STEP 1: BUILD A SURVEY FORM

You will need to build a simple survey form.

The questions and answers of the survey should come from a database so it is possible to **add, edit, remove and reorder** questions and answers only by adding, editing, and deleting records from the database (with no need to touch the code or change the database schema).

There are two types of questions: radio (single answer) and checkboxes (multiple choice).

The completed survey will be saved in the database. Many users can submit the survey.

Insert these questions into your database (You don't need to build an admin page. You can simply insert with SQL queries):

```
How old are you? (radio)
- Less than 18
- 18-99
- More than 99

Are you happy? (radio)
- Yes
- No

What countries have you visited? (checkboxes)
- Spain
- France
- Italy
- England
- Portugal

What is your favorite sport? (radio)
- Football
- Basketball
- Soccer
- Volleyball
```

```
What programming languages do you know? (checkboxes)
- PHP
- Ruby
- JavaScript
- Python
```

STEP 2: BUILD A SURVEY RESULTS PAGE TO INVESTIGATE USER HAPPINESS

Build a simple page that shows for each question, what is the most popular answer for happy users and what is the most popular answer for sad users.

Example results page:

```
-------------------------------
RESULTS:

Most picked answers by happy people:

- How old are you? => More than 99
- Are you happy? => Yes
- What countries have you visited? => France
- What is your favorite sport? => Soccer
- What programming languages do you know? => Ruby


Most picked answers by sad people:


- How old are you? => 18-99
- Are you happy? => No
- What countries have you visited? => France
- What is your favorite sport? => Football
- What programming languages do you know? => PHP
```

NOTES

- You can use any platform, language, tool, and resource that you like, but no ORMs, we want to see your general SQL coding skills.

- You might need to "cut corners" to make this work in just 4 hours. The DB Schema is not the right place to cut corners, please make a proper DB schema and **do not make a "bad" DB design just to make your life easier for step 2!**
- **The primary goal is to see you complete the project within the time given**. Do not over-engineer!
- After you're finished you will sit down with one of our engineers to discuss and review your code.