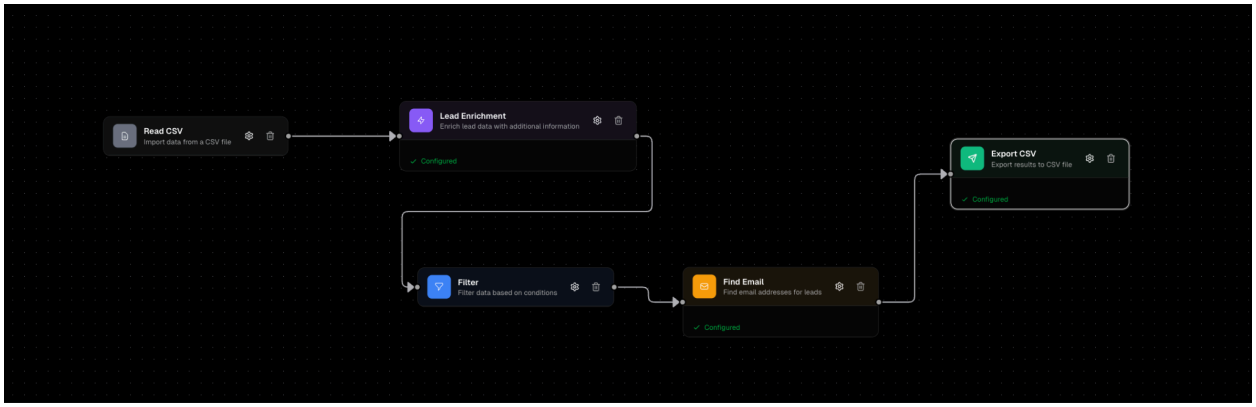
Fullstack Engineer – Take Home

Task:

Build a simplified replica of Sixtyfour's Workflow Engine with both a frontend and a backend. The goal is to allow users to configure and execute workflows made up of modular blocks. Here's how it looks:



Now your job is to build a frontend + backend that can do the following:

Backend Requirements

At a minimum, the backend should support the following blocks:

1. **Enrich Lead** – Use the /enrich-lead endpoint.
2. **Find Email** – Use the /find-email endpoint.
3. **Read CSV** – Load a CSV file into a dataframe.
4. **Filter** – Apply filtering logic to the dataframe (similar to pandas, e.g., df[df['name'].str.contains('64')]). Only rows that return True should pass to the next block.
5. **Save CSV** – Save the current dataframe back to a CSV file.

The blocks should be fully chainable in any order, enabling users to build configurable workflows. You do not need to implement the Sixtyfour enrichment endpoints themselves—simply call and use them as described in the documentation. The backend's main responsibility is to manage a dataframe and execute jobs against it, while also incorporating concepts like parallelization, asynchronous job handling, and efficient execution management.

Frontend Requirements

- Provide a simple UI where users can configure workflows by arranging blocks in sequence. [drag and drop would be ideal], no need to worry about authentication, and other stuff for this assignment.
- The user should be able to see the progress on the jobs.
- Allow users to specify parameters for each block (e.g., filter conditions).
- Display intermediate and final results in a clear way.

Example Workflows

1. **Basic Workflow:**

   Read_csv → Enrich_lead → Save_csv

2. **Filtered Workflow:**

   Read_csv → Filter (company name contains 'Ariglad Inc') → Enrich_lead (return educational background, including undergrad university) → Add boolean field is_american_education → Filter (is_american_education = True) → Save_csv

For testing, you can use this sample CSV file:

Sample CSV

Note: No need to implement persistent storage. Local files are fine.

Evaluation Criteria

We will score your project on:

- **Product Experience** – How intuitive and smooth the UI feels.

- **Backend Stability** – How reliably the backend executes workflows. Since each Sixtyfour api call takes a long time, how do you make it run faster?

- **State Management** – How well you handle and transition between different data states.

Discussion Topics (No Need to Implement)

Be ready to discuss the following in your follow-up:

- How would you implement the enrich_company endpoint?
- How to prevent incompatible blocks from being chained together (e.g., enrich_company should not connect to a lead block).
- How would you scale the backend to process thousands of rows in a CSV? How will you make sure it remains fast?
- Product decisions you made, the tradeoffs behind them, and what you might change with more time.
- What you learned about the Sixtyfour API, what changes you would suggest, and how you would improve it.

PS: When you log in to app.sixtyfour.ai, you will automatically receive $25 in API credits. If you need more, let us know.