This document summarizes the unit tests for a Flask-based feedback API, as detailed in the FeedbackAppTestCase class. The tests utilize the unittest framework and an in-memory SQLite database for isolated and consistent testing.

### **POST /feedback Endpoint Tests:**

* **test\_post\_feedback\_success**:
  + **Purpose**: To verify successful submission of valid feedback.
  + **Input**: A JSON payload containing a message (string) and a rating (integer).
  + **Expected Outcome**:
    - HTTP Status Code: 201 Created
    - JSON Response: {"status": "ok"}
* **test\_post\_feedback\_invalid\_payload**:
  + **Purpose**: To ensure that requests with incomplete data are rejected.
  + **Input**: JSON payloads missing either the message or the rating field.
  + **Expected Outcome**:
    - HTTP Status Code: 400 Bad Request
    - JSON Response: An error message indicating that both 'message' and 'rating' are required.
* **test\_post\_feedback\_invalid\_rating\_type**:
  + **Purpose**: To validate that the rating field must be an integer.
  + **Input**: A JSON payload where the rating field is not an integer (e.g., a string).
  + **Expected Outcome**:
    - HTTP Status Code: 400 Bad Request
    - JSON Response: An error message indicating an invalid data format for 'rating'.

### **GET /feedback Endpoint Tests:**

* **test\_get\_feedback\_empty**:
  + **Purpose**: To confirm correct behavior when no feedback has been submitted.
  + **Input**: A GET request to /feedback with no query parameters.
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: An empty JSON array [].
* **test\_get\_feedback\_with\_data**:
  + **Purpose**: To test the retrieval of multiple feedback entries.
  + **Pre-condition**: Multiple valid feedback entries are posted.
  + **Input**: A GET request to /feedback with no query parameters.
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: A list containing all posted feedback items, sorted in descending order by default (most recent first).
* **test\_get\_feedback\_filter\_by\_rating**:
  + **Purpose**: To verify the filtering functionality based on the rating query parameter.
  + **Pre-condition**: Multiple feedback entries with varying ratings are posted.
  + **Input**: A GET request to /feedback?rating={value} (e.g., ?rating=5).
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: A list containing only feedback items that match the specified rating.
* **test\_get\_feedback\_sort\_asc**:
  + **Purpose**: To test sorting feedback by creation date in ascending order.
  + **Pre-condition**: Multiple feedback entries are posted.
  + **Input**: A GET request to /feedback?sort=asc.
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: A list of feedback items sorted from oldest to newest based on their created\_at timestamp.
* **test\_get\_feedback\_sort\_desc**:
  + **Purpose**: To test sorting feedback by creation date in descending order.
  + **Pre-condition**: Multiple feedback entries are posted.
  + **Input**: A GET request to /feedback?sort=desc.
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: A list of feedback items sorted from newest to oldest based on their created\_at timestamp.
* **test\_get\_feedback\_invalid\_sort\_parameter**:
  + **Purpose**: To ensure robustness when an invalid sort parameter is provided.
  + **Pre-condition**: Multiple feedback entries are posted.
  + **Input**: A GET request to /feedback?sort=invalid-value.
  + **Expected Outcome**:
    - HTTP Status Code: 200 OK
    - JSON Response: A list of feedback items, defaulting to the descending sort behavior (newest first), as if no valid sort parameter was provided.