**Requirements for developer evaluation exercise**

**A. Summary**

Create a dashboard that displays all lenders associated with “Constant Servicer” and the loans they own. (Note that servicer, lenders and the data provided in the attached spreadsheet are all mock and not real).

1. A middleware server that retrieves lender and loan data from a relational DB and returns a JSON response. Python should be used along with your favorite Python framework. (Flask, FastAPI, Django). For relational DB, PostgreSQL is preferred, but MySQL is fine as well.

2. A React front-end that parses the JSON, written in Javascript ES6 or above or TypeScript. The components should be either functional or class based, with functional preferred.

**B. Details**

**1. Data**

In the chosen database, create:

* A lenders table with data from the Lenders sheet.
* A loans table with data from the Loans sheet.

**2. Middleware / API server**

1. Create an API server that contains three endpoints:

* /lenders - Returns all lenders in the lenders database table.
* /loans - Returns all loans in the loans database table.
* /loans/<lender\_id> - Returns all loans associated with a lender.

2. Each endpoint should send a JSON response.

3. Write some tests (needs to be comprehensive only if time permits).

**3. React front end to implement wireframe**

1. Components

Header component

* Display a logo (your choice) in the top left corner and servicer name bold faced and left aligned.

Left navigation menu

* Label (‘Lenders’) can have any styling you feel appropriate
* Each lender in the left will be a link which when clicked will load content into the main window.
* The content will be specific to the lender.

Main content window

* First component is a lender information box, the second displays all loans owned by the lender. The data is lender specific and loaded when clicking a lender in the navigation menu.
* The lender information box displays the lender name, address, city, state, zip and phone number.
* A loans table that listed all loans owned by the lender.
* See wireframe for a suggested layout.

2. Write some Jest tests to verify the components.

3. Data formatting examples

Currency - $12,221.11; Phone - 555-333-1212;. Rate - 7.50% for DB value of 0.75; Date - mm/dd/yyyy.

**4. Extra credit**

* Use SQLAlchemy in the Python middleware server.
* Code the front end using TypeScript instead of JavaScript