**Hands on WEBAPI-2**

**INTCDE21ID008**

**C N S Varun**

**916214**

**HANDSON-1**

Startup.cs

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using Microsoft.OpenApi.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Sam1

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllers();

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo

{

Title = "Swagger Demo",

Version = "v1",

Description = "TBD",

TermsOfService = new Uri("https://www.google.com"),

Contact = new OpenApiContact() { Name = "John Doe", Email = "john@xyzmail.com", Url = new Uri("https://www.example.com") },

License = new OpenApiLicense() { Name = "License Terms", Url = new Uri("https://www.example.com") }

});

});

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseSwagger();

app.UseSwaggerUI(c =>

{

// specifying the Swagger JSON endpoint.

c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");

});

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

}

}

}

SamController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Sam1.Controllers

{

[ApiController]

[Route("[controller]")]

public class WeatherForecastController : ControllerBase

{

[HttpGet]

public IEnumerable<string> Get()

{

return new string[] { "value1", "value2" };

}

// GET api/<ValuesController>/5

[HttpGet("{id}")]

public string Get(int id)

{

return "value";

}

// POST api/<ValuesController>

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT api/<ValuesController>/5

[HttpPut("{id}")]

public void Put(int id, [FromBody] string value)

{

}

// DELETE api/<ValuesController>/5

[HttpDelete("{id}")]

public void Delete(int id)

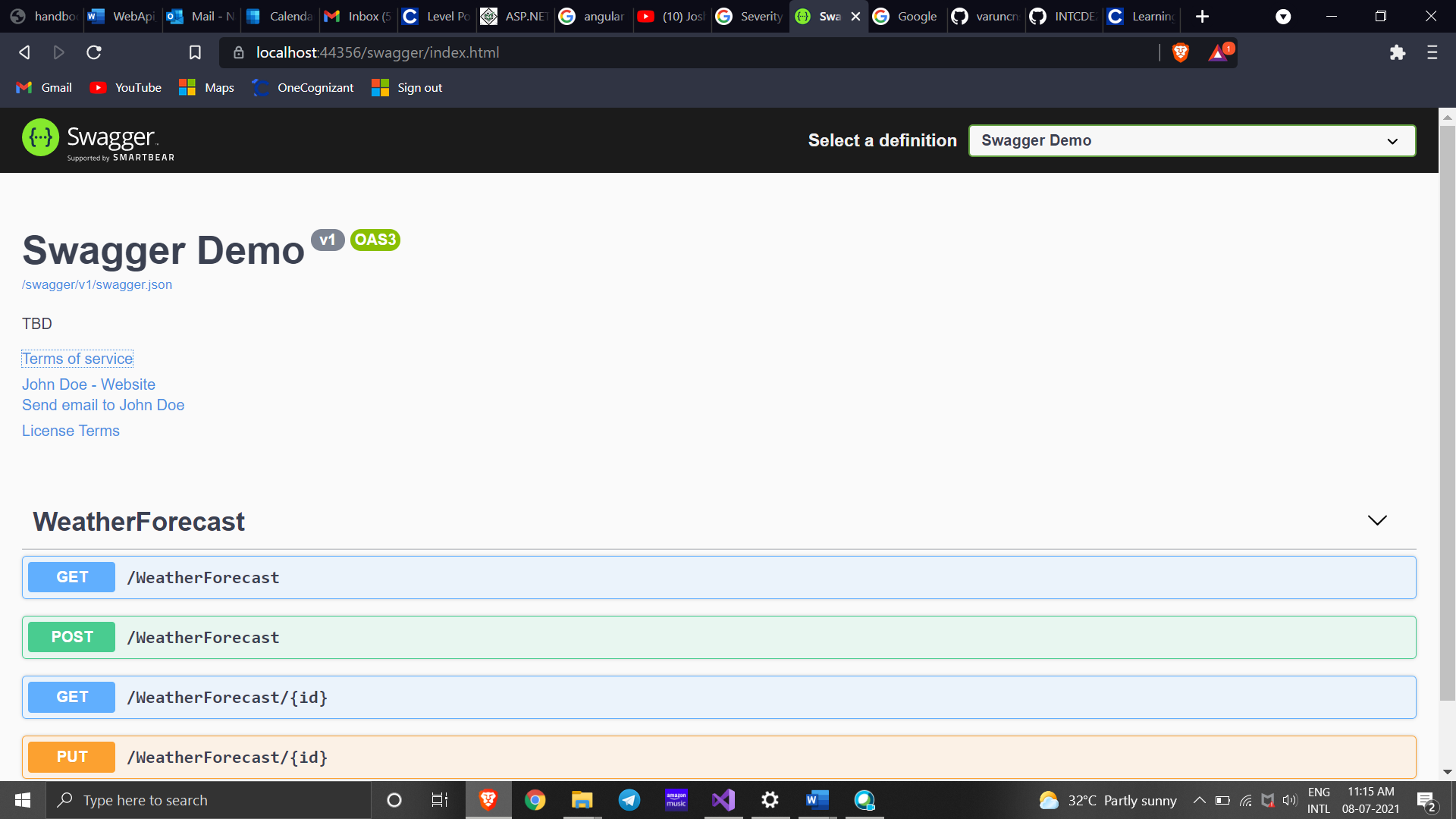
{

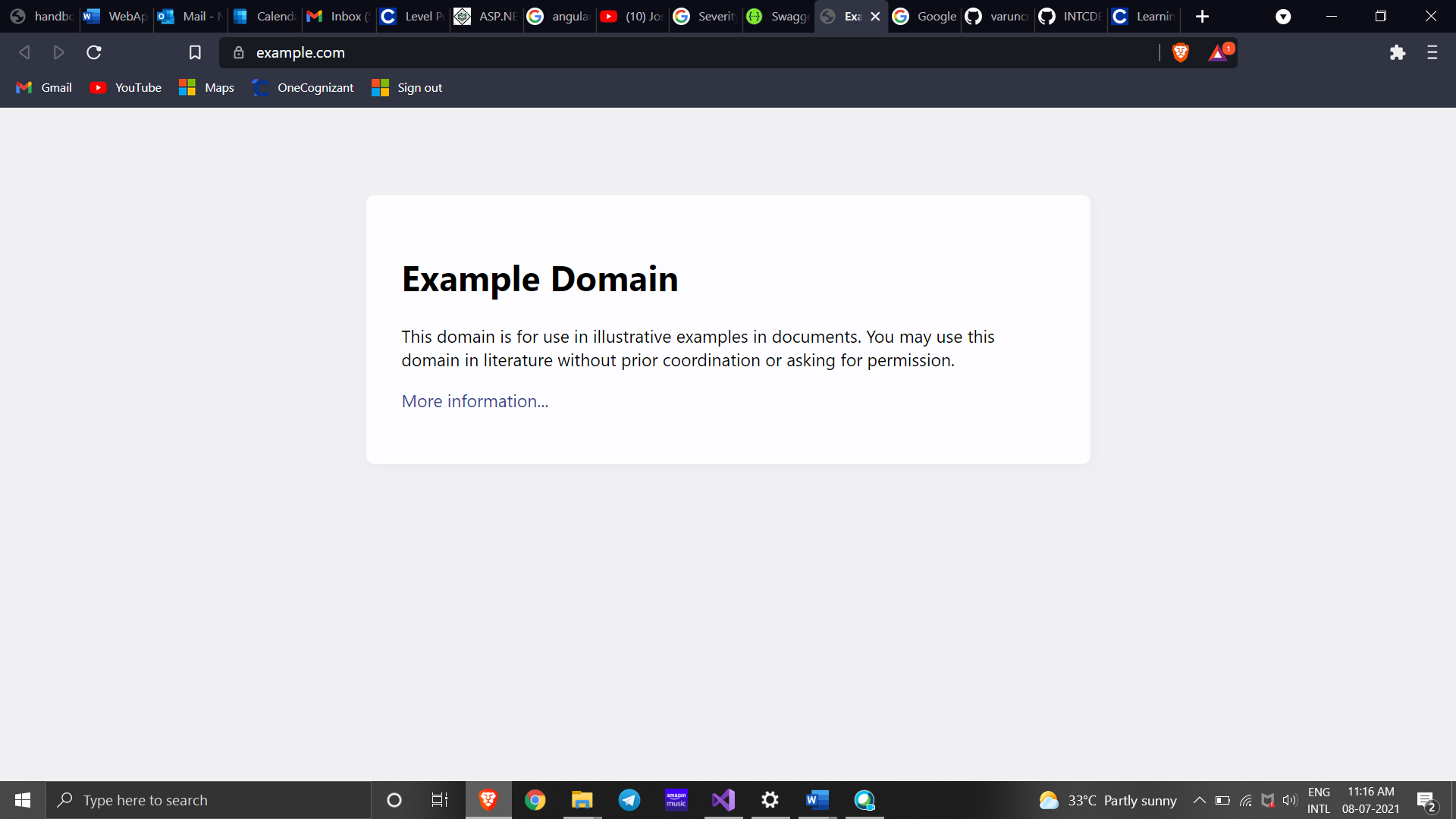
}

}

}

OUTPUT





**HANDSON-2**

CustomAuthFilter.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using Microsoft.Extensions.Primitives;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace Sam1.Filters

{

public class CustomAuthFilter : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext context)

{

if (context.HttpContext.Request.Query.ContainsKey("Authorization") && context.HttpContext.Request.Query["Authorization"] == "true")

{

context.Result = new UnauthorizedResult();

}

else

{

base.OnActionExecuting(context);

}

}

}

}

EmployeeController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http;

using System;

using System.Collections.Generic;

using System.Linq;

using Sam1.Filters;

namespace Sam1.Controllers

{

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

[ApiController]

public class EmployeeController : Controller

{

private Employee[] emp = new Employee[]

{

new Employee { Id=1 , Name="Eoin Morgan" , Salary=20000 , Permanent=true, Department="ENG",Skills="White" ,DateOfBirth=new DateTime(1998,08,07) },

new Employee { Id=2 , Name="Virat Kohli" , Salary=35000 , Permanent=true, Department="IND",Skills="Red" ,DateOfBirth=new DateTime(1999,11,09) } ,

new Employee { Id=3 , Name="Aakash" , Salary=500 , Permanent=false, Department="None",Skills="Commentary" ,DateOfBirth=new DateTime(1997,02,12) } ,

};

private IEnumerable<Employee> GetStandardEmployeeList()

{

return emp;

}

// GET: api/<ValuesController1>

[CustomAuthFilter]

[HttpGet]

public IEnumerable<Employee> Get()

{

return GetStandardEmployeeList();

}

// GET api/<ValuesController1>/5

[HttpGet("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status404NotFound)]

public IActionResult GetById(int id)

{

var prod = emp.FirstOrDefault((p) => p.Id == id);

if (prod == null)

{

return NotFound();

}

return Ok(prod);

}

// POST api/<ValuesController1>

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT api/<ValuesController1>/5

[HttpPut("{id}")]

public void Put(int id, [FromBody] string value)

{

}

// DELETE api/<ValuesController1>/5

[HttpDelete("{id}")]

public void Delete(int id)

{

}

}

}

Employee.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Sam1

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public string Department { get; set; }

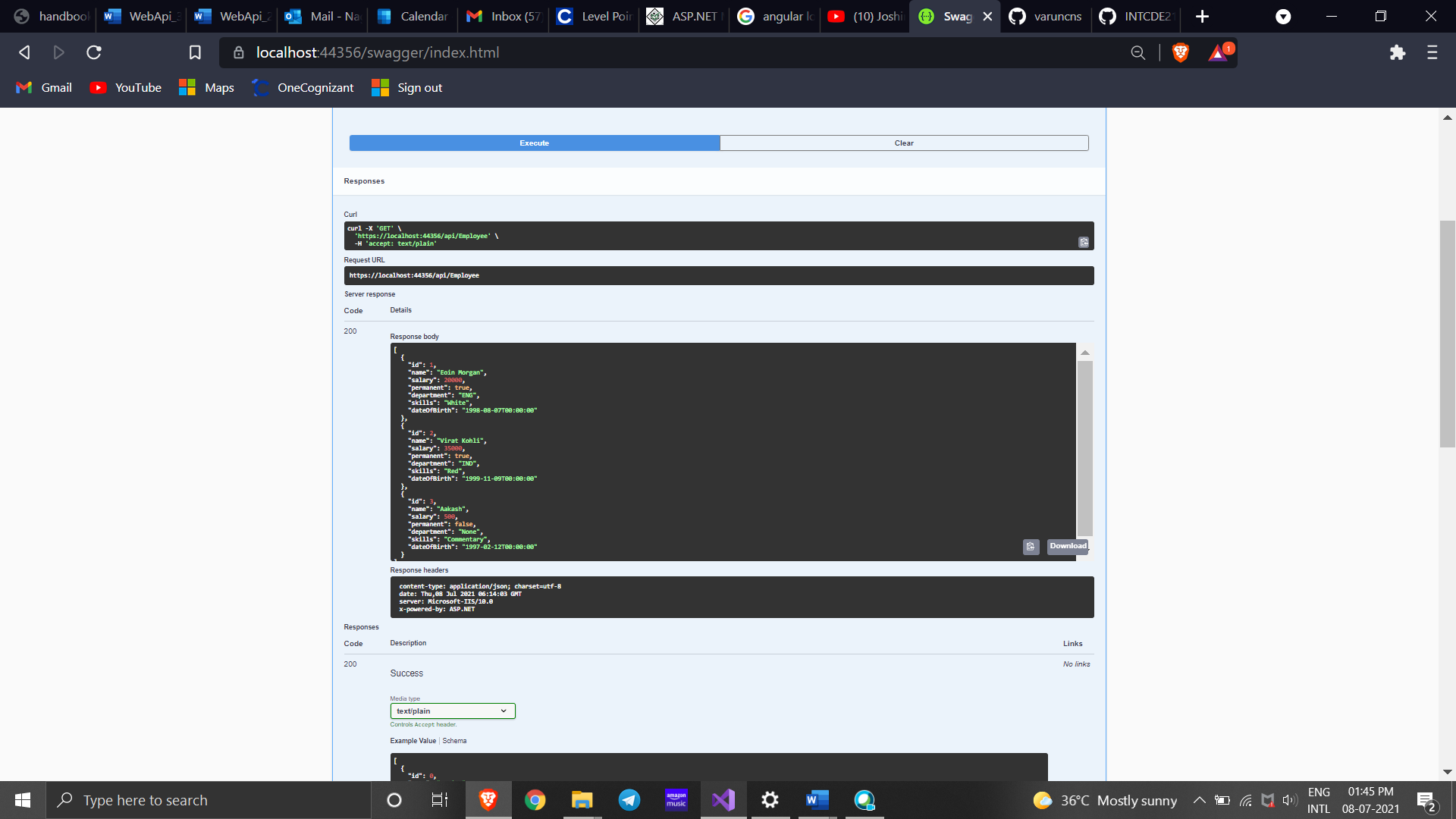
public string Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

OUTPUT



**HANDSON-3**

EmployeeController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http;

using System;

using System.Collections.Generic;

using System.Linq;

using Sam1.Filters;

namespace Sam1.Controllers

{

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

[ApiController]

public class EmployeeController : Controller

{

private static List<Employee> \_emp = new List<Employee>();

// GET: api/<ValuesController1>

[HttpGet(Name = "GetAllStudent")]

public IActionResult Get()

{

return new ObjectResult(\_emp);

}

// GET api/<ValuesController1>/5

[HttpGet("{id}", Name = "GetStudent")]

public IActionResult Get(int id)

{

return new ObjectResult(\_emp.FirstOrDefault(p => p.Id == id));

}

// POST api/<ValuesController1>

[HttpPost(Name = "CreateStudent")]

public IActionResult Post([FromBody] Employee emps)

{

\_emp.Add(emps);

return CreatedAtRoute("GetStudent", new { id = emps.Id }, emps);

}

// PUT api/<ValuesController1>/5

[HttpPut("{id}", Name = "UpdateStudent")]

public IActionResult Put(int id, [FromBody] Employee emps)

{

\_emp.FirstOrDefault(p => p.Id == id).Name = emps.Name;

return CreatedAtRoute("GetStudent", new { id = emps.Id }, emps);

}

// DELETE api/<ValuesController1>/5

[HttpDelete("{id}", Name = "DeleteStudent")]

public IActionResult Delete(int id)

{

var \_emps = \_emp.FirstOrDefault(p => p.Id == id);

\_emp.Remove(\_emps);

return new NoContentResult();

}

}

}

Employee.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Sam1

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

//public int Salary { get; set; }

//public bool Permanent { get; set; }

//public string Department { get; set; }

//public string Skills { get; set; }

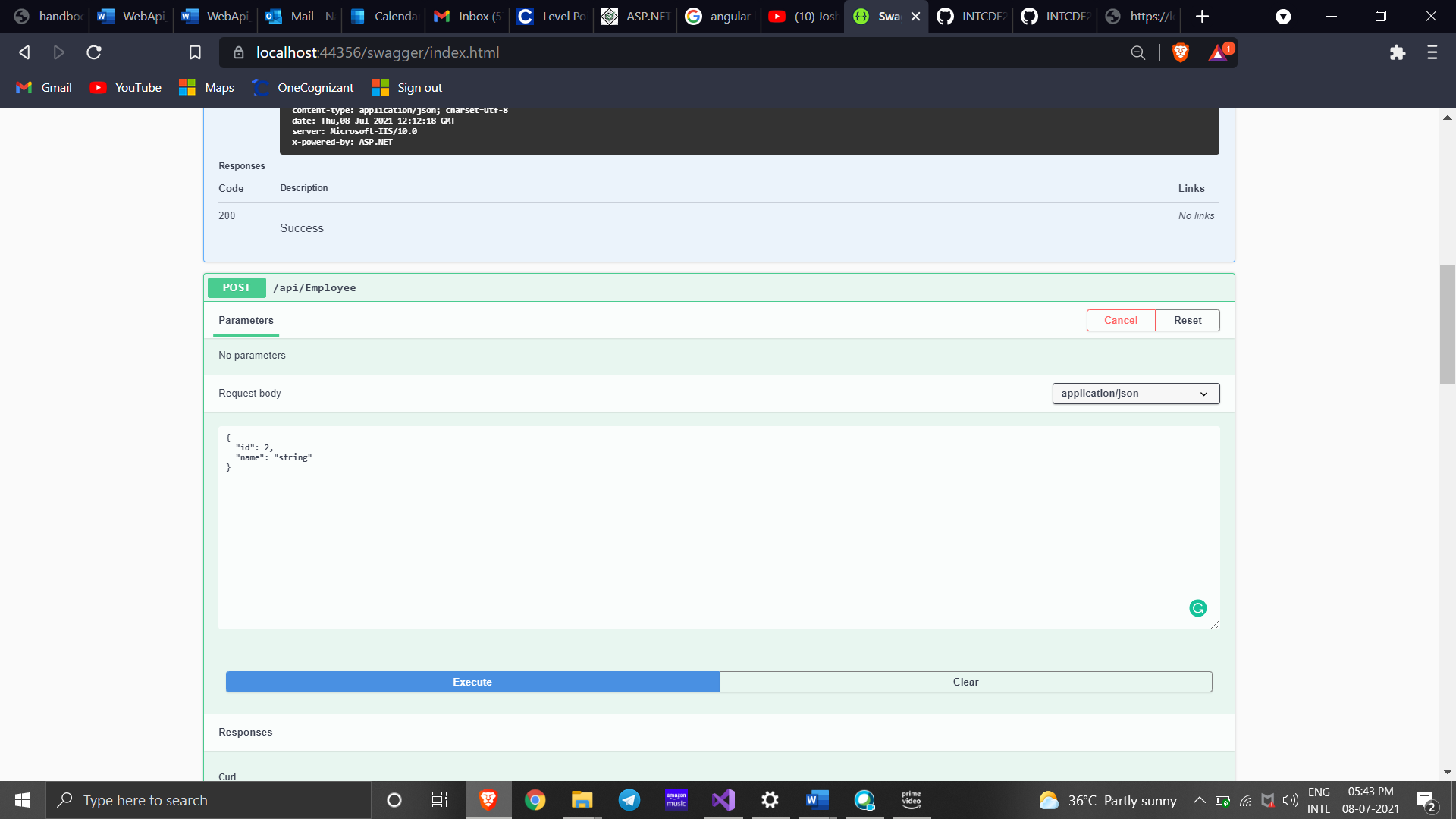
//public DateTime DateOfBirth { get; set; }

}

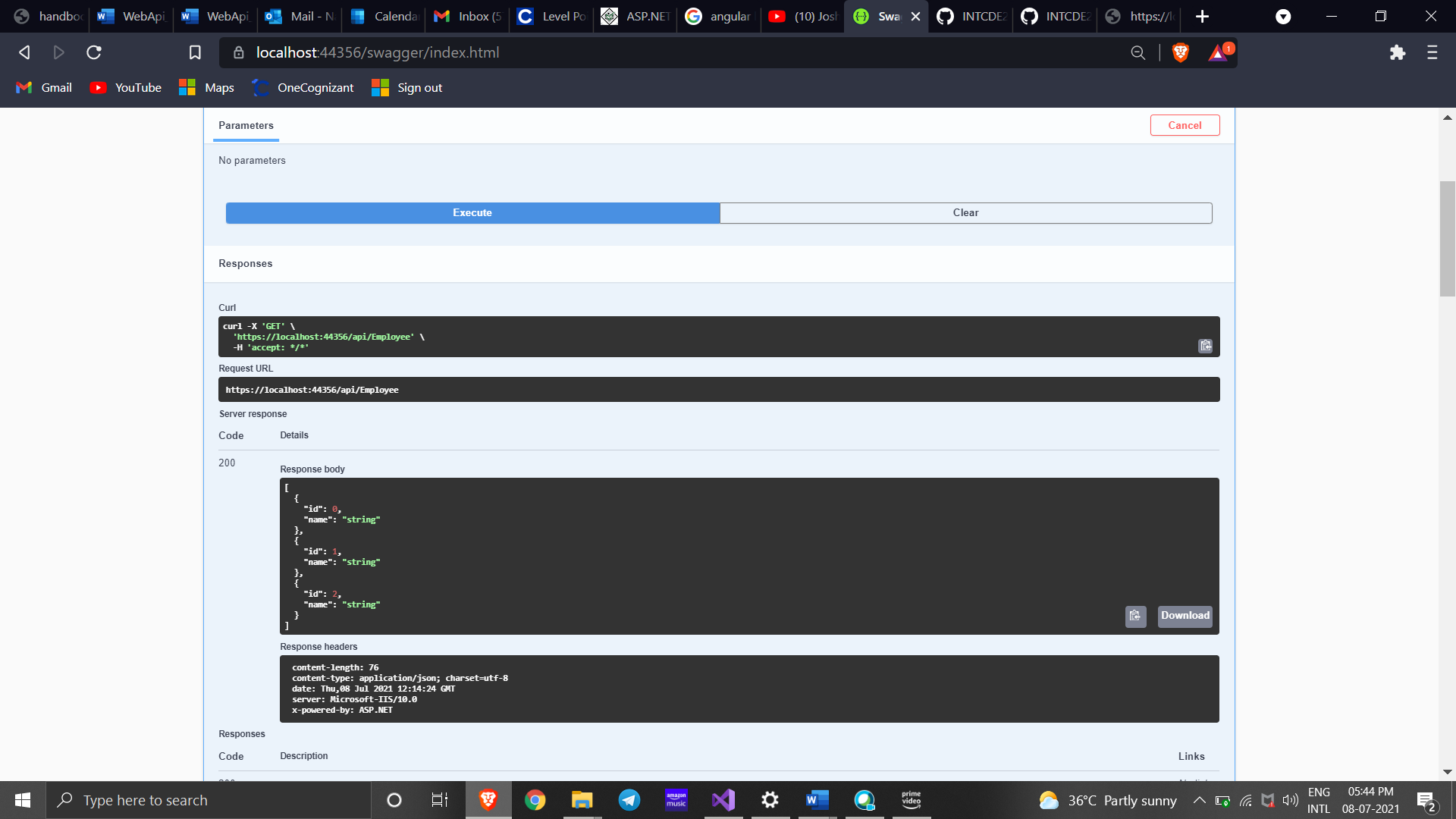
}

OUTPUT

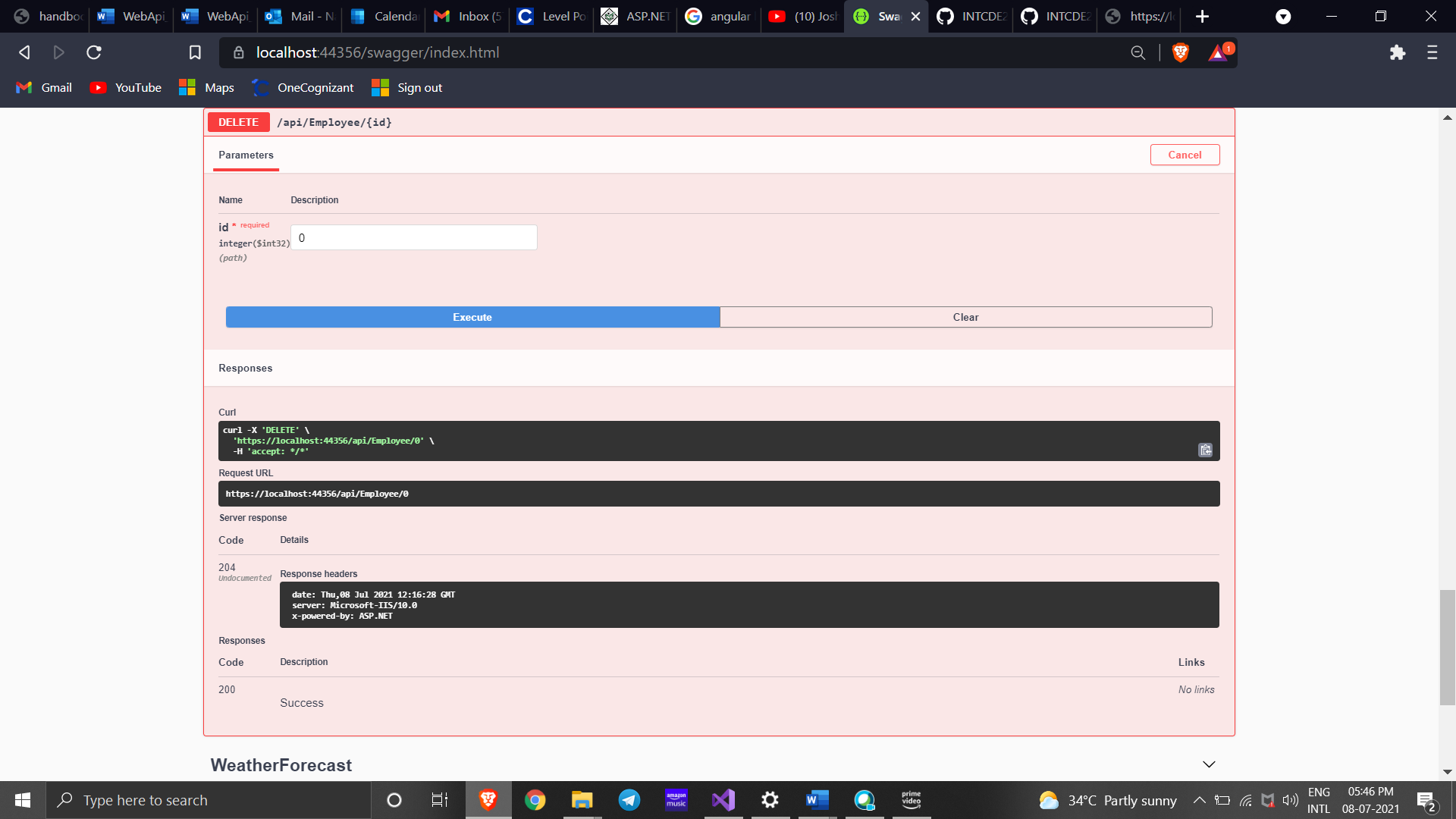
POST(CREATE)



GET METHOD



DELETE METHOD



GET METHOD After Deleting

