/SIDELINES

PART 1

Your task is to identify and fix all bugs on the existing code.
Download the code repo from here: https://be-test.s3.amazonaws.com/exam_part1.zip

Explanation

The file stream.json contains a stream of stock purchases on an hourly basis, at a certain day.
The code takes the data from the stream.json file and aggregates the total amount of stocks that were purchased on that day, per stock.
The result needs to be stored in the file: output-file.json

Example of desired structure of the output-file.json needs to be:
Pay attention that the numbers are just an example. The real numbers are different.

```
{
   "APPLE": 1550,
   "GOOGLE": 1660,
   "TESLA": 1470,
   "META": 1900,
   "AMAZON": 1480
}
```

PART 2

Your task is to write a system that retrieves upcoming games and presents them to users.
The system includes **two** components, a job and an API endpoint.

1) The Job

The job will fetch information on upcoming games and store it in the **games** table in the DB:

You can get all upcoming NFL games using the following url:

1. GET https://sql-api-wp-be.sidelines.io/games/upcoming/nba

games table description

1. **game_key** - unique game identifier
2. **game_date** - the date the game will be played

3. **game_time** - the time the game will be played
4. **away_team_abbr** - away team abbreviation, for example: ATL, JAC, DET…
5. **home_team_abbr** - home team abbreviation
6. **status** - the game status
7. **league_name**
8. **away_consensus_line** - the betting line of the away team in the wager (if the team is the favorite to win, it will have a negative value; if the team is NOT favored to win, it will have a positive value)
9. **home_consensus_line** - the betting line of the home team in the wager

Store only games that has all the following data :

1. **isEverGreen** = true
2. away_consensus_line is not null
3. home_consensus_line is not null

Support updates on the table: game_key field is a unique identifier and the updates will be based on this field.

2) The API endpoint

The endpoint will retrieve the following information from game table:

1. The game with the **smallest** consensus line gap between the home and away team.
2. The game with the **largest** consensus line gap between the home and away team.

If there is more than one of each, just return one of the smallest and one of the largest.

For example:
If the game table contains the following information

game_key	…	away_consensus_line	home_consensus_line
NFL@NYJ@JAC@2023-09-03	…	3.5	-3.5
NFL@DAL@ATL@2023-09-03	…	-4.5	4.5
NFL@LAR@MIA@2023-09-03		5.5	-5.5

You will return the following data in the endpoint:

```
[{
        gameKey: “NFL@NYJ@JAC@2023-09-03”
        type: “smallest”
        gap: 7
    },
    {
        gameKey: “NFL@LAR@MIA@2023-09-03”
        type: “largest”
        gap: 11
}]
```

DB connection details:

host: database-test.c4xgxkzb2pfl.us-east-1.rds.amazonaws.com
user: team
password: sidelinesteam
DB: teams

Wifi: sidelines4ever