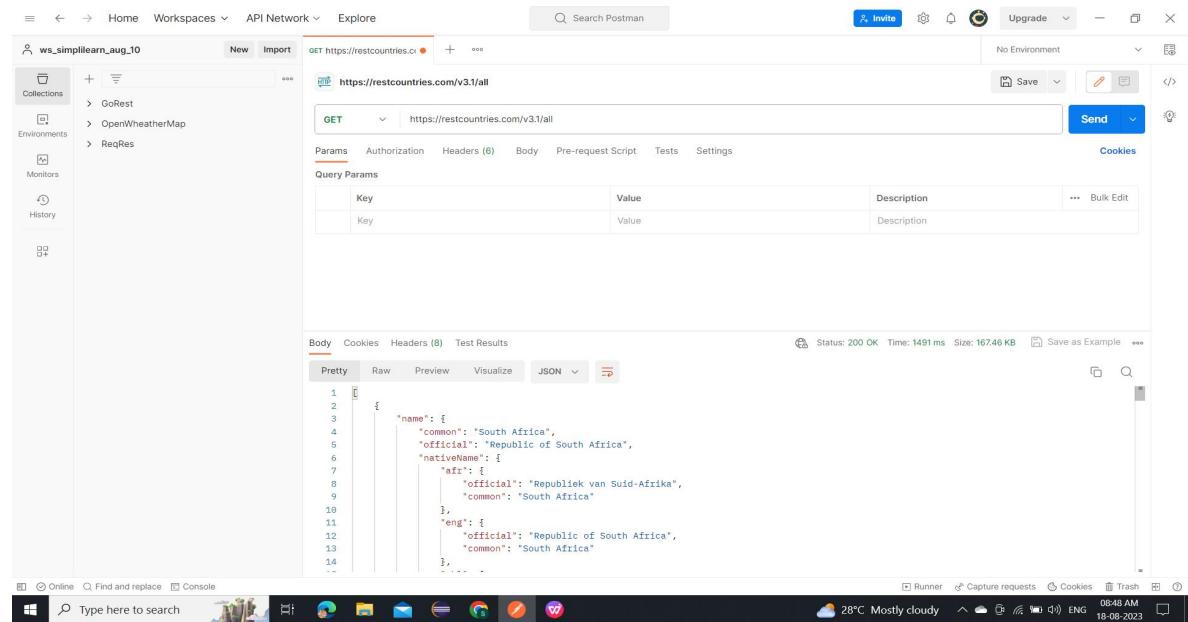


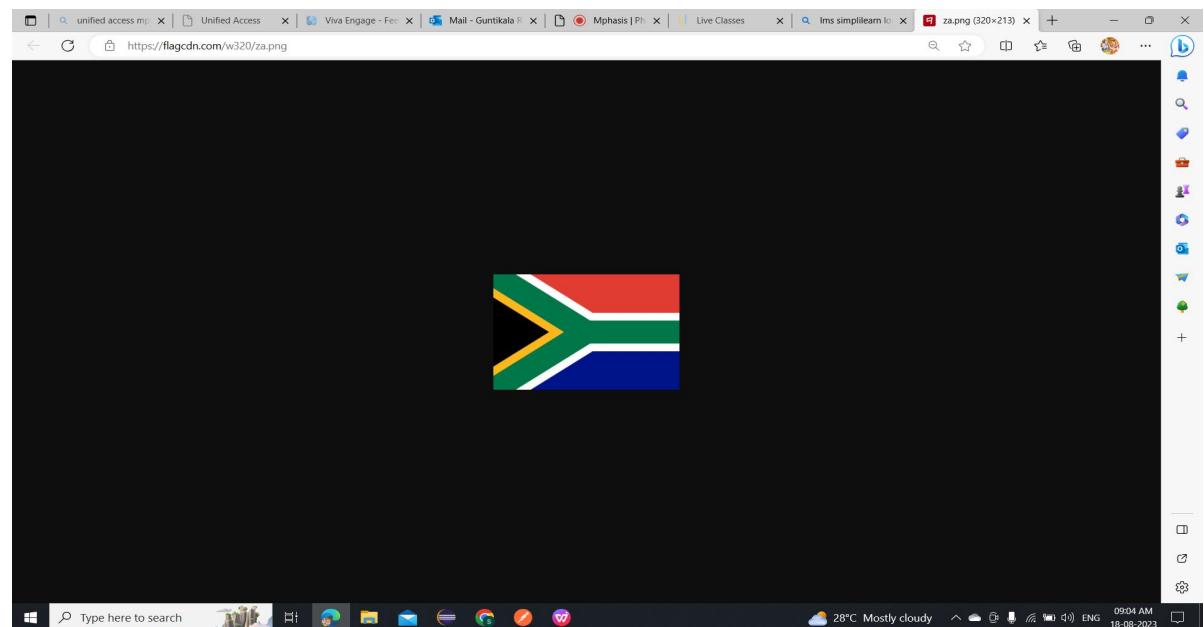
PRACTICE PROJECTS OF API TESTING

GET REQUEST:

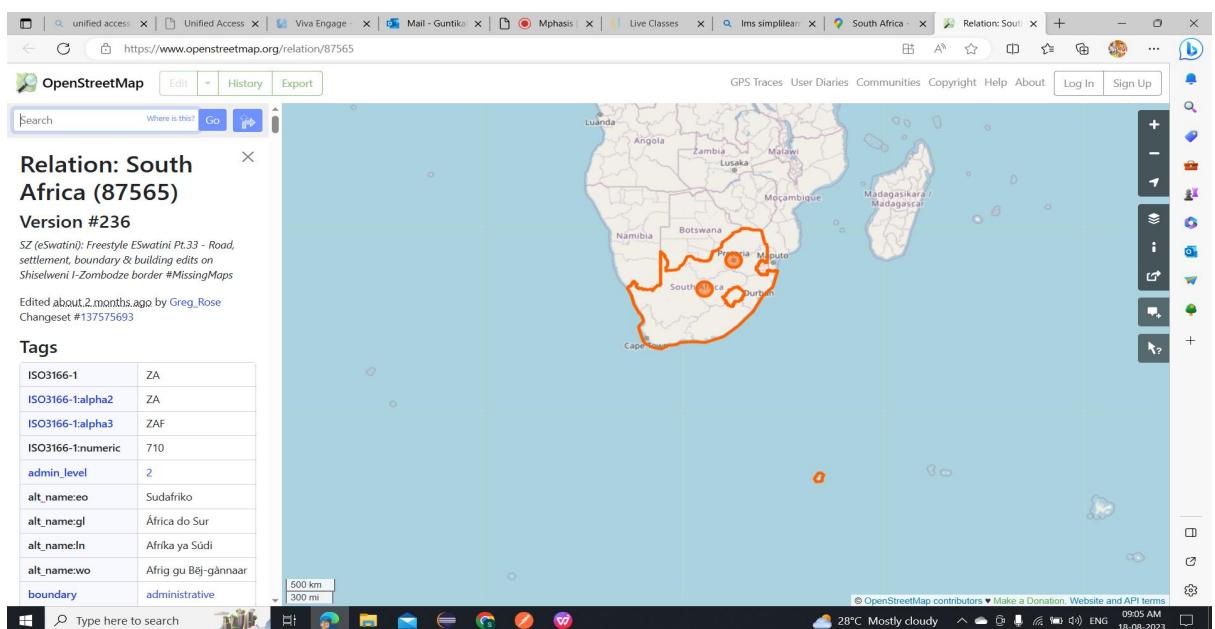
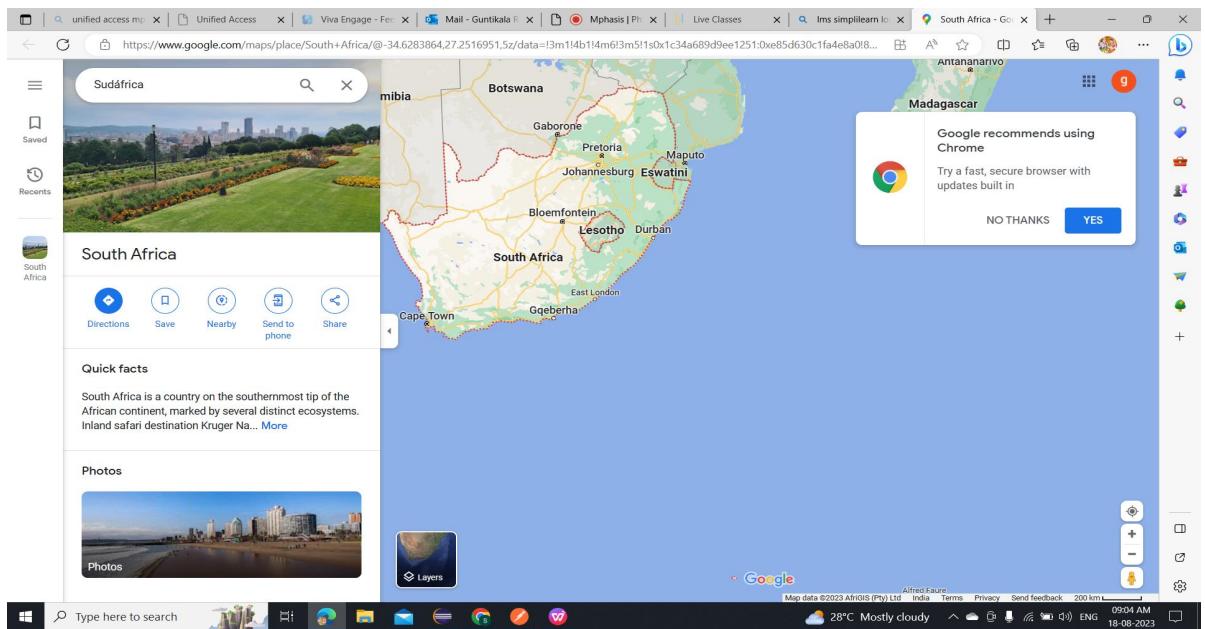


The screenshot shows the Postman application interface. A GET request is made to <https://restcountries.com/v3.1/all>. The response body is displayed in JSON format, showing data for South Africa. The JSON output is as follows:

```
1  {
2   "name": {
3     "common": "South Africa",
4     "official": "Republic of South Africa",
5     "nativeName": {
6       "af": {
7         "official": "Republiek van Suid-Afrika",
8         "common": "South Africa"
9       },
10      "eng": {
11        "official": "Republic of South Africa",
12        "common": "South Africa"
13      }
14    }
15  }
```



The screenshot shows a web browser window with multiple tabs open. The active tab displays the flag of South Africa (<https://flagcdn.com/w320/za.png>) against a black background. The browser's address bar shows the URL <https://flagcdn.com/w320/za.png>.



POST REQUEST:

The screenshot shows the Postman interface with a collection named "ws_simplilearn_aug_10". A POST request is being made to <https://gorest.co.in/public/v2/users>. The request body is a JSON object:

```
1 {  
2   "name": "Aravind",  
3   "email": "aravind95534@gmail.com",  
4   "gender": "male",  
5   "status": "active"  
6 }
```

The response status is 201 Created. The bottom status bar shows the date and time as 18-08-2023.

PARAMETRIZED REQUEST:

The screenshot shows the Postman interface with a collection named "ws_simplilearn_aug_10". A GET request is being made to <https://api.openweathermap.org/data/2.5/weather?q=Bangalore&appid=6a02f39bba891ba278b64ee12215cd9>. The request includes query parameters:

Key	Value	Description
q	Bangalore	
appid	6a02f39bba891ba278b64ee12215cd9	

The bottom status bar shows the date and time as 18-08-2023.

COLLECTIONS:

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade.
- Left Sidebar:** Collections (selected), Environments, Monitors, History.
- Collection View:** A collection named "ws_simplilearn_aug_10" is selected. It contains a "GoRest" environment and several requests:
 - POST Create User**
 - GET Verify User**
 - PUT Update User**
 - DEL Delete User**
 - New Collection** (selected)
 - OpenWeatherMap**
 - GET Current Weather**
 - GET https://restcountries.com/v3.1/...**
 - ReqRes**
 - POST Create User**
 - PUT Update User**
 - DEL Delete User**
 - GET All users**
 - POST xml**
- Right Panel:** Overview tab is active. It shows a "New Collection" button and a "Speed up your work with collection templates" section with options like Conditional workflow, Data visualization, REST API basics, and More templates.
- Bottom Status Bar:** Online, Find and replace, Console, Runner, Capture requests, Cookies, Trash, 32°C Mostly cloudy, 04:47 PM, 18-08-2023.

The screenshot shows the Postman interface with the "Runner" tab selected in the header bar. The "ws_simplilearn_aug_10" collection is still selected. The "Run order" section lists the requests to be run:

- POST Create User** (checked)
- GET Verify User** (checked)
- PUT Update User** (checked)
- DEL Delete User** (checked)

The right panel contains the "Functional" tab, which includes the following configuration:

- Choose how to run your collection:**
 - Run manually**: Run this collection in the Collection Runner.
 - Schedule runs**: Periodically run collection at a specified time on the Postman Cloud.
 - Automate runs via CLI**: Configure CLI command to run on your build pipeline.
- Run configuration:**
 - Iterations:** 1
 - Delay:** 0 ms
 - Data:** Select File
 - Persist responses for a session**: Unchecked
 - Advanced settings**
- Run GoRest** button.

The bottom status bar remains the same as the first screenshot.

ENVIRONMENT:

The screenshot shows the Postman application interface. On the left, the sidebar displays the workspace 'ws_simplilearn_aug_10' containing collections like 'GoRest', 'OpenWeatherMap', and 'ReqRes'. The main area shows a request for 'GET All users' with a query parameter 'page=2'. To the right, the 'Variables' tab lists 'url' with initial and current values 'https://qa.reqres.in'. The 'Globals' tab lists 'id' with initial value 'null' and current value '4495254'. A tooltip about variable reuse is visible. At the bottom, the taskbar shows system icons and the date/time.

TESTS SCRIPTS:

The screenshot shows the Postman application interface. The left sidebar displays the workspace 'ws_simplilearn_aug_10' with the 'GoRest' collection selected. The main area shows a POST request to 'https://gorest.co.in/public/v2/users'. The 'Tests' tab contains a detailed JavaScript test script. A context menu is open over the script, with the option 'Script with Postbot' highlighted. The taskbar at the bottom shows system icons and the date/time.

```

1 var response = pm.response.json();
2 pm.globals.set("id", response.id);
3
4 pm.test("Response status code is 201", function () {
5   pm.response.to.have.status(201);
6 });
7
8
9 pm.test("Id is a positive integer", function () {
10   const responseData = pm.response.json();
11
12   pm.expect(responseData.id).to.be.a("number");
13   pm.expect(responseData.id).to.be.above(0);
14 });
15
16
17 pm.test("Email is in a valid format", function () {
18   const responseData = pm.response.json();
19
20   pm.expect(responseData.email).to.match(/^(?:[a-zA-Z0-9]+(?:[-\.\_]?[a-zA-Z0-9]+)*@[a-zA-Z0-9]+\.(?:[a-zA-Z]{2,3})+$/, "Email format is not valid");
21 });
22
23
24 pm.test("Gender is either 'male' or 'female'", function () {
25   const responseData = pm.response.json();
26
27   pm.expect(responseData.gender).to.exist.and.to.be.oneOf(['male', 'female']);
28 });

```

API DOCUMENTATION :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (GoRest, ReqRes), 'Environments', 'Monitors', and 'History'. The main area displays a request for 'All users' with a GET method and a URL of `((url))/api/users?page=2`. The 'Params' tab shows a query parameter 'page' with a value of '2'. Below the request, there's a 'Response' section featuring a cartoon illustration of an astronaut pointing at a star.

BEARER TOKEN:

This screenshot shows the Postman interface again. It's focused on the 'Update User' endpoint with a PUT method and a URL of `https://gorest.co.in/public/v2/users/((id))`. A note says 'Need help authorizing with Go REST API?'. In the 'Authorization' tab, it specifies 'Type: Bearer Token'. A 'Token' field contains a long string of characters: `05172fc44401e43cf1bacf15ab1da175fc...` . The interface includes a 'Response' section with a similar cartoon illustration.

SOAP API REQUEST:

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections, environments, monitors, and history. The main area displays a request for a SOAP API. The request method is POST, directed to <http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>. The body is set to XML and contains the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
<soap:Body>
<CapitalCity xmlns="http://www.oorsprong.org/websamples.countryinfo">
<sCountryISOCode>IN</sCountryISOCode>
```

Below the body, the response status is shown as 200 OK with a time of 439 ms and a size of 552 B. The response XML is also displayed:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Body>
<m:CapitalCityResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
<m:CapitalCityResult>New Delhi</m:CapitalCityResult>
</m:CapitalCityResponse>
</soap:Body>
</soap:Envelope>
```