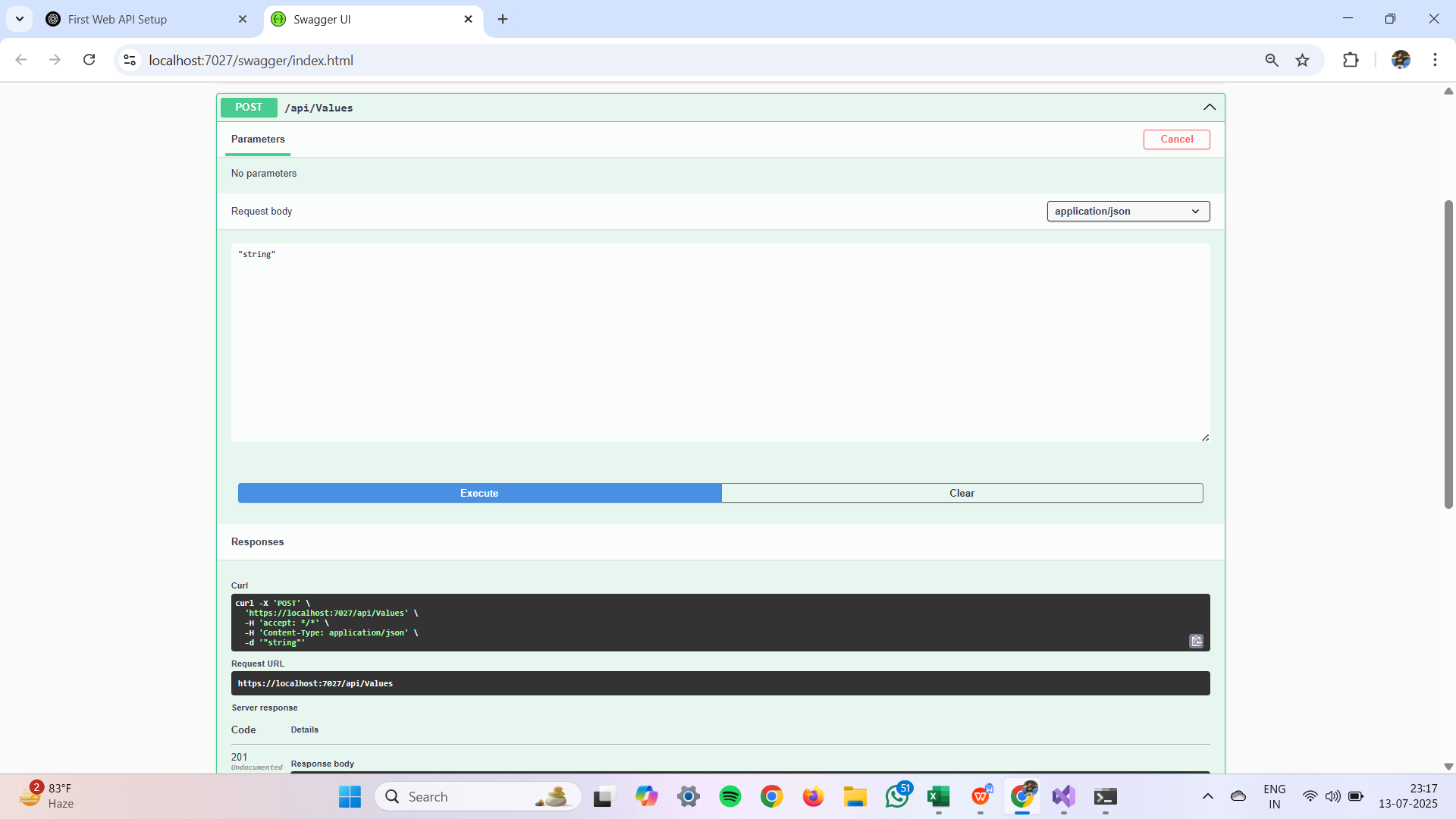
ASP.NET Core 8.0 Web API

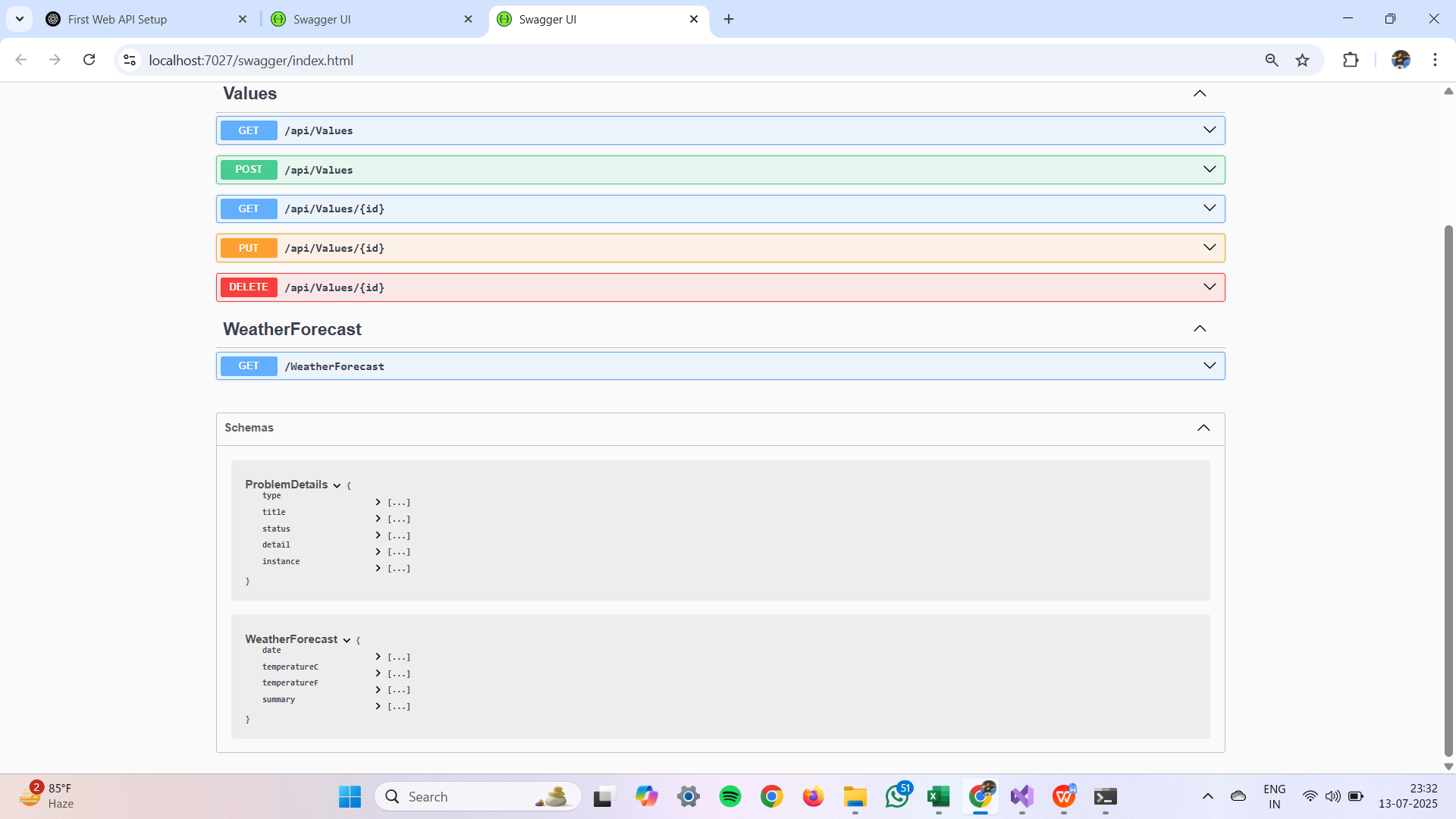
# 1. First Web API using .NET Core

* Introduced RESTful Web API and HTTP action verbs (GET, POST, PUT, DELETE).
* Created a basic Web API project using .NET Core with a default controller.
* Tested the GET method via browser or Swagger to verify API is working.

Explained configuration files like Startup.cs and appSettings.json.  
  
  


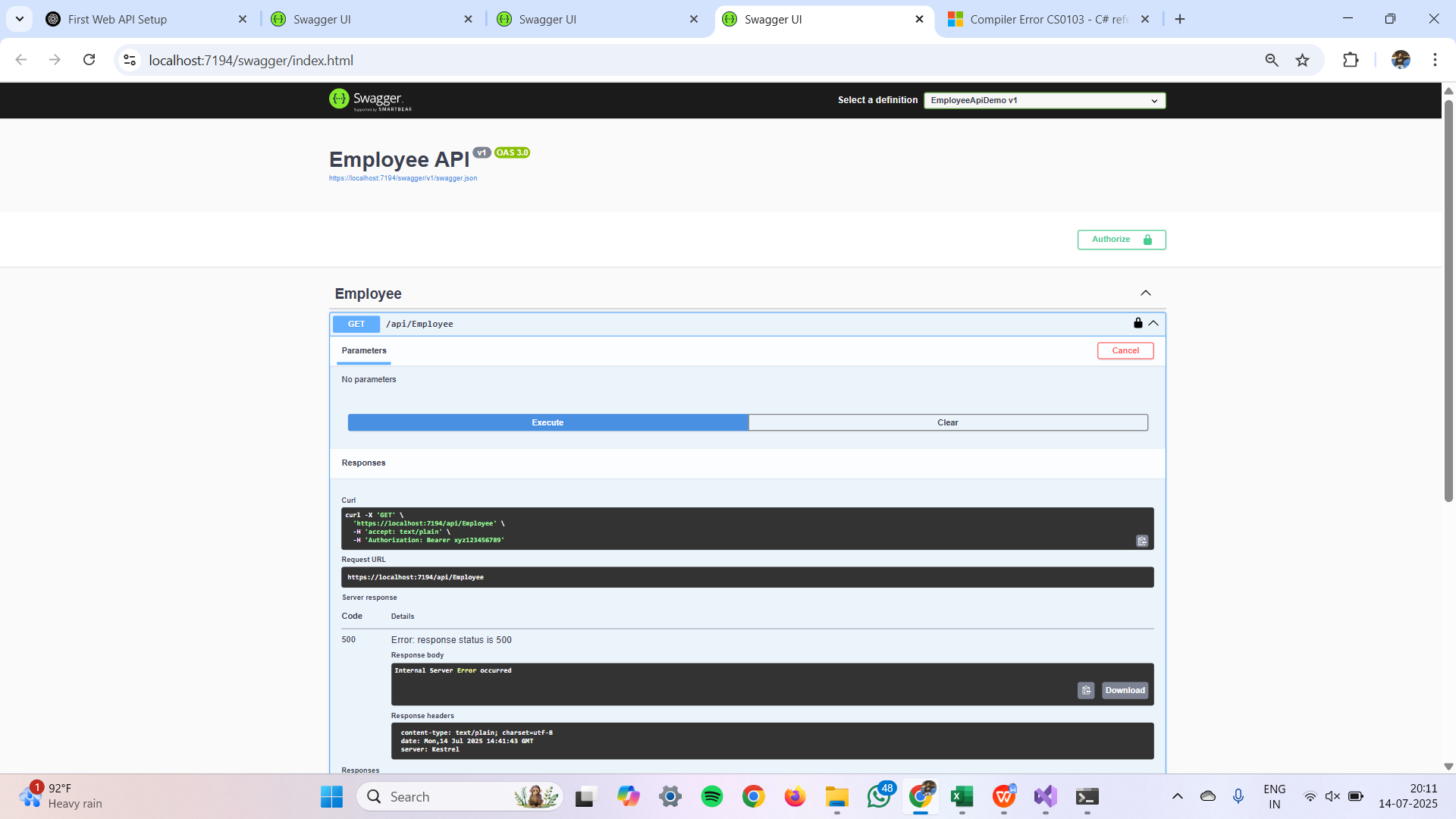
# 2. Web API with Swagger and Postman

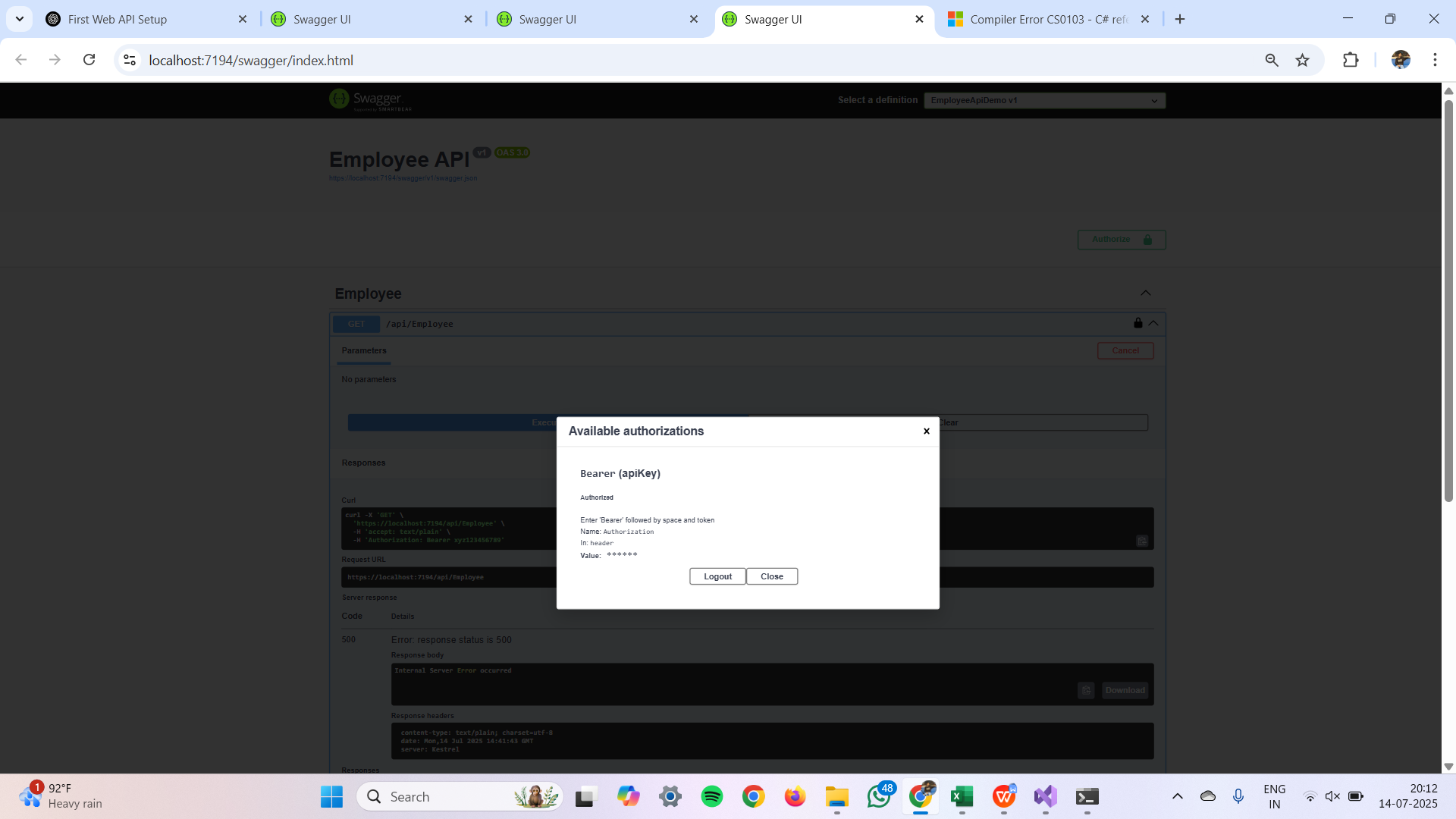
* Installed Swashbuckle to enable Swagger UI for API testing.
* Configured Swagger in Startup.cs using AddSwaggerGen and UseSwagger.
* Tested endpoints using Postman with headers and JSON body.

Demonstrated routing and named actions for better API structure.  
  
  
  
  
  


# 3. Web API using Custom Model and Filters

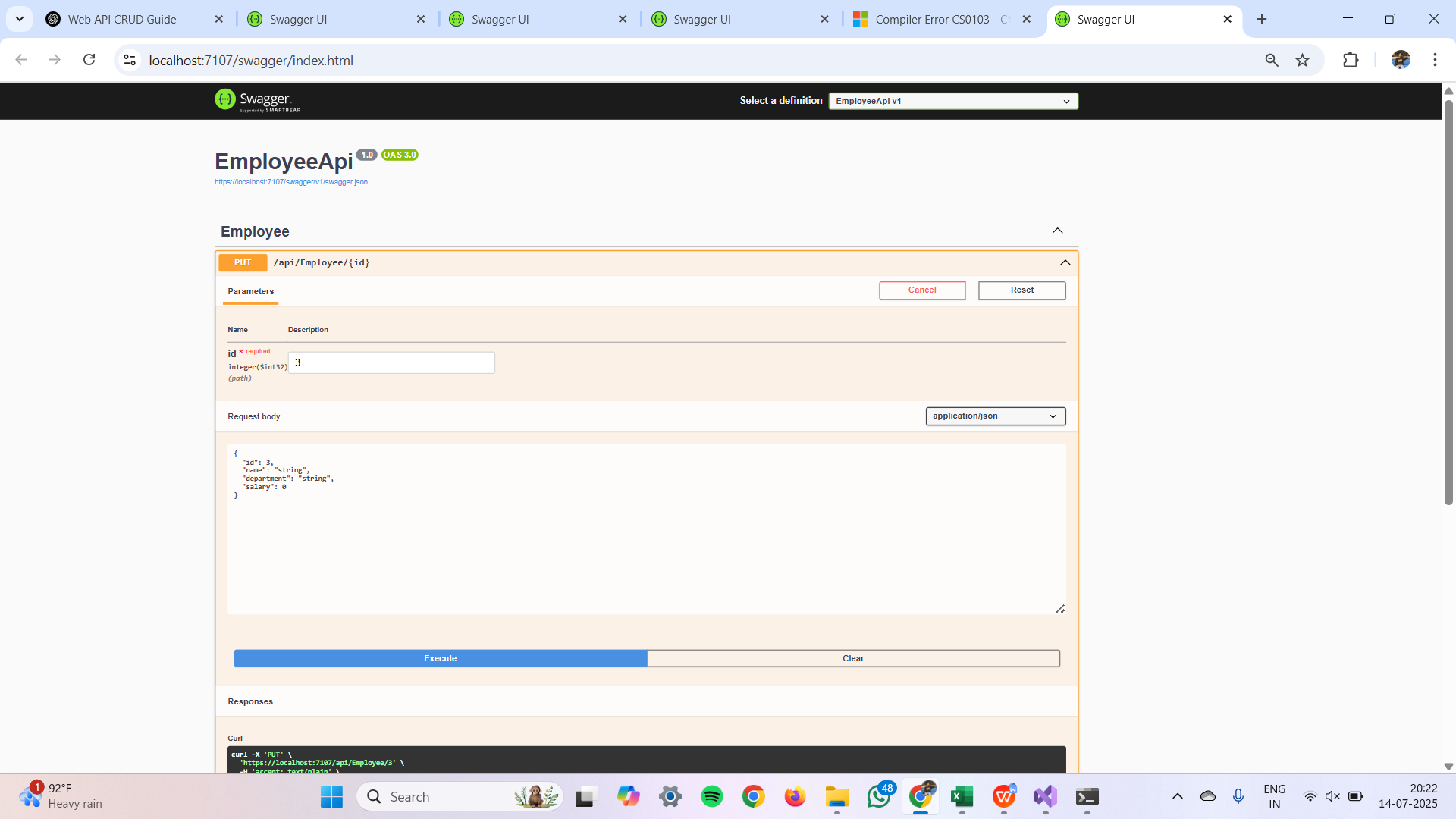
* Created a custom model class 'Employee' with complex nested data.
* Used [FromBody] to read data from HTTP request body into models.
* Created a custom action filter to validate Authorization headers.

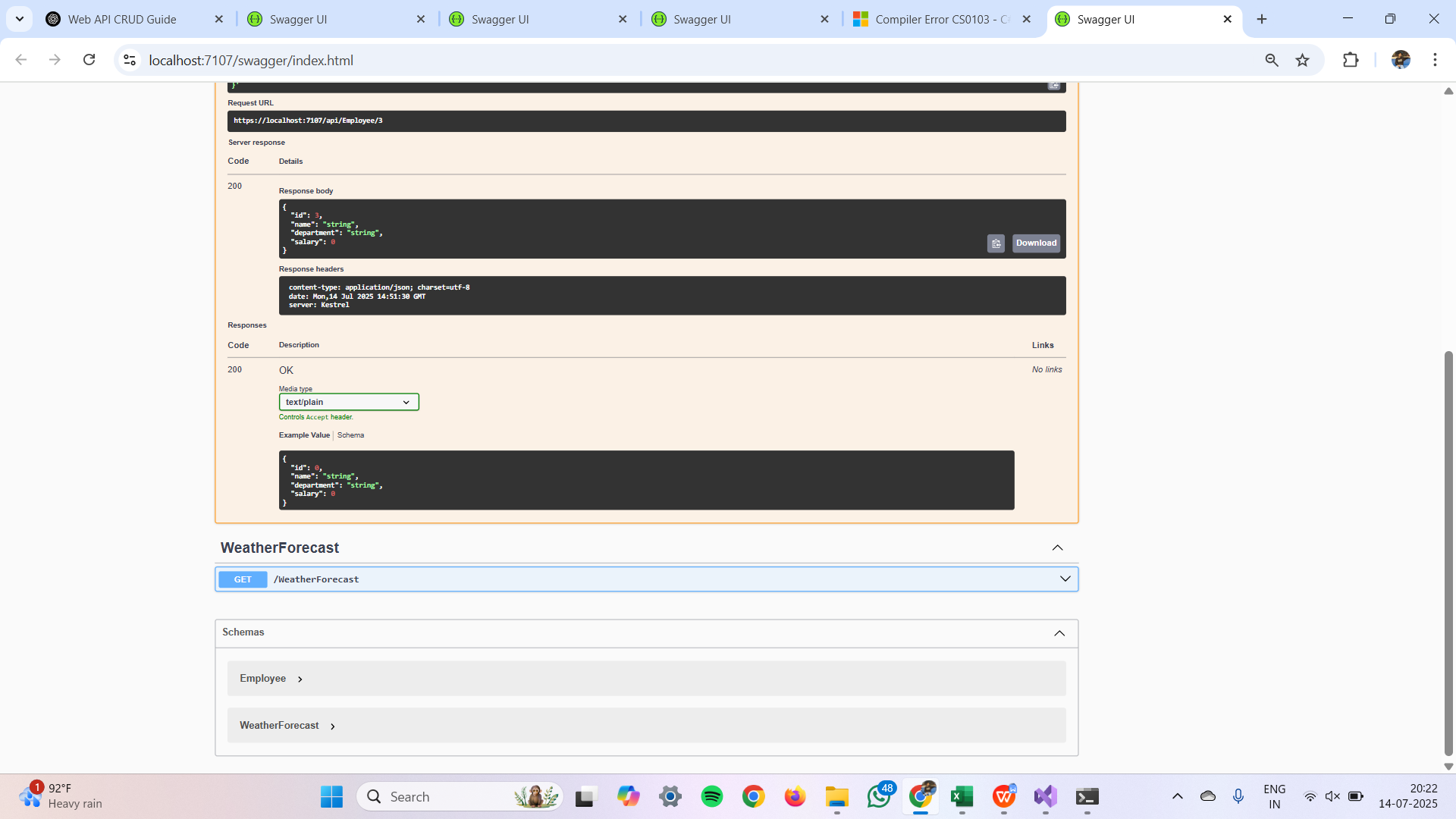
Handled exceptions using a custom exception filter to log and return messages.  
  
  




# 4. Web API CRUD Operations

* Implemented PUT method to update employee data using input JSON.
* Validated ID for correctness and existence before updating data.
* Used hardcoded list to simulate a database update.

Tested the method using Swagger and Postman tools.  
  




# 5. JWT Authentication and CORS in Web API

* Enabled JWT authentication using a security key in Startup.cs.
* Generated token using AuthController and tested with Postman.
* Used [Authorize] attribute to protect API endpoints by roles.  
    
  Handled token expiration and validated roles in token claims.  
    
  