Practical
Full Stack Developer (Java & Angular)

Instructions

Develop an application with following requirements

1. **Backend – Java >= 17 Spring Boot 3.4.5**
2. **Frontend – Angular >= 18**
3. **Database – PostgreSQL**

NB: No user login/authentication is required (Just develop the functionality required)

a) Data generation (generate an Excel file with 1000000 records)

The system should be able to generate data and save in an Excel document.

The user should have a simple UI to enter the number of records to be generated and a button to send request to generate the data.

NB:

Test case: - User enter 1000000. The system should generate an Excel file with 1000000 records.

Data to be generated - Student records

Columns [studentId, firstName, lastName, DOB, class, score]

studentId – numeric incremental by one, starting value 1.

firstName – string (random alphabet characters) Min -3, Max 8

lastName – string (random alphabet characters) Min -3, Max 8

DOB – date (random date of birth between 1-1-2000 and 31-12-2010)

class – string (random class name **OPTIONS** [Class1, Class2, Class3, Class4, Class5])

score - numeric (random number between 55 and 75).

The generated Excel file location **MUST** be as follows

- For windows **"C:\var\log\applications\API\dataprocessing\<excel-file-name>"**
- or similar for linux

b) Data processing (Read Excel file and Save into CSV file)

The system should be able to read the Excel file generated in (step (a) Data generation) and save it in a CSV file.

The user should have a simple UI to pick the generated file from the local system and a button to start the task. Once user selected the file and click the button, the system should upload the file and process.

NB: **Before saving the records, the system should update the student scores and add 10 (student CSV score = student Excel score + 10)**

c) Data upload (Read CSV and save to Database)

The system should be able to read the CSV file generated in (step (b) Data processing) and save the data in SQL database

The user should have a simple UI button to pick the CSV file and upload for processing.

NB: **Before saving the records, the system should update the student scores and add 5. (student database score = student Excel score + 5)**

d) Report

Student report with following functionalities

1. Pagination
2. search by StudentId
3. filter – By Class - A dropdown
4. Export – Excel, csv, pdf

Submission:

- **Fill the tasks status and implementation performance in the table below.**
- **Host your projects on Github and add the repository links in the table.**
- **Reply to this email with this file attached.**

Task	Status (Done/Not Done)	Performance (in Mins)
a) Data generation (generate an Excel file with 1000000 records)		
b) Data processing (Read Excel file and Save into CSV file)		
a) Data upload (Read CSV and save to Database)		
b) Report		**Not Applicable**
GitHub Backend link		
GitHub Frontend link		