# Test Cases for JSON Response Validation

## Validate JSON Structure

\*\*Objective:\*\* Ensure the JSON response adheres to the defined schema.

\*\*Test Steps:\*\*

- Fetch the API response.

- Validate the structure against the JSON schema.

\*\*Expected Result:\*\* JSON structure matches the defined schema.

## Validate Required Fields Exist

\*\*Objective:\*\* Ensure that all required fields (`api`, `mobile`, `courseTitle`, `price`) exist.

\*\*Test Steps:\*\*

- Fetch the API response.

- Check for the presence of `api` and `mobile` at the root level.

- Check if each object inside `api` and `mobile` arrays contains `courseTitle` and `price`.

\*\*Expected Result:\*\* All required fields are present.

## Validate Data Types

\*\*Objective:\*\* Ensure all fields have correct data types as defined in the schema.

\*\*Test Steps:\*\*

- Fetch the API response.

- Verify `api` and `mobile` are arrays.

- Verify `courseTitle` and `price` are strings.

\*\*Expected Result:\*\* Data types match the schema.

## Validate `api` and `mobile` Arrays Are Not Empty

\*\*Objective:\*\* Ensure that both arrays contain at least one entry.

\*\*Test Steps:\*\*

- Fetch the API response.

- Verify `api` and `mobile` arrays are not empty.

\*\*Expected Result:\*\* Both arrays contain at least one object.

## Validate `courseTitle` Field Contains Meaningful Data

\*\*Objective:\*\* Ensure `courseTitle` is not empty or null.

\*\*Test Steps:\*\*

- Fetch the API response.

- Validate that `courseTitle` has a non-empty string.

\*\*Expected Result:\*\* `courseTitle` contains valid course names.

## Validate `price` Field Contains Numeric Values

\*\*Objective:\*\* Ensure the `price` field contains valid numeric data represented as a string.

\*\*Test Steps:\*\*

- Fetch the API response.

- Validate that `price` contains numeric characters only.

\*\*Expected Result:\*\* `price` contains valid numeric values.

## Validate API Handles Missing `api` or `mobile` Keys Gracefully

\*\*Objective:\*\* Ensure the system handles missing required keys properly.

\*\*Test Steps:\*\*

- Fetch the API response with missing `api` or `mobile` key.

- Check system behavior.

\*\*Expected Result:\*\* The system should return a validation error.

## Validate API Handles Missing `courseTitle` or `price` Keys

\*\*Objective:\*\* Ensure the system handles missing `courseTitle` or `price` fields properly.

\*\*Test Steps:\*\*

- Fetch the API response with missing `courseTitle` or `price` key.

- Validate the error handling.

\*\*Expected Result:\*\* The system should return an error message.

## Validate API Handles Invalid Data Types

\*\*Objective:\*\* Ensure the API rejects responses with incorrect data types.

\*\*Test Steps:\*\*

- Modify `courseTitle` as an integer or `price` as a boolean.

- Validate if the system correctly flags the response.

\*\*Expected Result:\*\* The system should return an error for invalid data types.

## Validate API Handles Unexpected Fields

\*\*Objective:\*\* Ensure the API ignores or handles additional unexpected fields.

\*\*Test Steps:\*\*

- Add an unexpected field (e.g., `duration: 6 months`) to the response.

- Validate if the API response is still valid.

\*\*Expected Result:\*\* The system should ignore extra fields or return a valid response.