littlepay

Test Cases:

To ensure comprehensive and effective testing of the Littlepay coding exercise, the following testing process can be implemented. It will cover both the **business logic** (matching tap on/off and calculating charges) and the **technical aspects** (CSV file reading/writing, system interactions).

Assumptions:

1. **Tap events are ordered**: The input file has `TapOn` events followed by corresponding `TapOff` events.

2. **Data format**: The input CSV is well-formed and does not contain missing or corrupted data. Each row will have `ID`, `DateTimeUTC`, `TapType`, `StopId`, `CompanyId`, `BusID`, and `PAN`.

3. **Trip matching**: We assume that each tap event corresponds to one trip (either completed, incomplete, or cancelled).

4. **Charge logic**: The price chart is fixed and only applies to trips between Stop1, Stop2, and Stop3.

Testing Process

1. Tests for Core Business Logic

- **Test Case 1: Completed Trip**
    - **Scenario**: Tap on at `Stop1`, tap off at `Stop2`. The charge should be `$3.25`.
    - **Expected Result**: The system calculates the duration of the trip, the stop IDs, and charges `$3.25` under the `COMPLETED` status.
- **Test Case 2: Incomplete Trip**
    - **Scenario**: Tap on at `Stop2`, no tap off. The charge should be `$5.50` (the highest charge for the stop).
    - **Expected Result**: The system charges `$5.50` for an incomplete trip and assigns it `INCOMPLETE` status.

- **Test Case 3: Cancelled Trip**
    - **Scenario**: Tap on and tap off at the same stop (`Stop1`). The charge should be `$0.00`.
    - **Expected Result**: The system identifies the trip as cancelled and assigns it `CANCELLED` status, with no charge.
- **Test Case 4: Multiple Trips with Same PAN**
    - **Scenario**: A passenger with the same PAN (`5500005555555559`) has multiple completed and incomplete trips.
    - **Expected Result**: The system matches each tap on/off event correctly and charges appropriately for each trip.
- **Test Case 5: Trip Fare Calculation**
    - **Scenario**: Calculate the Fare for a trip from `Stop1` to `Stop3`.
    - **Expected Result**: The charge should be `$7.30`.
- **Test Case 6: Invalid Tap Event Sequence**
    - **Scenario**: Tap off before tap on (e.g., tap off at `Stop1` without any tap on).
    - **Expected Result**: The system handles this gracefully and either ignores or flags the event as erroneous.

---

2. Check Integration Tests

- **Test Case 7: Reading CSV and Generating Trip CSV**
    - **Scenario**: Given an input CSV (`taps.csv`), validate the entire system can correctly read the input, match tap on/off events, and generate the expected output file (`trips.csv`).
    - **Expected Result**: The system processes the CSV correctly and outputs a valid `trips.csv` with proper trip details (including `StartTime`, `EndTime`, `ChargeAmount`, etc.).
- **Test Case 8: Multiple Customers and Stops**

- **Scenario**: Simulate multiple passengers and stops, ensuring the system can handle various trips for different passengers.

- **Expected Result**: Each passenger's trips are processed correctly with correct charges and statuses.

---

3. Edge Cases

- **Test Case 9: Missing fields in the input file**

  - **Scenario**:Some of the fields in the input file are empty.

  - **Expected Result**: System should handle the request gracefully and proper error message should be thrown based on the missing field.

- **Test Case 10: Large Input Data**

  - **Scenario**: The system processes a large input CSV file (1000+ rows) to test the scalability of the solution.

  - **Expected Result**: The system handles the large input efficiently and generates the correct output.