The test should be done using the Java Spring Boot framework and MySQL database.

**Task 1:**

Develop an endpoint called ‘/import’ that accepts a CSV file that contains data of the following columns:

1. id (a running number starts with 1)
2. game\_no (an integer value between 1 to 100)
3. game\_name (a string value not more than 20 characters)
4. game\_code (a string value not more than 5 characters)
5. type (an integer, 1 = Online | 2 = Offline)
6. cost\_price (decimal value not more than 100)
7. tax (9%)
8. sale\_price (decimal value, cost\_price inclusive of tax)
9. date\_of\_sale (a timestamp of the sale)

**Task 2:**

1. Design and create a database table called ‘game\_sales’ that will store the content of the CSV.
2. Design and create the necessary tables to track the progress of the CSV import.

**Task 3:**

Design and develop an endpoint called ‘/getGameSales’ that returns the following result:

1. A list of game sales
2. A list of game sales during a given period.
3. A list of game sales where sale\_price is less than or more than a given parameter.

The endpoint should only return 100 records per request and must support pagination.

The endpoint should return results in less than 1 second.

**Task 4:**

Design and develop an endpoint called ‘/getTotalSales’ that returns the possible result:

1. The total number of games sold during a given period.
2. The total sales generated (total sale\_price) during a given period.
3. The total sales generated (total sale\_price) during a given period with a given game\_no.

The endpoint should return results in less than 1 second.

**Task 5:**

1. Prepare a CSV of 1,000,000 rows with random generated values based on the rules defined in Task 1.
2. ‘date\_of\_sale’ values to be populated randomly between 1 April to 30 April.
3. Load the CSV with 1,000,000 rows using ‘/import’ endpoint.

The import process should complete within 1 minute.

**Assessment Criteria:**

1. API development
2. Validation and Error handling
3. Performance of the endpoints
4. Scalability
5. Database management
6. Testing
7. Critical thinking

You may use additional middleware to achieve the above outcome.

Please upload all relevant source codes to a github account and share with us the link to the repository.

In the repository, please include a README file that contains the build and run steps.