response headers

```
 content-type: text/plain; charset=utf-8
 date: Sat,06 Sep 2025 07:49:25 GMT
 server: Kestrel
 transfer-encoding: chunked
```

Responses

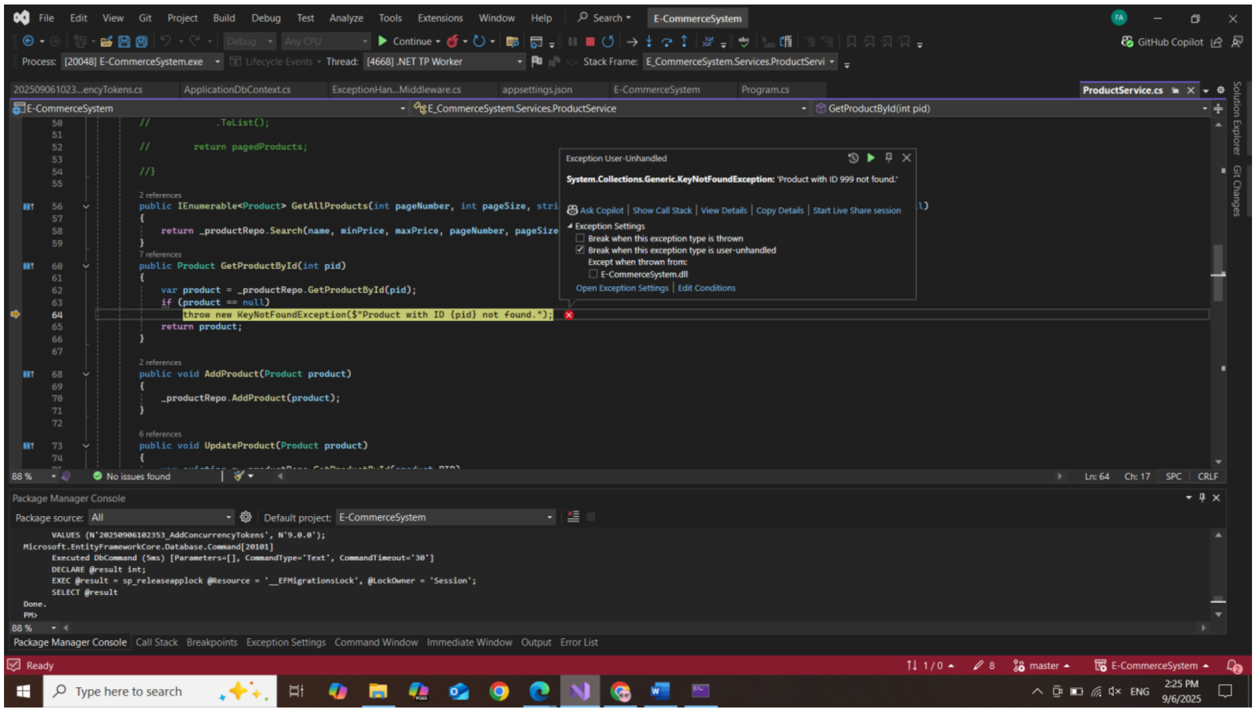
Code	Description	Links
200		No links

Solve :

Add in review profile

```
CreateMap<ReviewDTO, Review>()
.ForMember(d => d.ReviewID, o => o.Ignore())
.ForMember(d => d.PID, o => o.MapFrom((src, dest, _, ctx) => (int)ctx.Items["pid"]))
.ForMember(d => d.UID, o => o.MapFrom((src, dest, _, ctx) => (int)ctx.Items["uid"]))
.ForMember(d => d.ReviewDate, o => o.MapFrom(_ => DateTime.UtcNow));
```

Error:



```
//            .ToList();

//        return pagedProducts;

//}

2 references
public IEnumerable<Product> GetAllProducts(int pageNumber, int pageSize, stri
{
    return _productRepo.Search(name, minPrice, maxPrice, pageNumber, pageSize
}
7 references
public Product GetProductById(int pid)
{
    var product = _productRepo.GetProductById(pid);
    if (product == null)
        throw new KeyNotFoundException($"Product with ID {pid} not found.");
    return product;
}

2 references
public void AddProduct(Product product)
{
    _productRepo.AddProduct(product);
}

6 references
public void UpdateProduct(Product product)
{
```

Exception User-Unhandled

System.Collections.Generic.KeyNotFoundException: 'Product with ID 999 not found.'

Exception Settings
- [ ] Break when this exception type is thrown
- [x] Break when this exception type is user-unhandled

Except when thrown from:
- [ ] E-CommerceSystem.dll

Package Manager Console

```
        VALUES (N'20250906102353_AddConcurrencyTokens', N'9.0.0');
Microsoft.EntityFrameworkCore.Database.Command[20101]
        Executed DbCommand (5ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
        DECLARE @result int;
        EXEC @result = sp_releaseapplock @Resource = '__EFMigrationsLock', @LockOwner = 'Session';
        SELECT @result
Done.
PM>
```

Causes:

Allow the services to throw logical exceptions (such as KeyNotFoundException when a product is not found).

Prevent the controllers from using general try/catch blocks.

Add a unified error handling middleware that automatically converts exceptions into JSON format (ProblemDetails) with an appropriate HTTP status code (400/404/etc.).

In Visual Studio, disable "Break on all exceptions" so that the debugger doesn't stop at every exception that can be handled (and you can continue execution).

Solve:

Press continuos

Error:

ASP.NET Core Warning: "Failed to determine the https port for redirect"

Symptom

Console output shows:

```
Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
Failed to determine the https port for redirect.
```

Why this happens

Your app is listening only on HTTP (e.g., http://localhost:5077) but the middleware tries to redirect HTTP to HTTPS. Because no HTTPS URL/port is configured (via Kestrel, launchSettings.json, or ASPNETCORE_URLS), the middleware cannot determine a target and logs this warning.

Fix Option A (Recommended): Enable HTTPS endpoint

1) Trust a local HTTPS developer certificate (once on your machine):

```
dotnet dev-certs https --trust
```

2) Ensure your app listens on HTTPS and HTTP. In Properties/launchSettings.json, set applicationUrl for the project profile, for example:

```
{
  "profiles": {
    "E-CommerceSystem": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "https://localhost:7257;http://localhost:5077",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

3) Keep UseHttpsRedirection() in Program.cs:

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
```

```
var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();   // keep this

app.UseAuthorization();
app.MapControllers();
app.Run();
```

Fix Option B: Only use HTTP (development fallback)

If you do not want HTTPS locally, disable the redirection when no HTTPS port is configured.

```
var builder = WebApplication.CreateBuilder(args);
var httpsPort = builder.Configuration.GetValue<int?>("ASPNETCORE_HTTPS_PORT");

var app = builder.Build();

if (httpsPort.HasValue)
{
    app.UseHttpsRedirection();
}
// Else: no HTTPS endpoint configured, skip redirection

app.MapControllers();
app.Run();
```

Alternatively, guard by environment (e.g., only redirect in Production):

```
if (app.Environment.IsProduction()) app.UseHttpsRedirection();
```