Test Program Contents

Please do this test properly to produce best results. The work quality should reflect your level of experience and expertise.

A) Programming Test

1. Theme: Playing cards will be given out to n(number) people
2. Purpose: Total 52 cards containing 1-13 of each Spade(S), Heart(H), Diamond(D), Club(C) will be given to n people randomly.
3. Language to be used : PHP / Javascript / jQuery / ReactJS

   * You can use any or combination of those languages. If you decide to use a combination of languages, please provide in 2 types, front end & back end codes. (the more combination of languages used, the higher the chances for you to be called in for the interview)

   * These codes will be tested in the platform mentioned in 7(b). And chrome browser will be used as the testing browser.
4. Program file (source code) character code must be UTF-8 and line feed code must be LF.
5. Program Input :
   a. Number of people (numerical value)
   b. It does not matter how cards are given if recompile of program arguments, parameter, keyboard input and so on are not necessary.
   c. In case input value is nil or value is invalid then the error message of “Input value does not exist or value is invalid” must be displayed and the process must be terminated.
   d. Any number less than 0 is an invalid value.
   e. Greater than 53 are normal values and cards must be distributed to a number of people instead of having it as an error.
6. Output format :
   a. Spade = S, Heart = H, Diamond = D, Club = C
   b. Card 2 to 9 are, as it is, 1=A,10=X,11=J,12=Q,13=K
   c. The card distributed to the first person on the first row will be separated (comma),
   d. The card distributed to the second person on the second row will be separated(comma),
   e. [LF] is not allowed. Example:

      S-A,H-X,.....
      D-3,H-J,.....
7. Remarks :
   a. Please submit your work in Github repo.
   b. The written code must comply with either:
      i. Minimum supported environment PHP7, Node v12, React v16, MySQL v5.7 (depends on your chosen programming language); OR
      ii. Create, build and run code within a virtual development environment using Vagrant or Docker.
   c. Please enter comments if you think it is necessary.
   d. Please enter irregular processing to where there might be a possibility that an irregular occurs.

   e. Message “Irregularity occurred” must be displayed and the process must be terminated if any irregular occurs.
   f. All comments, usage manuals, and remarks must be explained in the source code.
   g. Please create the above and reply to me with the total time you have spent on.
   h. Reproduction / Reprint is prohibited.

B) SQL Improvement Logic Test

1. Purpose: These are a SQL query that produces a search result. The query produces results in approximately 8 seconds.
2. Task: Please suggest what improvements should be done to the query in order to improve its performance.
3. Remarks :
   a. Please submit your work in Github repo.
   b. The answer can be in the form of:
      i. Written explanation of the logical improvement; OR
      ii. Written SQL with the improved query; OR
      iii. If you set up a virtual environment in A) 7(b)(ii), please create the MySQL database and make the improvement.
   c. Please create the above and reply to me with the total time you have spent on.
   d. Reproduction / Reprint is prohibited.

------------------------- the SQL -----------------------------------

```
SELECT Jobs.id AS `Jobs__id`,
      Jobs.name AS `Jobs__name`,
      Jobs.media_id AS `Jobs__media_id`,
      Jobs.job_category_id AS `Jobs__job_category_id`,
      Jobs.job_type_id AS `Jobs__job_type_id`,
      Jobs.description AS `Jobs__description`,
      Jobs.detail AS `Jobs__detail`,
      Jobs.business_skill AS `Jobs__business_skill`,
      Jobs.knowledge AS `Jobs__knowledge`,
      Jobs.location AS `Jobs__location`,
      Jobs.activity AS `Jobs__activity`,
      Jobs.academic_degree_doctor AS `Jobs__academic_degree_doctor`,
      Jobs.academic_degree_master AS `Jobs__academic_degree_master`,
      Jobs.academic_degree_professional AS `Jobs__academic_degree_professional`,
      Jobs.academic_degree_bachelor AS `Jobs__academic_degree_bachelor`,
      Jobs.salary_statistic_group AS `Jobs__salary_statistic_group`,
      Jobs.salary_range_first_year AS `Jobs__salary_range_first_year`,
      Jobs.salary_range_average AS `Jobs__salary_range_average`,
      Jobs.salary_range_remarks AS `Jobs__salary_range_remarks`,
      Jobs.restriction AS `Jobs__restriction`,
      Jobs.estimated_total_workers AS `Jobs__estimated_total_workers`,
      Jobs.remarks AS `Jobs__remarks`,
      Jobs.url AS `Jobs__url`,
      Jobs.seo_description AS `Jobs__seo_description`,
      Jobs.seo_keywords AS `Jobs__seo_keywords`,
      Jobs.sort_order AS `Jobs__sort_order`,
      Jobs.publish_status AS `Jobs__publish_status`,
      Jobs.version AS `Jobs__version`,
      Jobs.created_by AS `Jobs__created_by`,
      Jobs.created AS `Jobs__created`,
      Jobs.modified AS `Jobs__modified`,
      Jobs.deleted AS `Jobs__deleted`,
      JobCategories.id AS `JobCategories__id`,
      JobCategories.name AS `JobCategories__name`,
      JobCategories.sort_order AS `JobCategories__sort_order`,
      JobCategories.created_by AS `JobCategories__created_by`,
      JobCategories.created AS `JobCategories__created`,
      JobCategories.modified AS `JobCategories__modified`,
      JobCategories.deleted AS `JobCategories__deleted`,
      JobTypes.id AS `JobTypes__id`,
      JobTypes.name AS `JobTypes__name`,
      JobTypes.job_category_id AS `JobTypes__job_category_id`,
      JobTypes.sort_order AS `JobTypes__sort_order`,
      JobTypes.created_by AS `JobTypes__created_by`,
      JobTypes.created AS `JobTypes__created`,
      JobTypes.modified AS `JobTypes__modified`,
      JobTypes.deleted AS `JobTypes__deleted`
FROM jobs Jobs
LEFT JOIN jobs_personalities JobsPersonalities
  ON Jobs.id = (JobsPersonalities.job_id)
LEFT JOIN personalities Personalities
  ON (Personalities.id = (JobsPersonalities.personality_id)
      AND (Personalities.deleted) IS NULL)
LEFT JOIN jobs_practical_skills JobsPracticalSkills
  ON Jobs.id = (JobsPracticalSkills.job_id)
LEFT JOIN practical_skills PracticalSkills
  ON (PracticalSkills.id = (JobsPracticalSkills.practical_skill_id)
      AND (PracticalSkills.deleted) IS NULL)
LEFT JOIN jobs_basic_abilities JobsBasicAbilities
  ON Jobs.id = (JobsBasicAbilities.job_id)
LEFT JOIN basic_abilities BasicAbilities
  ON (BasicAbilities.id = (JobsBasicAbilities.basic_ability_id)
      AND (BasicAbilities.deleted) IS NULL)
LEFT JOIN jobs_tools JobsTools
  ON Jobs.id = (JobsTools.job_id)
LEFT JOIN affiliates Tools
  ON (Tools.type = 1
      AND Tools.id = (JobsTools.affiliate_id)
      AND (Tools.deleted) IS NULL)
LEFT JOIN jobs_career_paths JobsCareerPaths
```

```
    ON Jobs.id = (JobsCareerPaths.job_id)
LEFT JOIN affiliates CareerPaths
    ON (CareerPaths.type = 3
        AND CareerPaths.id = (JobsCareerPaths.affiliate_id)
        AND (CareerPaths.deleted) IS NULL)
LEFT JOIN jobs_rec_qualifications JobsRecQualifications
    ON Jobs.id = (JobsRecQualifications.job_id)
LEFT JOIN affiliates RecQualifications
    ON (RecQualifications.type = 2
        AND RecQualifications.id = (JobsRecQualifications.affiliate_id)
        AND (RecQualifications.deleted) IS NULL)
LEFT JOIN jobs_req_qualifications JobsReqQualifications
    ON Jobs.id = (JobsReqQualifications.job_id)
LEFT JOIN affiliates ReqQualifications
    ON (ReqQualifications.type = 2
        AND ReqQualifications.id = (JobsReqQualifications.affiliate_id)
        AND (ReqQualifications.deleted) IS NULL)
INNER JOIN job_categories JobCategories
    ON (JobCategories.id = (Jobs.job_category_id)
        AND (JobCategories.deleted) IS NULL)
INNER JOIN job_types JobTypes
    ON (JobTypes.id = (Jobs.job_type_id)
        AND (JobTypes.deleted) IS NULL)
WHERE ((JobCategories.name LIKE '%キャビンアテンダント%'
        OR JobTypes.name LIKE '%キャビンアテンダント%'
        OR Jobs.name LIKE '%キャビンアテンダント%'
        OR Jobs.description LIKE '%キャビンアテンダント%'
        OR Jobs.detail LIKE '%キャビンアテンダント%'
        OR Jobs.business_skill LIKE '%キャビンアテンダント%'
        OR Jobs.knowledge LIKE '%キャビンアテンダント%'
        OR Jobs.location LIKE '%キャビンアテンダント%'
        OR Jobs.activity LIKE '%キャビンアテンダント%'
        OR Jobs.salary_statistic_group LIKE '%キャビンアテンダント%'
        OR Jobs.salary_range_remarks LIKE '%キャビンアテンダント%'
        OR Jobs.restriction LIKE '%キャビンアテンダント%'
        OR Jobs.remarks LIKE '%キャビンアテンダント%'
        OR Personalities.name LIKE '%キャビンアテンダント%'
        OR PracticalSkills.name LIKE '%キャビンアテンダント%'
        OR BasicAbilities.name LIKE '%キャビンアテンダント%'
        OR Tools.name LIKE '%キャビンアテンダント%'
        OR CareerPaths.name LIKE '%キャビンアテンダント%'
        OR RecQualifications.name LIKE '%キャビンアテンダント%'
        OR ReqQualifications.name LIKE '%キャビンアテンダント%')
        AND publish_status = 1
        AND (Jobs.deleted) IS NULL)
GROUP BY  Jobs.id
ORDER BY  Jobs.sort_order desc,
         Jobs.id DESC LIMIT 50 OFFSET 0
```