

Web API Labs Summary

Lab 1

- Create a .NET Core Web API using the API template.
- Use the default controller (ValuesController) with GET, POST, PUT, DELETE methods.
- Run the application and validate GET method in Swagger or browser.

The image displays three screenshots of a web browser interface, likely Swagger UI, showing API responses and request details.

Top Screenshot: Shows the "Responses" tab for a POST request to `/api/Values`. The request body is `"hello"`. The response is a 201 status code with a response body of `received: hello world`. The response headers include `content-type: text/plain; charset=utf-8`, `date: Sun, 13 Jul 2025 16:33:54 GMT`, `location:`, and `server: Restra`.

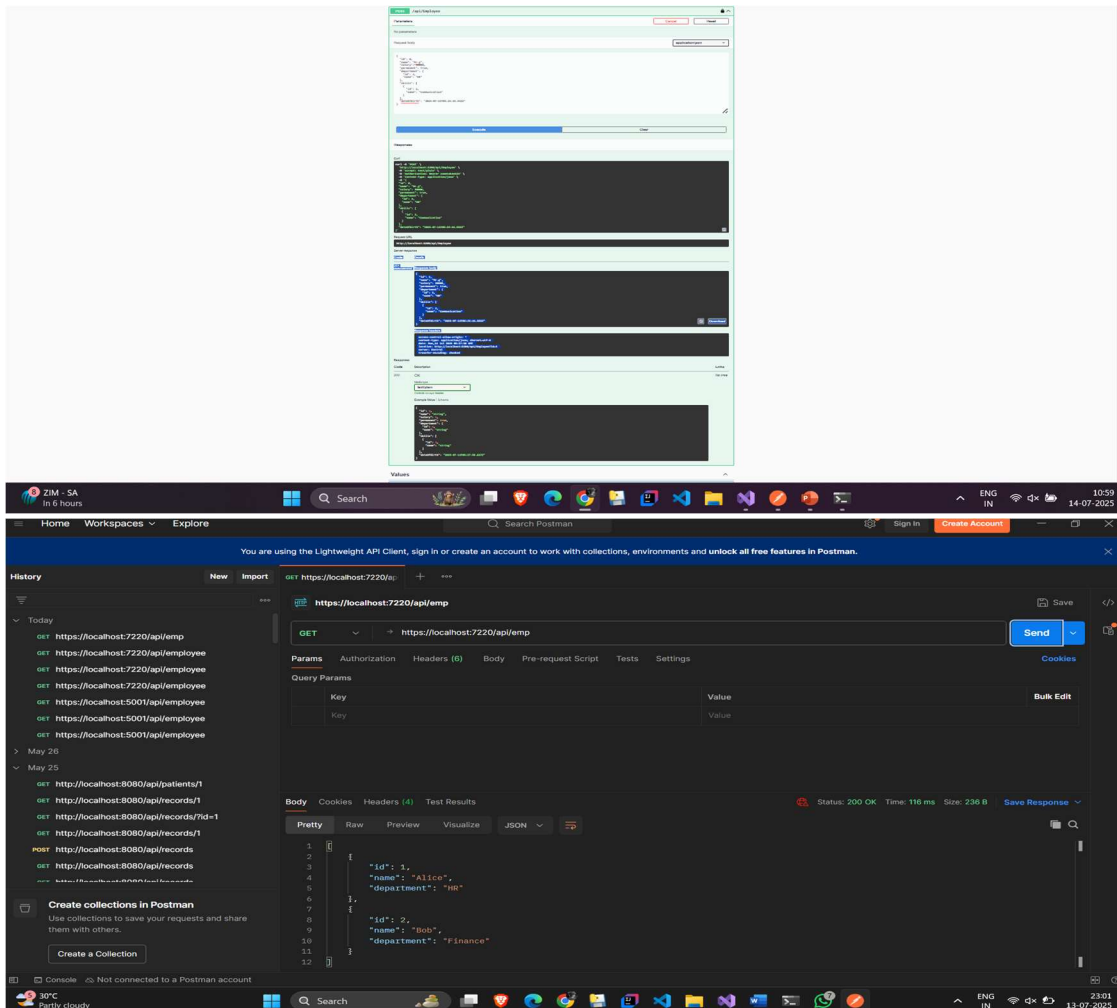
Middle Screenshot: Shows the "POST" tab for `/api/Values`. The request body is `"hello"`. The response is a 201 status code with a response body of `received: hello world`. The response headers include `content-type: text/plain; charset=utf-8`, `date: Sun, 13 Jul 2025 16:33:54 GMT`, `location:`, and `server: Restra`.

Bottom Screenshot: Shows the "Values" tab for a GET request to `/api/Values`. The response is a 200 status code with a response body of `{ "values": ["hello"] }`. The response headers include `content-type: application/json; charset=utf-8`, `date: Sun, 13 Jul 2025 16:37:28 GMT`, and `server: Restra`.

Lab 2

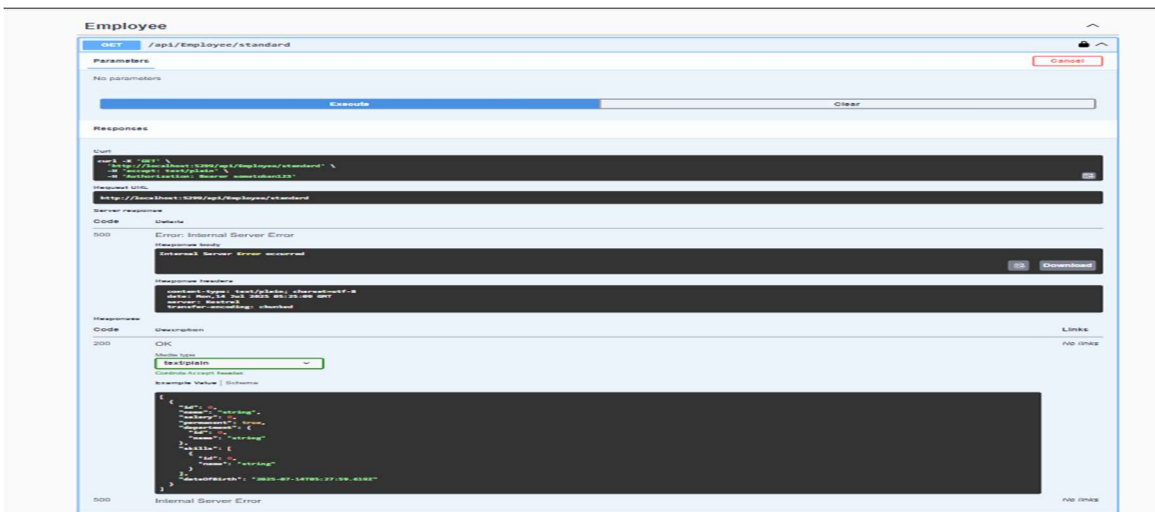
- Install Swashbuckle.AspNetCore NuGet package to enable Swagger.
- Configure Swagger in Program.cs to generate API documentation.

- • Test GET method through Swagger UI by clicking 'Try it out' and 'Execute'.
- • Verify output in Swagger UI and POSTMAN.
- • Rename route in Employee controller to 'Emp' and test.



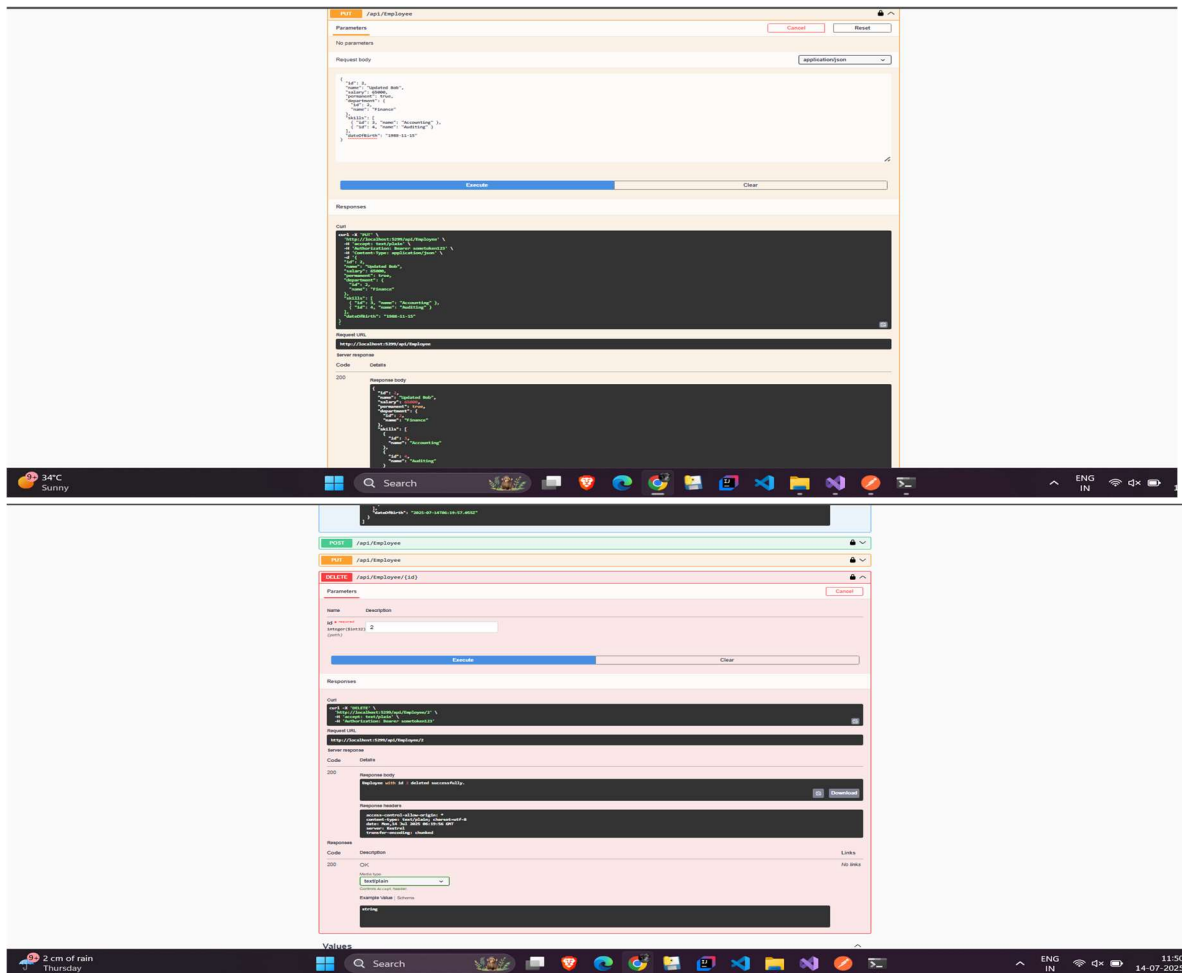
Lab 3

- • Create custom 'Employee' model with nested 'Department' and 'Skill' types.
- • Implement GET and POST methods in EmployeeController.
- • Add CustomAuthFilter to validate Authorization token in request header.
- • Add CustomExceptionHandler to handle and log exceptions globally.



Lab 4

- Implement PUT method in EmployeeController to update employee details.
- Validate employee ID: return BadRequest if invalid or not found.
- Update employee details from JSON input and return updated object.
- Test via Swagger and verify proper error/status responses.



Lab 5

- Enable JWT Authentication in Program.cs with secret key.
- Create AuthController with GET method to generate JWT token.
- Add [Authorize] attribute to EmployeeController to protect routes.
- Test with Swagger or POSTMAN using Authorization: Bearer <token>.
- Check for 401 Unauthorized on invalid/expired tokens.
- Add role-based authorization with [Authorize(Roles="Admin,POC")] and test.

[illegible]

Swagger UI

localhost:5299/index.html

Swagger

Select a definition

Swagger Demo

Swagger Demo

swagger.io/swagger.com

2017 Swagger 100k API

Join Docs - 100k+ APIs

Join GitHub - 100k+ APIs

License Terms

Authenticate

Auth

POST /api/auth/token

Parameters

No parameters

Execute

Clear

Responses

200

Response body

Response headers

Response cookies

Code

Details

200

OK

No error

Employee

2 cm of rain Thursday

Search

ENG IN

20:57 14-07-2025