Backend Project:

Project goal:

"BuyMe" website sanity testing extended by backend testing.

Solution architecture:

Development language: Java.

IDE: IntelliJ idea.

Third-Party usage: TestNG, Maven, JDBC.

Distribution type: Public.

Keep same guidelines as the previous project.

New guidelines:

1. Create a Singleton for instantiating and getting DB connection.
2. Website URL and browser type will be dynamic and stored inside a table called **config** in a remote DB
   - **config** table will have three columns:
     - config_id – (primary key, int, not null)
     - config_name – (varchar[45], not null)
     - config_data – (varchar[100], not null)
     - For example:

config_id	config_name	config_data
1	URL	https://www.buyme.co.il
2	BROWSER	chrome

3. The web test will run on the browser type and URL stored in the **config** table.
4. In case DB is unavailable - get configurations from your XML file (like in the previous project).
5. Write tests run results into **another** table called **history:**
   - **history** table will have two columns:
     - test_id – (primary key, int, not null)
     - test_date – (varchar[50]) which will store test run date (in any format)
     - For example:

test_id	test_date
1	2020-08-01 13:10:36
2	2021-01-02 10:04:10

6. In case DB is unavailable – write results into a local text file named results.txt

Steps:

1. Get run configurations from **config** table / XML (offline).
2. Run the web project (previous project) accordingly.
3. Write results to **history** table / text file (offline).

Extras (Place inside **Extra** class):

- Add Javadoc to your project.
- Read about prepared statements (in JDBC) and use it for insert statement.
- Create another table to write your history and save the **date** as DATETIME (and not varchar).
- In case there is no DB connection write your results into a CSV file (and not text).
- In sender and receiver screen – download an image from the web and upload it from code (instead of using a local image)
- Perform a REST request (GET) to the below URL to get the URL and the driver type using JSON parsing from your code.
  https://my-json-server.typicode.com/Dgotlieb/JSFakeServer/config