Feedback Received:

No details provided regarding the implementation. For final submission please do submit the design sketch and readme file mentioning how to run your code also, no testing (Unit testing, tests for edge cases, integration tests etc.) seems to be done at this stage so please implement that as well. - Please add better comments in your code to get better points for code quality.

Changes Made:

Removed WeatherData.java

- In the initial submission, the WeatherData.java class was used to handle weather data in a manual, object-based format. This approach was later found to be redundant and error-prone.
- Instead of manually managing weather data with WeatherData.java, the new implementation uses the Gson library for converting weather data to and from JSON. This significantly simplifies the handling of weather information and eliminates the need for the WeatherData class. The new approach also reduces potential errors associated with manual parsing and serialization.

Improved JSON Parsing and Error Handling

- Improved the JSON processing by adding null checks and handling missing fields gracefully.
- Added better error handling for network failures and invalid input, ensuring robust communication between servers and clients.

Added Detailed Design in the README

- README file was added, detailing how to run the Aggregation Server, Content Server, and GETClient, including how to configure the port and server URLs.

Detailed Comments

- Detailed comments were added.

Added Unit Tests and Integration Tests

- Different test cases including the edge cases, invalid inputs, empty data were implemented.