

# Atlan Frontend Internship Task 2025

## Outline

Create, design and implement a web-based application capable of running SQL queries and displaying the results of said query. The application must include a space which accepts SQL queries in the form of user inputs, then runs the given query, and displays the result within the application..

Please note:

- **This is a dummy application.**
- **You must include a space to accept SQL queries.** This can be as simple as a text area or as complex as a full-blown code editor. This does not figure in the judging criteria, though.
- Your application **does not need to have a backend, a query engine or query validation.**
- You **do not need to add any syntax validation.**
- The data that your application displays can be any chunk of data. It does not need to be the actual result that the query would display if it were actually run. You can choose any block of data from [here](#) or you can choose any other data source. **The data and the query do not need to be in sync.**
- You can write down any text as the query as you wish. **The query syntax is not important, nor is the result.** You are free to put in dummy stubs and mocks wherever the need arises.
- **Your application must have more than one query accompanied by its corresponding table of data.** You can have a predefined set of queries and a predefined set of tables showing some data. You must also provide a way for us to toggle between these queries and choose any query and view the table of data accompanying it. It can be a dropdown or any other toggling mechanism.

# Breakdown

## Ideation

- Put on your thinking caps for this stage. Think of all the features you feel would be mandatory for this application. Features that make the application just usable. Something that, if removed, would render the whole application pointless. For example, a form is usable only if it has inputs and a way to submit the given form.
- Now that you have a list of essential and crucial features, start thinking of features that will add more value to your application, features that act as top-ups, cherries on the cake, if you will. These additional features make your application more useful and make your user's life a tad bit easier. Think of forms with inputs, buttons, *and custom field validations*.
- *Assume that a data analyst is going to use your application for an entire day. What are the features that will help them breeze through the day's workload?*

## Design

- **This is not a design-centric role, nor will you ever be asked to design any part of the product if you end up joining us.**
- In this stage, scribble out a rough layout of your webpage. Decide where the blocks of your application will go. **You will not be judged on the basis of your design proficiency.** This part mainly tests your ability to predict your user's behaviour, anticipate how they would use your product and the way you go about enhancing the user flow or the user experience..

## Implementation

- That's it. You have the list of features you wish to implement and the layout of those features. Now, get to coding, but only after you've gone through the **Notes below**.

# Submission

## Video Submission:

- Record a 5–7 minute video where you:
  - Walk through the application and demonstrate its key features.
  - Explain the system design and technical decisions made.
  - Discuss any challenges you faced while building a system of this scale, and how you resolved them.

## Submission Guidelines:

- **Code:** Upload your project to a GitHub repository and share the link. To prevent plagiarism, we recommend using a meaningless or obfuscated project name.
- **App:** Optionally deploy the app using a platform like Vercel, Netlify, or Heroku, and share the link.
- **Documentation, ER Diagram, and Video:** Share links to your architecture diagram, ER diagram, explanation document (PDF), and video walkthrough.

# Judging Criteria

- How well did you think out and plan the core and advanced functionalities of your application?
- Did you take into account the most basic needs of your user?
- What was your thought process behind choosing the features that you did?
- What features did you include to add value to the application? Do these add value or increase complexity?
- How well did you plan your layout? Does this layout enhance the overall experience for the user?
- Have you anticipated the actions your user will take?
- Your code structure, quality and overall readability.
- The load time of your application.
- The overall “snappiness” of your application: this includes the initial load time, the time taken to re-render, and the performance of the web application.

- Adherence to the fundamental practices of coding.

## Notes

1. Use a JavaScript framework for your application, such as [React](#), [Vue.js](#), [Svelte](#), or any other framework. **DO NOT CODE IN VANILLA JAVASCRIPT.**
2. **DO NOT USE TAILWIND CSS for styling**
3. Host your code on a version control system, such as GitHub, GitLab or BitBucket. **SEND US THE LINK TO YOUR PROJECT, NOT A ZIP FILE.**
4. Deploy your application on services such as [Netlify](#) or [Vercel](#), and send us the link to your deployed project. **DO NOT EXPECT THE JUDGE TO INSTALL YOUR APPLICATION ON A LOCAL INSTANCE.**
5. You are free to use any external library and dependency, as long as you mention it in the README.
6. In the README of your project, include the following details:
  1. A basic overview of your project.
  2. The JavaScript framework you chose, along with any major plugins or packages you installed.
  3. The page load time of your application, and how you measured this time.
  4. Any optimisations you did to decrease the load time or increase performance.

You get brownie points if you can render a large amount of rows in your application without breaking the browser, or without crashing it. If you can't, no biggie.