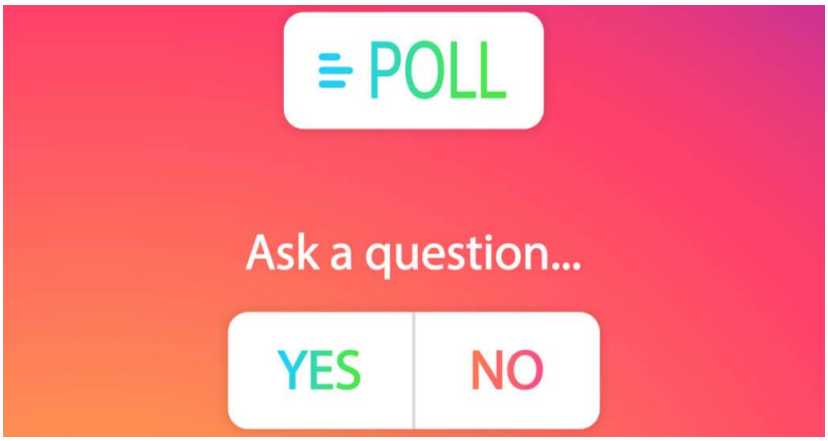
Polls System - Backend Implementation

Instructions:

1. Your mission is to create a poll system that will be used to ask the company customer's any questions they want and process calculations about the data.

for 

Example questions:

●Between the following, what do you most love to do?

a. Watch TV

b. Play the computer

c. Hanging out with friends

d. Travel the world

- Where is your preferred place to travel

  a. USA

  b. France

  c. South America

  d. Thailand

2. Your system should implement two different services:
   a. First service → **User service**
   b. Second service → **Poll service**
3. Let's understand what each service should be responsible for:

The User service → will be responsible to save all the data about your system users. Such data will be:

- Unique user id
- User first name
- User last name
- User email
- User age
- User address
- User joining date (The date the user was joined the company as a customer)

Each user in your system is a different company customer. Only registered users can answer your system poll questions.
If a user is not registered he can't submit answers to any questions.
If a user delete himself from your system, all the answers he gave to any question should be deleted as well.

You should support full CRUD implementation for your system users meaning you will need to provide API that allow create a user, update a user, delete a user and get user info.

The poll service → will be responsible to save all the polls questions and the user answers for each question.

Each question will be an american question which mean the user should get the question text and 4 optional answers. The user should choose the answer he wants from the 4 available options and your system should save the user choice accordingly.

For each poll question your system should save the following data:

- Unique question id (Each question should have a different id)
- The question title (Here you should save the question itself)
- The question first answer option
- The question second answer option
- The question third answer option
- The question fourth answer option

You should support full CRUD implementation for the poll questions meaning you will need to provide API that allow create a question, update a question, delete a question and get question info.

In addition you should think of a way of saving the answers of each user to each question.
Remember that it's possible that not all users will answers to all the available questions and it’s also possible that will be some questions without any answer.

Finally, your system should give the company the ability to get relevant info from the users answers to each question meaning you should support API that implement the following:

- By passing the question id → Return how many users choose each of the question options
- By passing the question id → Return how many users answer to this question in total
- By passing the user id → Return the user answer to each question he submitted
- By passing the user id → Return how many questions this user answered to
- Return all questions and for each question how many users choose each of the question options.

4. Your system should implement only the Backend part, no UI needed
   Your system will be written in **Java** and use **Spring Boot** framework & **H2 database**.
   Write the system according to all the best practices you learned in the course
   For example:
   - Use MVC architecture (Controller, Service, Repository)
   - Use configuration when needed
   - Write valid Java class names and SQL table names
   - Use the Request to Response flow as learned in class when needed

   Your services should communicate with each other by using the API endpoints you create.
   Each service should be connected to its own H2 database and save the data that is relevant only to its own responsibility.

Good Luck

ECOM SCHOOL

המכללה למקצועות הדיגיטל וההייטק