Background

Workflows are quickly emerging as a major product category with the rise of AI. Tools like [n8n](#), [Gumloop](#), and [OpenAI's agent builder](#) are gaining traction because drag-and-drop blocks combined with AI-driven nodes make automation accessible, flexible, and intuitive.

At Valyx, we're building in this direction—creating a workflow-first experience that fundamentally transforms how users interact with our platform and unlocks what they can do with it.

Our customers rely on manual steps to manage their account receivables, but these can be automated through workflows built with our existing backend tools. What's missing is an intuitive, polished interface that lets users access and configure these workflows easily. We need a clean, user-friendly UI to unlock the automation we've already built.

Terminology

- **Node** - A node is a block in workflow that does some kind of action. This could be an AI agent, a dataframe parser, a slack notification sender etc. There are mainly three types of nodes in our system.

    - **Trigger (Node)** - A node which is the starting point of workflow. For ex - A workflow can be started when a mail is received or based on a timer or anything. **There should be at least one trigger node in workflow.**

    - **Controller (Node)** - These are condition based routing nodes. For ex - if X happens then go to node A else to node B

    - **Activity (Node)** - A node that performs some task. Most of the nodes are of type activity.

- **Node Params** - Params that a user can define while creating an activity node. For ex - for a AI agent like node they may specify what model to use or for a slack notification node, they may specify webhook URL

- **Edge** - Connection between 2 nodes

- **Workflow** - A set of collections of nodes and edges.

Problem Statement

API Details

- Base URL - https://rubik.valyx.com/
- Documentation -
  https://gist.github.com/anirudh-valyx/d15192e79fd2456bdba45123f870195e

Go through the api reference and try to understand what they are used for. Based on that, we want to do the following -

- Create a canvas where a workflow structure can be rendered.
- Read the workflow with ID = {{twflow_b210db0a85}} and display on canvas. This workflow already has some defined nodes and edges.
- Show users a list of nodes that they can select. Users can drag and drop these nodes on the main canvas.
- Allow connection between nodes (edges). Trigger nodes should not have any incoming edge.
- Users can modify the rendered workflow and update.

Note 1 — You may address these points in any order and may choose to submit without all points covered. The solution doesn't need to cover every possible case, but whatever is delivered should be polished and high quality.

Note 2 - Using n8n for reference is highly recommended.

Judging criteria

- A working solution where every implemented piece functions reliably.
- A user-first, intuitive experience that feels smooth, clear, and easy to understand, with thoughtful details that improve usability.
- Clean, modular code with strong structure and effective use of third-party libraries.