SENIOR BACKEND ENGINEER ASSESSMENT

Instructions:

- This assessment comes in two (2) parts and is a detailed simulation of what your work at Duplo will entail if hired.
- Part A tests your technical skills. Please attempt all questions
- Part B will test your leadership, business ethics and cultural fitness. For this part, please click on the link, read instructions, type in your full name and start. Once submitted, your assessment will be received on our admin dashboard.
- The deadline to complete all assessments is stated in the assessment email.

PART A

Question 1

Write a query that prints a list of employee names (i.e.: the name attribute) for employees in Employee having a salary greater than per month who have been employees for less than months. Sort your result by ascending employee_id.

Input Format

The Employee table containing employee data for a company is described as follows:

Column	Type
employee_id	Integer
name	String
months	Integer
salary	Integer

where employee_id is an employee's ID number, name is their name, months is the total number of months they've been working for the company, and salary is their monthly salary.

Sample Input

employee_id	name	months	salary
12228	Rose	15	1968
33645	Angela	1	3443
45692	Frank	17	1608
56118	Patrick	7	1345
59725	Lisa	11	2330
74197	Kimberly	16	4372
78454	Bonnie	8	1771
83565	Michael	6	2017
98607	Todd	5	3396
99989	Joe	9	3573

Sample Output

Angela
Michael
Todd
Joe

Explanation

Angela has been an employee for month and earns per month.

Michael has been an employee for months and earns per month.

Todd has been an employee for months and earns per month.

Joe has been an employee for months and earns per month.

We order our output by ascending employee_id.

Question 2: Build a RESTful API for a Blog

Instruction:

Build a RESTful API for a simple blog platform. The API should support the following operations:

Create a new blog post with a title, content, and author.
Retrieve a list of all blog posts.
Retrieve a single blog post by its ID.
Update a blog post.
Delete a blog post by its ID.
Implement pagination for listing blog posts.

Requirements:

- Use TypeScript for the project.
- For Nodejs framework, use fastify
- Store blog post data in a database (PostgreSQL).
- For interacting with the PostgreSQL database, use Prisma
- Implement validation to ensure each post has a title, content, and author.
- Add timestamps (created and updated) to blog posts.
- Write unit tests for the API routes.
- Write the code with the same mindset as you would for a real world project
- The solution should be submitted in the form of a deployed public server and the code should be shared in the form of a gzipped/zipped git repository over email

Question 3

What is logged first?

```
function hungry() {
  eatFruits();
  console.log("I am hungry.");
}
function eatFruits() {
  console.log("I'm eating fruits");
}


hungry();
```

Question 4

What are the syntax rules of JSON?

or

Write a json doc that supports all data types.

PART B

https://assessment.testgorilla.com/testtaker/publicinvitation/3f123509-87dd-40ee-9197-8c38e6c96a4b