**Welcome to the NocoDB Technical Round!**

*Please read the instructions carefully until the end.*

**Interview Guidelines**

This interview is designed to evaluate how you think and build a feature from start to finish. The test will involve both frontend and backend development with a database.

You are free to use any frameworks or packages as needed.

**Step 1: Provide high-Level Design (10-20 mins)**

* Create a HLD for the system in writing or a canvas of your liking ([tldraw](https://www.tldraw.com/), [drawio](https://www.drawio.com/) etc.)

**Step 2: Provide low-Level Design (10-30 mins)**

* DB
* APIs
* Frontend

Write down your LLD here.

**Step 3: Implementation (80-110 mins)**

* Implement the design you have conceived. Depending on your approach and progress, we may decide if you can finish the remaining bits as a take-home task.

**Task: Table Browser + Table Creator with CSV**

You are tasked with building a database table browser that includes a CSV uploader functionality, as outlined in this [Figma design](https://www.figma.com/design/mtBcpHT89wlP99PvzGRk6v/Interview?node-id=1-240&t=C4FyeIiZErLxb8aB-1). Pixel perfection is not required—consider the UI design as a guideline.

Please review the details thoroughly before starting.

**Task Overview**

Create a simple web application with the following functionalities:

* A user should be able to create a table by uploading a CSV file.
* A user should be able to list all tables of the connected database via the backend.
* A user should be able to browse rows in a table, with pagination support.

**Technical Requirements**

* **Backend**: Use Node.js for backend implementation.
* **Database**: Any relational database (PostgreSQL, SQLite, etc.) can be used to store the data.
* **Frontend**: Use any frontend framework (Vue, React, Angular, etc.) to display the tables.

**Links**

* [Figma Design](https://www.figma.com/design/mtBcpHT89wlP99PvzGRk6v/Interview?node-id=1-240&t=C4FyeIiZErLxb8aB-1)
* [Sample CSV](https://github.com/mertmit/nocofs/blob/main/assets/demo.csv) : https://github.com/mertmit/nocofs/blob/main/assets/demo.csv

**Helpful Notes**

To get a list of existing tables within a database, you can use the following queries:

* SQLIte

SELECT name

FROM sqlite\_master

WHERE type = 'table';

* PG

SELECT table\_name

FROM information\_schema.tables

WHERE table\_schema = 'public'

AND table\_type = 'BASE TABLE';

* MySQL

SHOW TABLES;

**Happy coding!**