Write a React JS application supporting the given operations. Make use of the UI references and conditions given below.

App Operations

- Login screen
- Get all workflows screen
- Create/Edit a workflow screen
- View/Shuffle a completed workflow in the Create/Edit workflow screen

Workflows & Nodes

A workflow consists of nodes, arranged in sequential order. Workflows can have only two states, *completed* or *pending*.

Each node has two attributes, which can be filled by a user - *Title* & *Content.* And three states, which can be modified by a user: *Pending*, *In Progress* & *Completed.*

Rules & Requirements

- A workflow should be considered complete only if all its nodes are in the *Completed* state. Workflow state is toggleable from the workflow home screen provided it meets the above criteria.
- In a workflow there should be no *Pending* node before a *Completed* node, i.e nodes can be moved to *Completed* state only in the same sequential order as they are present in the workflow.
- All the nodes in a workflow should editable at all times.
- In a workflow, there should not be an empty node.
- In a workflow, nodes should be added or deleted only at the end of the sequence.
- If a workflow is in a *Completed* state, the user should be able to perform an action called 'shuffle' which will shuffle the order of the nodes in the workflow. The shuffle action should only be visible for *Completed* workflows.

UI References

Use the following screens as the reference, comments are add inline to help you understand the requirements & functionality. You are required to implement the functionality as mentioned in the following references.

- Login Screen: https://imgur.com/JnHbfW
- Workflow Home Screen - Get all workflows screen: https://imgur.com/kwQjp0S
- Workflow Create/Edit Screen: https://imgur.com/JpppsDy

Evaluation

- Clean, modular, elegant, and bug-free code.
- Good design and coding practices, like state-management, decoupling of data, and UI layer, correct use of component lifecycle.
- Handling of corner cases.
- Good UI.

Submission

- Github: It is advised to use git, and push the code to Github(or Bitbucket, Gitlab, etc). Commit frequently, we would like to see how the code evolved by looking at your commits. Share the public URL to the repo for the submission.
- Hosting: Bonus points for hosting the app and sharing the URL (on Heroku, AWS Github Pages, etc.)
- Contact: Please write to payal@automate.io for any queries.