JSON Proficiency Assessment

**Part 1: Fetch JSON Data with JavaScript Fetch**

Objective:

Write a JavaScript function to retrieve data from an API endpoint and log the entire JSON response to the console.

Endpoint:

http://202.51.74.168:302/api/Patient/SearchRegisteredPatient?search=&searchUsingHospitalNo=false

Tasks:

Use the fetch() function to retrieve data from the provided API.

Log the JSON response to the console.

Handle any errors gracefully and log an appropriate message.

**Part 2: AJAX and HTML Display**

Objective:

Use AJAX to retrieve JSON data and display specific patient information in an HTML table.

Information to Display:

Short Name

First Name

Last Name

Age

Tasks:

Retrieve data using AJAX (or fetch() if preferred).

Populate an HTML table with the retrieved patient information.

Ensure that the table is properly formatted and accessible.

**Part 3: JSON Object (Basic)**

Objective:

Create a JSON object representing a patient with specific properties.

Properties:

Patient ID (number)

First Name (string)

Last Name (string)

Date of Birth (string, format: YYYY-MM-DD)

Gender (string)

Tasks:

Define a JSON object with the above properties.

Ensure the data types and format match the specifications.

**Part 4: JSON Object (Extended)**

Objective:

Extend the basic JSON object to include additional details.

Additional Details:

Address (object with Street, City, State/Province, Country)

Contact Phone (string)

Allergies (array of strings)

Medications (array of objects with Name, Dosage)

Tasks:

Add the specified details to your existing JSON object.

Use proper structuring and nesting of data.

**Part 5: Lab Results**

Objective:Create a JSON object representing a set of lab results.

Properties:

Test Name (string)

Date (string, format: YYYY-MM-DD)

Result Value (number or string)

Units (string)

Tasks:

Define a JSON object with the lab result properties.

Use the correct data types and ensure the date format is as specified.

**Part 6: API Fetch (if feasible)**

Objective:

Retrieve and display lab results using an API or sample JSON response.

Tasks:

If an API is available, fetch lab results and display them in an HTML table.

If no API is available, use the provided sample JSON response.

Display the Test Name, Date, Result Value, and Units.

**Part 7: Map for Medication Names**

Objective:

Use the map() function to extract medication names from a patient record.

Tasks:

Given a JSON structure with an array of medications, use map() to create a new array of medication names.

Ensure the new array only contains the names of the medications.

**Part 8: Map with Dosage**

Objective:

Modify the previous task to include dosages with medication names.

Tasks:

Adjust your map() function to include both the medication name and dosage in the format "Medication Name - Dosage".

Ensure the output is correctly formatted.

**Part 9: Complex Data and Transformation**

Objective:

Transform and format nested JSON data.

Tasks:

Given a complex JSON structure with nested patient name information, extract and format the full name as "Last, First".

Use appropriate JavaScript methods to manipulate and transform the data.

Instructions for Submission:

Write your solutions in a clear and readable format.

Comment your code to explain your logic and thought process.

Submit your code files along with a document explaining each part of the assignment and any assumptions made.

If you have created any additional functions or helper utilities, include them in the submission and document their usage.

Assessment Criteria: Correctness of the code.