Assignment 3 - Solutions

Chakradhar Reddy - 112201022

September 22, 2025

1 Identify and Fix Errors in the JSON Data

The provided JSON data from the student management system contains several syntax and structural errors. Below is a detailed breakdown of the identified issues and the fully corrected JSON code.

1.1 Errors Identified

1. **Extra Root Braces:** The entire document is wrapped in an unnecessary outer pair of curly braces {}. A valid JSON document must have a single root element.
2. **Missing Comma (Student 2):** In the second student object (Alex Chen), a comma is missing between the `"id": 102` and `"name": "Alex Chen"` key-value pairs.
3. **Missing Comma in Array (Student 2):** In the `courses` array for Alex Chen, a comma is missing between the string literals `"CS101"` and `"CS102"`.
4. **Invalid Boolean Value (Student 2):** The value for the `active` key is `True` (with a capital 'T'). In JSON, boolean literals must be lowercase: `true` or `false`.
5. **Invalid Null Value (Student 2):** The value for the `advisor` key is `undefined`. This is not a valid JSON value. The correct way to represent a null or non-existent value is with the `null` literal.
6. **Missing Comma Between Objects:** A comma is missing between the closing brace } of the second student's object and the opening brace { of the third student's object within the `students` array.
7. **Data Type Inconsistency (Student 3):** The `id` for the third student is a string (`"103"`), while it is a number for the other two. For data consistency, it has been corrected to a number.
8. **Missing Comma (Root Object):** In the main object, a comma is missing between the `"last_updated"` and `"total_students"` key-value pairs.

1.2 Corrected JSON Data

The following code block contains the corrected, valid JSON data.

```
{
  "students": [
    {
      "id": 101,
      "name": "Sarah Johnson",
      "courses": [
        "CS101",
```

```
        "MATH200",
        "ENG150"
      ],
      "gpa": 3.85,
      "active": true,
      "graduation_date": null
    },
    {
      "id": 102,
      "name": "Alex Chen",
      "courses": [
        "CS101",
        "CS102",
        "STAT101"
      ],
      "gpa": 3.92,
      "active": true,
      "advisor": null,
      "notes": "Excellent student with strong analytical skills"
    },
    {
      "id": 103,
      "name": "Maria Rodriguez",
      "courses": [],
      "gpa": 3.67,
      "active": false,
      "special_programs": [
        "honors",
        "research"
      ]
    }
  ],
  "last_updated": "2024-09-15T10:30:00Z",
  "total_students": 3
}
```

Listing 1: Corrected student management system JSON data

2 Analysis of TOML Configuration File

This section provides answers to the questions based on the provided TOML configuration file for the web application.

a. How many feature flags are currently defined, and which ones are active?

- There are **two** feature flags defined in the configuration.
- The `"new_ui"` flag is active because its `enabled` key is set to `true`.
- The `"analytics"` flag is inactive as its `enabled` key is `false`.

b. What happens when the log file reaches 100MB?

- When the log file at `/var/log/myapp.log` reaches its maximum size of 100MB, the logging system will perform a **log rotation**. The setting `rotate = true` instructs the system to rename the current log file (e.g., to `myapp.log.1`) and create a new, empty `myapp.log` file to continue logging.

c. If you wanted to make the server accessible only from localhost, what should you change?

- To restrict server access to the local machine only, you must change the `host` parameter in the `[server]` section from `"0.0.0.0"` to `"127.0.0.1"` or `"localhost"`.

d. Calculate the total number of seconds that cached items will remain valid.

- The `[cache]` section defines `ttl = 3600`. TTL stands for "Time To Live" and is measured in seconds. Therefore, cached items will remain valid for **3600 seconds** (or 1 hour).

e. Explain the difference between the `[feature_flags]` and `[[feature_flags]]` syntax.

- The difference is fundamental to how TOML structures data:
- `[feature_flags]` (single brackets) defines a **table** (or dictionary/object). There can be only one table with this exact name in the file. Its JSON equivalent would be: `"feature_flags": { ... }`
- `[[feature_flags]]` (double brackets) defines an element in an **array of tables**. This syntax allows you to define a list of objects. Each time `[[...]]` is used with the same name, a new object is added to the array. Its JSON equivalent would be: `"feature_flags": [ { ... }, { ... } ]`