Practical

Full Stack Developer (Java & Angular)

Instructions

Tech stack:

- **Springboot – Java >= 17**
- **Angular >=18**
- **Microsoft SQL Server**
- **System architecture and System design (Your choice)**

1. **Frontend – Angular**

Should have a login page

Should have a dashboard – with a single tile showing the total number of student in the database

Should have the following menu

- **Data generation**
- **Data processing**
- **File upload**
- **Student management**
- **Student Report**

2. **Backend – Springboot API**

Should have all API end points

You are expected to design and develop a system that meet have the following user requirements

a) All Endpoints except login must be protected from unauthorized access. Use Spring security and JSON Web token (jwt). The token should be generated **ONLY** after successful login using Username and Password
NB: Test Users data **MUST** be generated and save in the database during application startup. Do not provide registration feature.

a) Data generation

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

The user should have a UI to enter the number of records to be generated and a button to send request to the API to generate the data and save in an excel file.

i. Best case (User enter 0) generate an Excel file with zero records.

ii. Worst case (User enter 1000000) generate an Excel file with 1000000 records.

Data to be generated - Student records

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

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 85).

status – numeric (0 – inactive, 1 – active) default 1- active

photoPath – string (Default – empty string)

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

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

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

The user should have a simple UI button to start the task. Once the user clicks the button, the system should pick the file from the path and process.

c) Data upload

The system should be able to read the Excel file generated in (step (a) Data generation) and save the data in Microsoft SQL database

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

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

d) Report & Export

Show a list of students with action buttons (View, edit, delete)

View – Show a detailed page with student details including the student photo. Show an avatar where student photo is not available.

Delete: -

- Should have a delete confirmation message
- Soft delete the selected record (Set the status to 0 – inactive)

Update/Edit: -

- Should allow the user to update the student details including the photo.
- Should allow the user to upload student photo (Photo upload should limit to **png/jpeg, max 5Mb**)
- Rename the filename and add a prefix of the student id number. Save the photo on the following path **"C:\var\log\applications\API\StudentPhotos\<studentId-filename>"**
- Update the respective student record in the database with updated data including the **photoPath**

e) Student management – Update/Edit student

The system should have student report with pagination.

The user should be able to export the report to Excel file

Reports filters – The report should have the following filters/search fields

a) Search by studentId
b) Filter by class - dropdown
c) Filter by dob (between – start date and end date) – use data picker

Submission:

- **Host your projects in Github and share**
- **Record and share a video demonstrating your work (max 30 minutes). Let us know your approach/design, optimization technique both application and database. Show case the worst case scenario.**
  Tell us about any challenge you encounter (if any) and your solution