SkillsForge Second Stage Selection

Track: Backend Development

Congratulations on successfully passing the initial stage of our selection process for the internship position at SkillsForge. We are impressed with your qualifications and would like to invite you to the second stage, which involves a practical task.

This task will assess your technical skills and problem-solving abilities.

Please note that your chosen tasks will not affect your chances of being selected for the position. We want to provide you with an opportunity to showcase your skills and approach to problem-solving, regardless of the specific project.

Answer all from section one and one from section two

Section one

1) Implement a string compression. For example, aaaabbbccddddddee would become a4b3c2d6e2. If the length of the string is not reduced, return the original string.

Test Cases
assert compress(‘bbcceeee’) == ‘b2c2e4’
assert compress(‘aaabbbcccaaa’) == ‘a3b3c3a3’
assert compress(‘a’) = a

2) Given an array of strings arr[] of length n representing non-negative integers, arrange them in a manner, such that, after concatenating them in order, it results in the largest possible number. Since the result may be very large, return it as a string.

Example 1:

Input:
n = 5
arr[] = {"3", "30", "34", "5", "9"}
Output: "9534330"
Explanation:
Given numbers are {"3", "30", "34", "5", "9"},
the arrangement "9534330" gives the largest value.

Example 2:

Input:
n = 4
arr[] = {"54", "546", "548", "60"}
Output: "6054854654"
Explanation:
Given numbers are {"54", "546", "548", "60"}, the
arrangement "6054854654" gives the largest value.
Your Task:
You don't need to read input or print anything. Your task is to complete the function printLargest() which takes the array of strings arr[] as a parameter and returns a string denoting the answer.

Section two

Project 1: Blogging Platform API

Build a RESTful API for a simple blogging platform. Users should be able to create, read, update, and delete blog posts and comment on posts. Implement basic CRUD operations for managing blog content.

Requirements:

CRUD Operations: Develop endpoints for creating, reading, updating, and deleting blog posts.
Comments: Enable users to add comments to existing blog posts.
Search Functionality: Include search functionality to allow users to search for specific blog posts based on whatever category you deem fit

Additional Features (Nice to have. Doesnt affect grade):

Pagination: Implement pagination for listing blog posts to improve performance and user experience.
User Authentication: Implement user authentication to ensure only authorized users can create, update, and delete posts.

Project 2: SIWES Logbook API

Create an API for logging daily activities, allowing users to fill their logbooks for today and future dates, retrieve all filled logbook entries, and overwrite entries for already filled dates. The API should support CRUD actions for logbook entries.

Requirements:

CRUD Operations:
Users should be able to create, read, update, and delete logbook entries.

Date Restrictions:
Users can only fill the logbook for today and future dates.
If a user attempts to fill an empty date for today, the API should automatically fill the logbook entry for the current date.

Overwrite Existing Entries:
If a user fills the logbook for a date already filled, the API should overwrite the existing entry with the new data.

Additional Features:
Filtering/search options based on date range, activity type, or any other relevant criteria.

Going the extra mile
Enable users to export their logbook data/Notes in various formats (pick one. e.g., CSV, PDF) for backup or sharing purposes.

Submission Guidelines:

- Write your program in any language of your choosing.
- Work with any database of your choosing. In-memory data structures like Lists can also be used.
- If there is any relevant codebase to your submission, push it to GitHub and provide the link in the email and report.
- Push your well-formatted program, answers to (i) and (ii) to a public Github Repo
- Include a README that explains how to run your program including details of your program logic.
- **Nice To Have:** For submissions that require specific environments to run, it would be best to dockerize the application.

Submission Deadline:
Please submit your completed task by the 28th of April, 2024 using this link. Note that late submissions will not be considered.

We look forward to reviewing your work and discussing it further during the next stage of the selection process. If you have any questions or need clarification, please don't hesitate to reach out to us by replying to this email.