**Ashutosh Dash  
Superset ID – 6365404  
WebAPI - 3**

**Models/Department.cs**

namespace CustomWebApi.Models

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**Models/Skill.cs**

namespace CustomWebApi.Models

{

public class Skill

{

public int Id { get; set; }

public string Name { get; set; }

}

}

**// Models/Employee.cs**

using System;

using System.Collections.Generic;

namespace CustomWebApi.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public Department Department { get; set; }

public List<Skill> Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

**// Filters/CustomAuthFilter.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

namespace CustomWebApi.Filters

{

public class CustomAuthFilter : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext context)

{

var hasAuth = context.HttpContext.Request.Headers.TryGetValue("Authorization", out var token);

if (!hasAuth)

{

context.Result = new BadRequestObjectResult("Invalid request - No Auth token");

return;

}

if (!token.ToString().Contains("Bearer"))

{

context.Result = new BadRequestObjectResult("Invalid request - Token present but Bearer unavailable");

return;

}

base.OnActionExecuting(context);

}

}

}

**// Filters/CustomExceptionFilter.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using System;

using System.IO;

namespace CustomWebApi.Filters

{

public class CustomExceptionFilter : IExceptionFilter

{

public void OnException(ExceptionContext context)

{

string logPath = Path.Combine(Directory.GetCurrentDirectory(), "Logs");

Directory.CreateDirectory(logPath);

string logFile = Path.Combine(logPath, "exceptions.txt");

File.AppendAllText(logFile, $"[{DateTime.Now}] {context.Exception.Message}{Environment.NewLine}");

context.Result = new ObjectResult("An unexpected error occurred.")

{

StatusCode = 500

};

}

}

}

**// Controllers/EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using CustomWebApi.Models;

using CustomWebApi.Filters;

using System.Collections.Generic;

namespace CustomWebApi.Controllers

{

[ApiController]

[Route("api/[controller]")]

[ServiceFilter(typeof(CustomAuthFilter))]

public class EmployeeController : ControllerBase

{

private readonly List<Employee> \_employees;

public EmployeeController()

{

\_employees = GetStandardEmployeeList();

}

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status500InternalServerError)]

public ActionResult<List<Employee>> GetStandard()

{

//throw new Exception("Test Exception"); // Uncomment to test custom exception filter

return Ok(\_employees);

}

[HttpPost]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public IActionResult CreateEmployee([FromBody] Employee employee)

{

if (employee == null)

return BadRequest("Employee object is null");

\_employees.Add(employee);

return Ok(employee);

}

[HttpPut("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status404NotFound)]

public IActionResult UpdateEmployee(int id, [FromBody] Employee updatedEmployee)

{

var emp = \_employees.Find(e => e.Id == id);

if (emp == null)

return NotFound();

emp.Name = updatedEmployee.Name;

emp.Salary = updatedEmployee.Salary;

emp.Permanent = updatedEmployee.Permanent;

emp.Department = updatedEmployee.Department;

emp.Skills = updatedEmployee.Skills;

emp.DateOfBirth = updatedEmployee.DateOfBirth;

return Ok(emp);

}

private List<Employee> GetStandardEmployeeList()

{

return new List<Employee>

{

new Employee

{

Id = 1,

Name = "Ashutosh Dash",

Salary = 50000,

Permanent = true,

DateOfBirth = new DateTime(1999, 8, 15),

Department = new Department { Id = 101, Name = "IT" },

Skills = new List<Skill>

{

new Skill { Id = 1, Name = "C#" },

new Skill { Id = 2, Name = ".NET Core" }

}

},

new Employee

{

Id = 2,

Name = "Simran Sharma",

Salary = 60000,

Permanent = false,

DateOfBirth = new DateTime(1998, 11, 20),

Department = new Department { Id = 102, Name = "HR" },

Skills = new List<Skill>

{

new Skill { Id = 3, Name = "Excel" },

new Skill { Id = 4, Name = "Recruitment" }

}

}

};

}

}

}

**// Program.cs**

using CustomWebApi.Filters;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers(options =>

{

options.Filters.Add<CustomExceptionFilter>();

});

builder.Services.AddScoped<CustomAuthFilter>();

builder.Services.AddScoped<CustomExceptionFilter>();

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

var app = builder.Build();

app.UseSwagger();

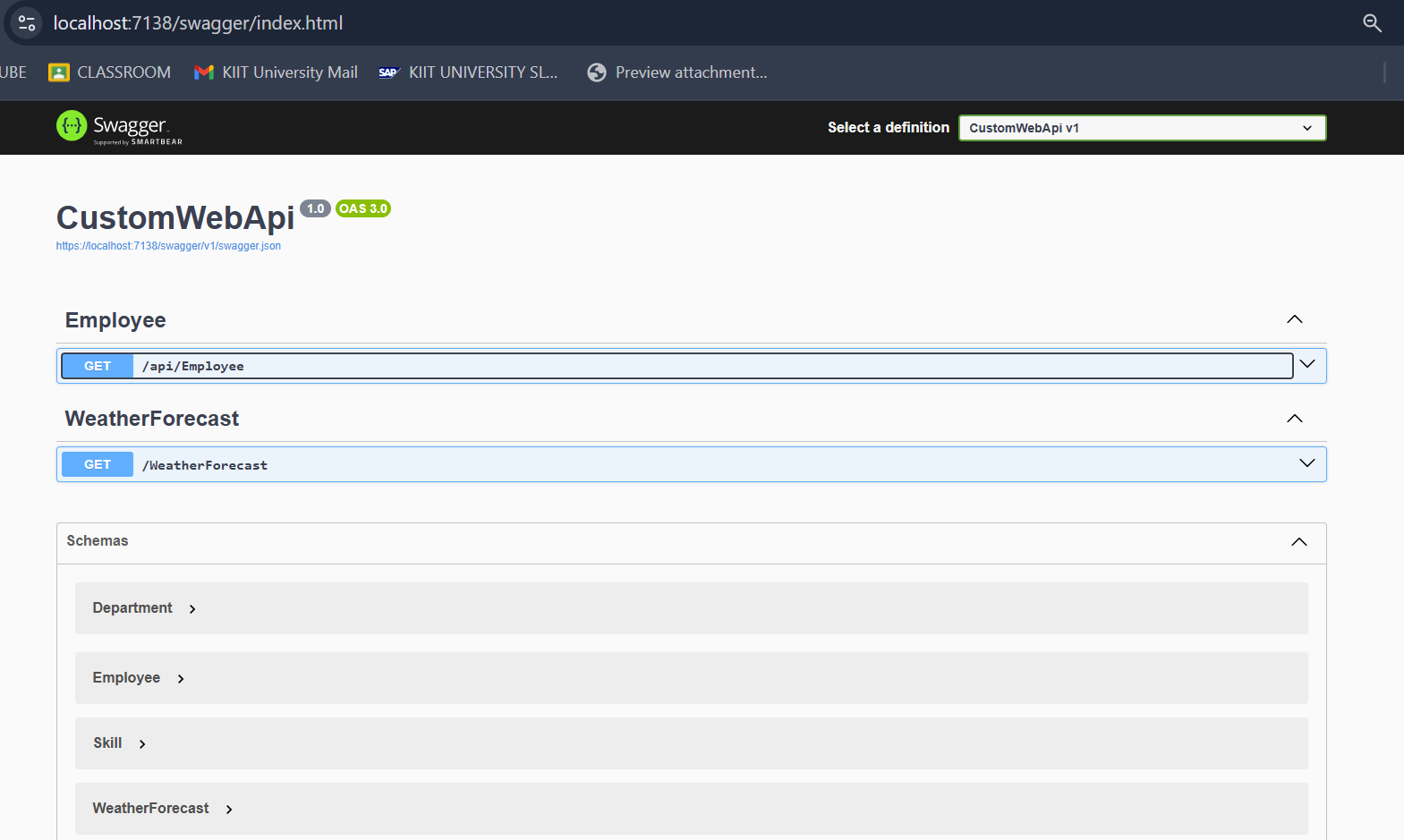
app.UseSwaggerUI();

app.UseAuthorization();

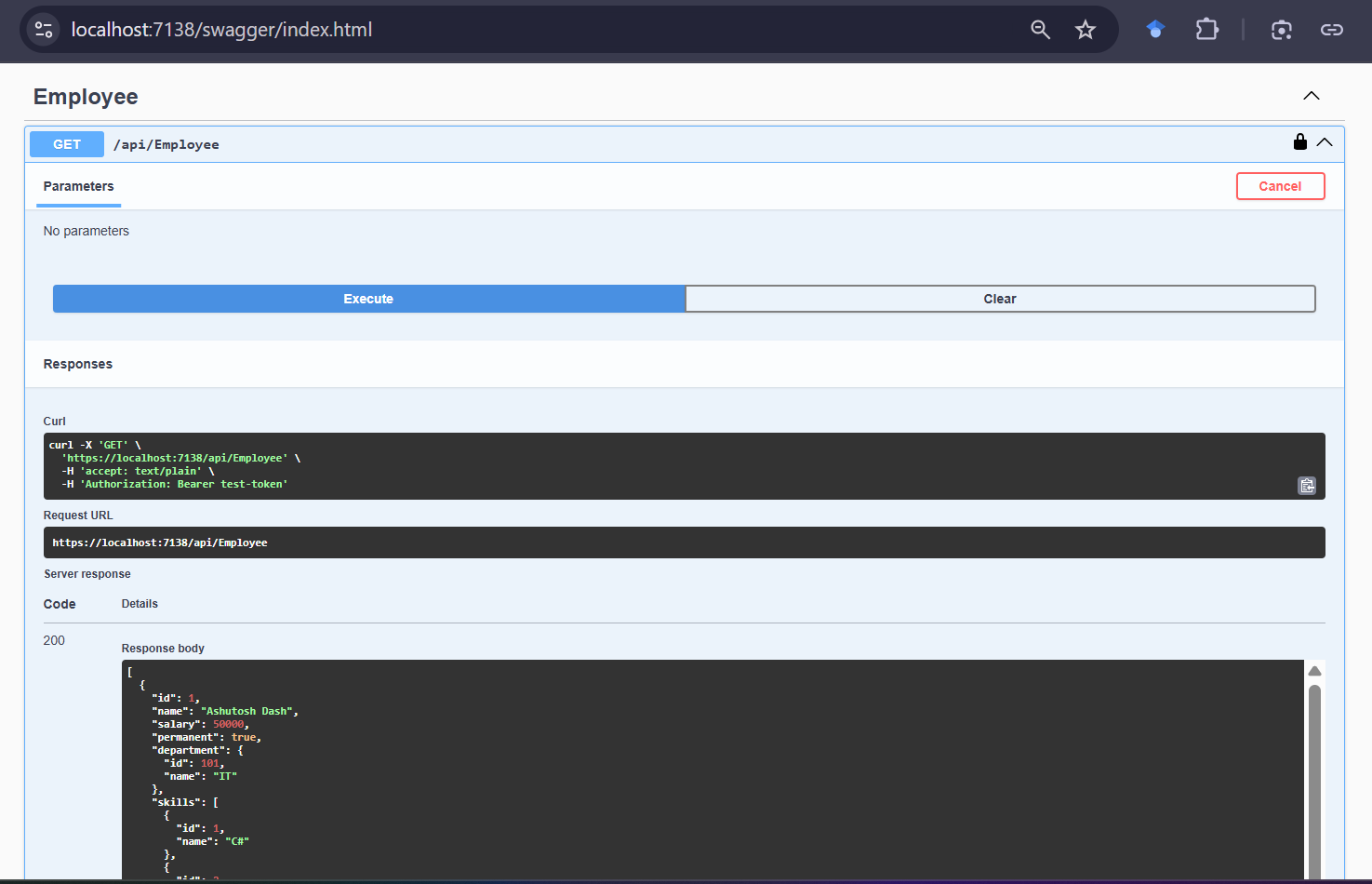
app.MapControllers();

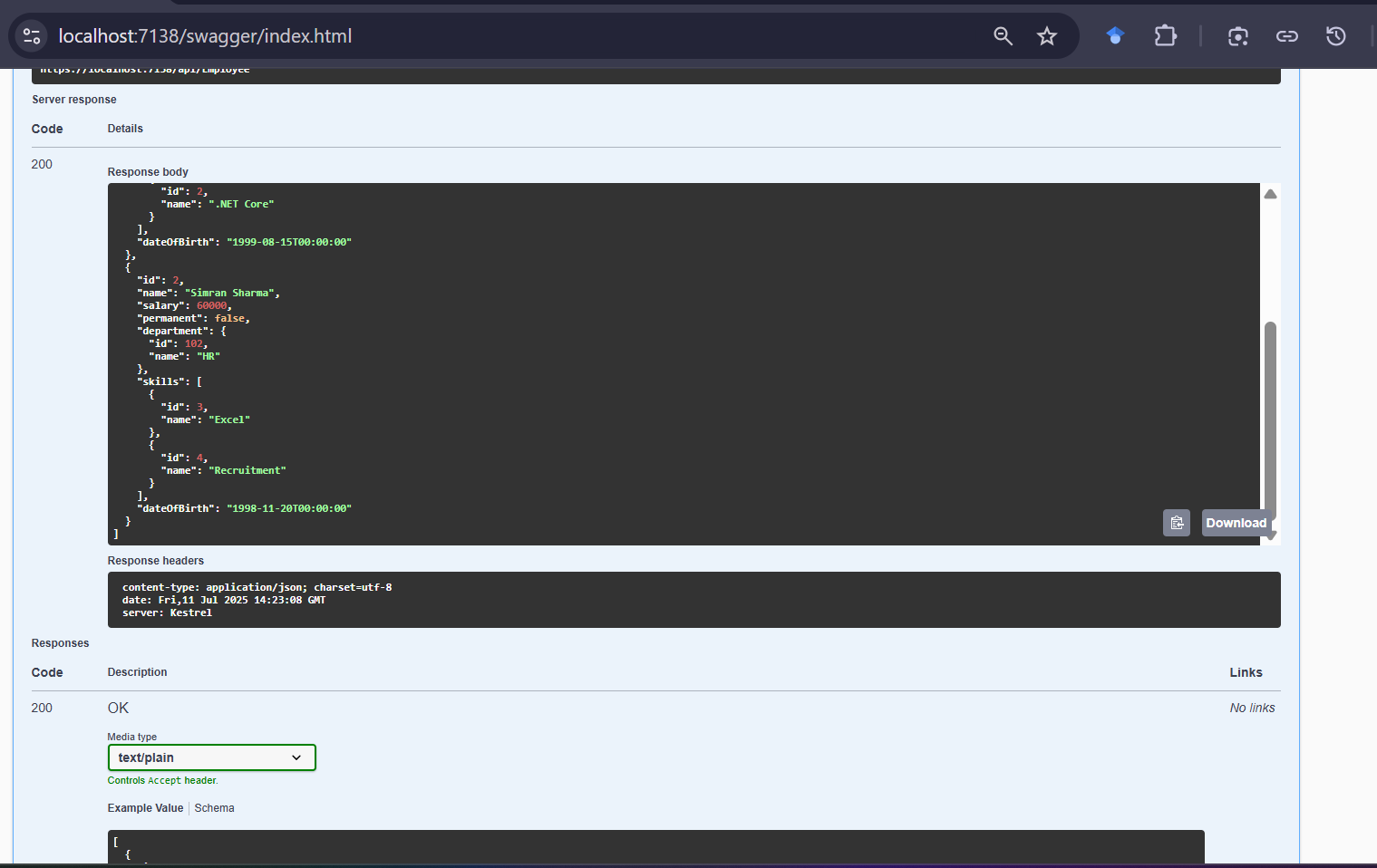
app.Run();

**1. Web Api using custom model class**

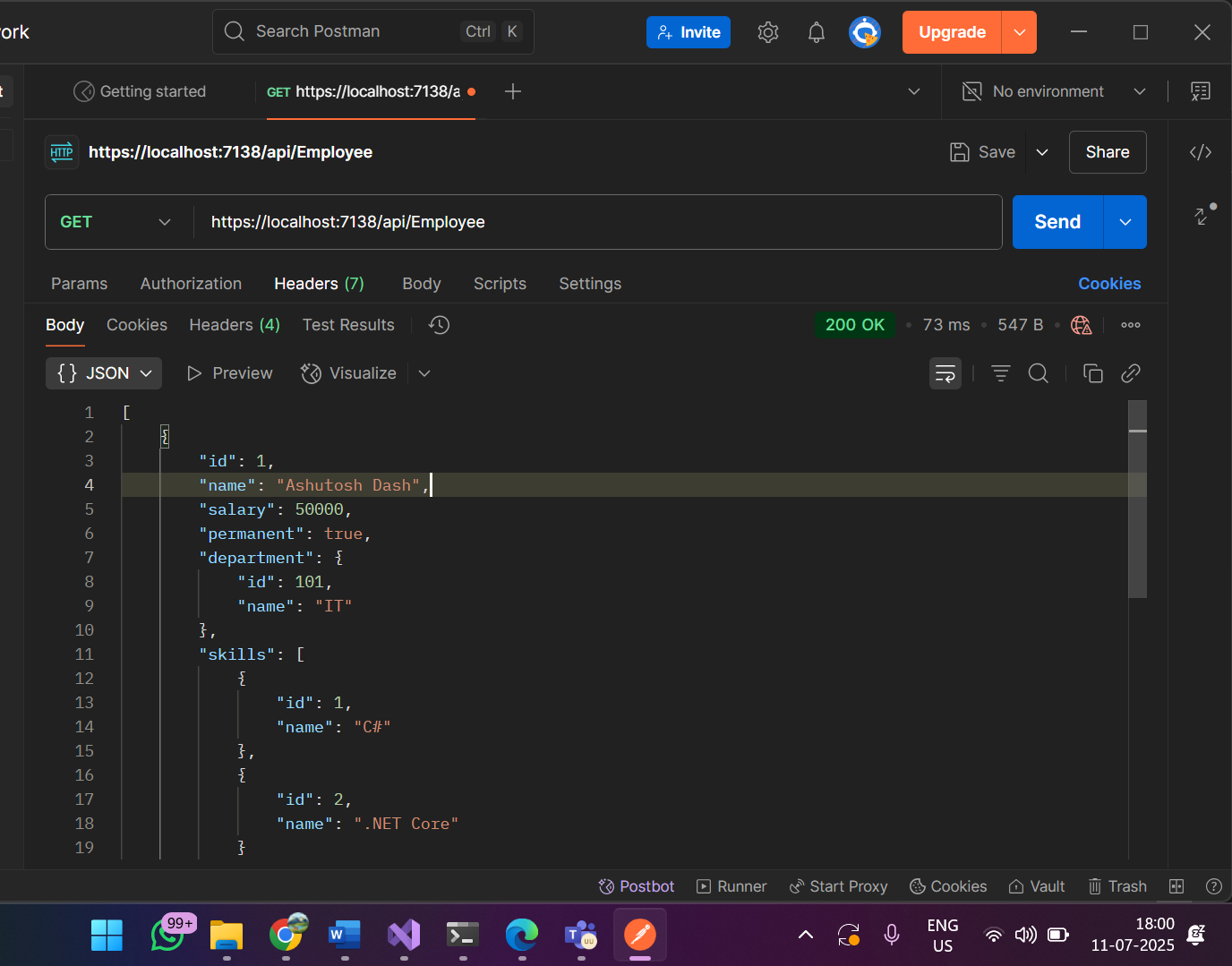


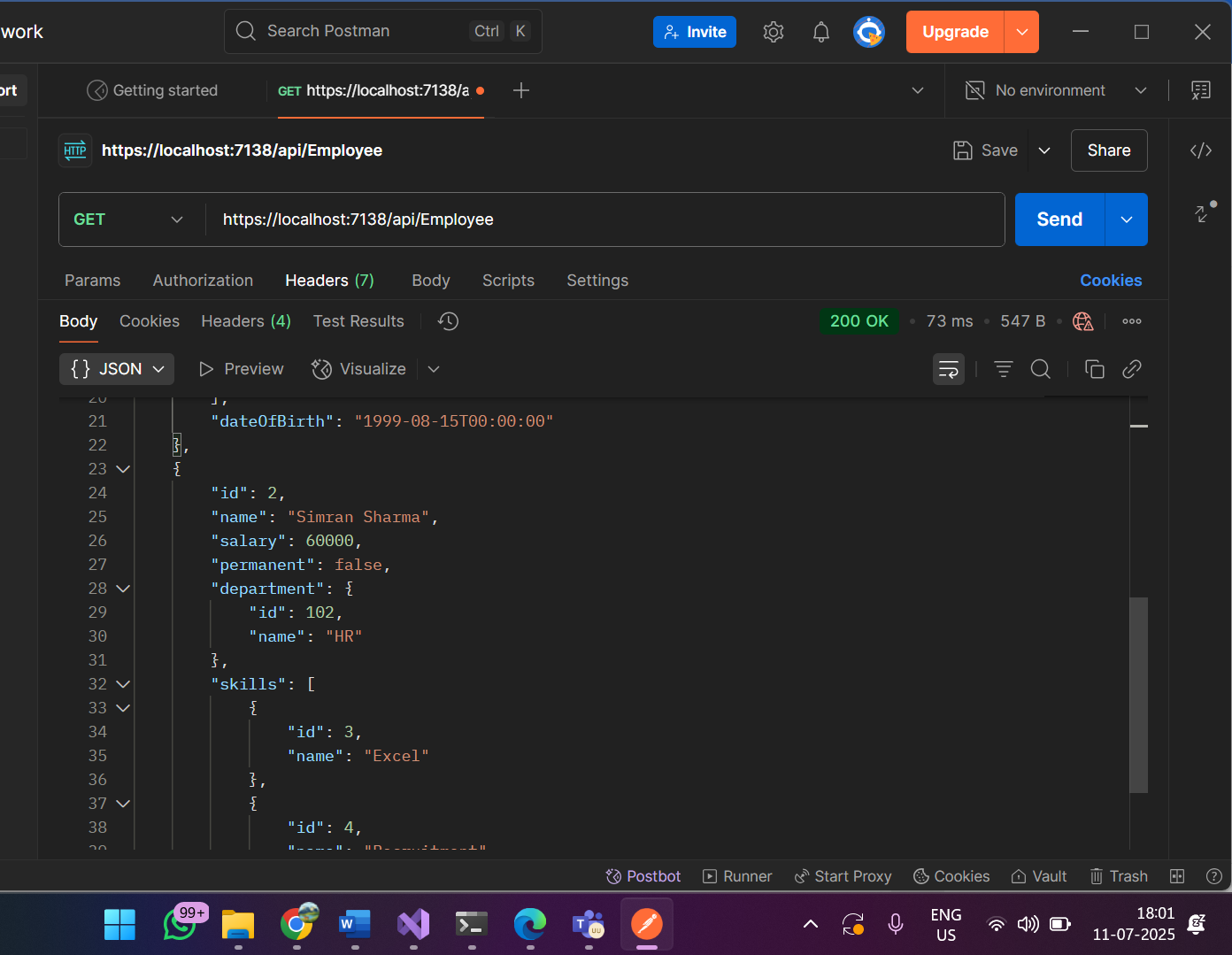
**2. Create a Custom action filter for Authorization.**

****

****

**Postman**

****

****

**3. Custom Exception filter**

public ActionResult<List<Employee>> GetStandard()

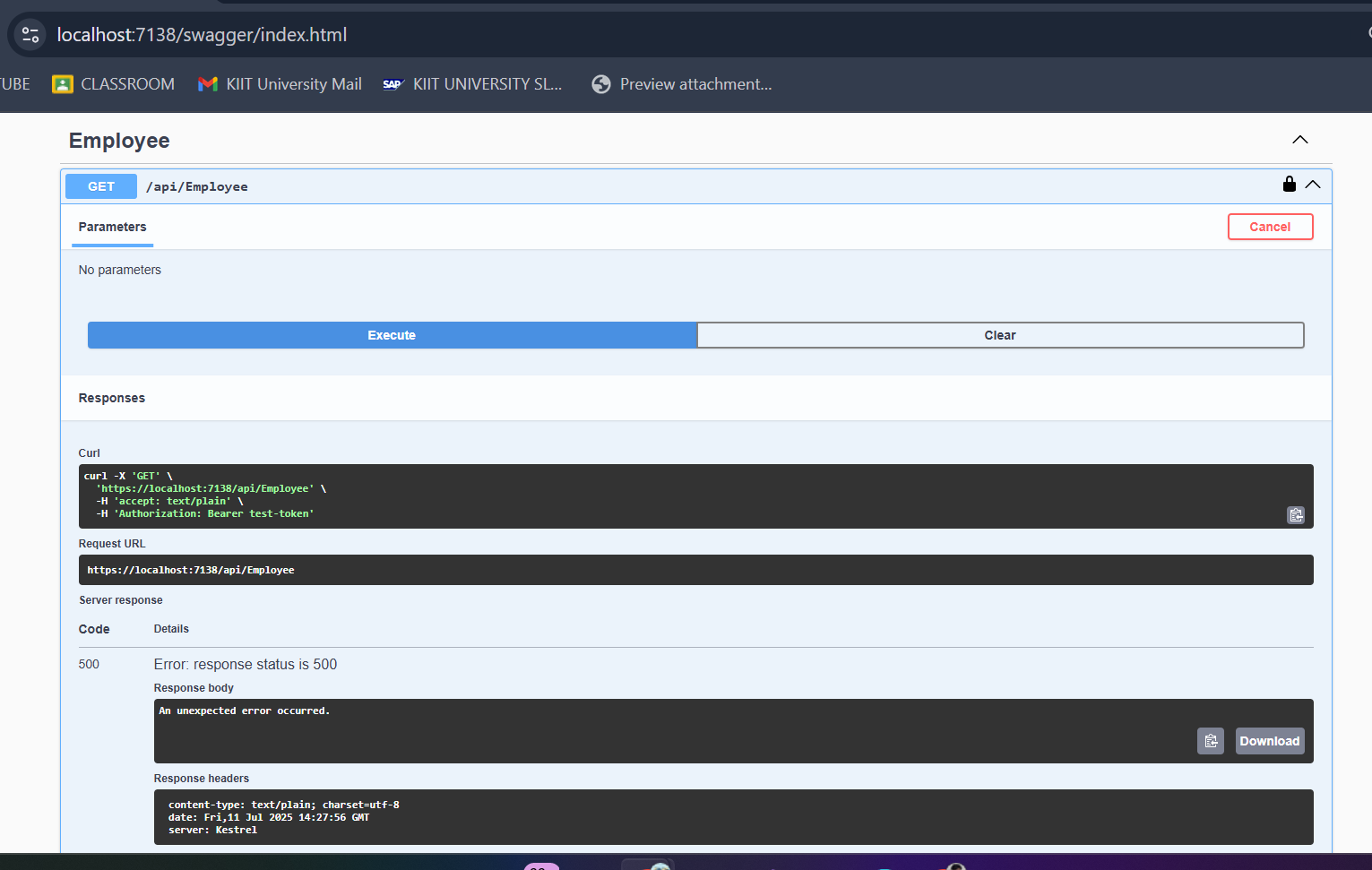
{

throw new Exception("Test Exception"); // Uncomment to test custom exception filter

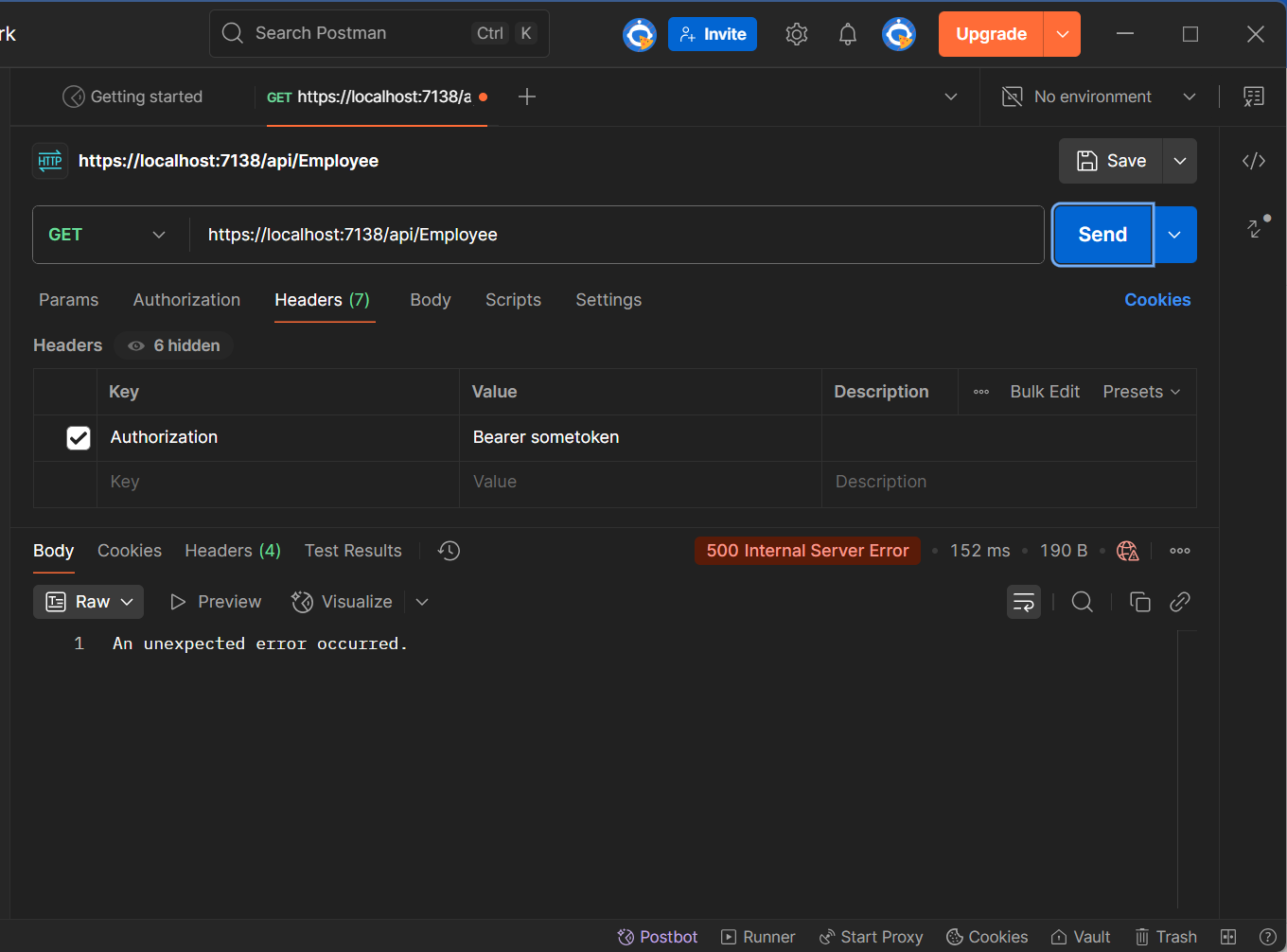
return Ok(\_employees);

}

**Swagger**

****

**Postman**

****