**Introduction:**

Welcome to the project overview for developing a flow in the existing content repository akin to Google Drive. This endeavor requires building a foundational flow and modifying an existing flow for the content portal. As a core member of the product team, you would be required to integrate existing endpoints with front end flows and producing desirable results.

**Primary Objective:**

The core objective of this project is to create the Rename flow which involves the below steps

1. Designing the Rename flow as shown in the design file.
2. Implement a multi selector for the rows retrieved.
3. When a content or a folder is selected, the rename button appears dynamically and the add button disappears.
4. When Rename is clicked, you see the popup box with the new name to be entered.
5. When the user clicks Save, the “**/edit-content/:contentId**” endpoint from the content.js file is evoked if it is a content or “**/edit-folder/:folderId**” from the folder.js file is evoked and the modified name is saved.
6. You can distinguish between a content and a folder using the endpoint “**/get-table-name**" and passing the id of the record to be modified in JSON format. This will return “content” if it is a content and “folder” if it is a folder. Post this you move to modifying the name using the respective edit endpoints.
7. A save successful banner is displayed as shown in the design file.

**Secondary Objective (if time permits):**

The second objective of this project is to merely know your level of expertise. This is not compulsory but is nice to have as we believe this would challenge candidates in a few ways.

In this section, you are expected to create a navigation between folders just like on Google Drive. The Content portal given to you as of today retrieves all the content and folder present on the root folder of the database. This uses the endpoint “/view-content-and-folders-sorted" and retrieves all the content and folders in the root directory in a sorted manner.

To create a navigation please achieve the below

1. When the user clicks a folder in the dashboard, the ID of the folder in the database should be passed to the same endpoint as folder\_id to “**/view-content-and-folders-sorted**" along with the viewer\_id which is already being passed.
2. What this does is, the endpoint retrieves all the content and folders under the passed folder\_id.
3. You then display the result on the dashboard.
4. The logic to remember here is that the endpoint **“/view-content-and-folders-sorted**" when no folder\_id is passed by default retrieves the root folder, but when a folder\_id is passed it retrieves the content and folders under that folder.

**Material Provided:**

1. Instruction file
2. Zip file with react code for the content dashboard and backend code for
   1. Edit Content Endpoint (/edit-content/:contentId)
   2. Edit Folder Endpoint (/edit-folder/:folderId)
   3. Get tablelname endpoint (/get-table-name)
   4. Retrieve content and folder end point (/view-content-and-folders-sorted)
3. MySQL DB Schema
4. Design file to create the Rename flow
5. Basic Frontend code with dashboard display

**Prerequisites and Setup Instructions:**

To ensure a smooth development process, please follow these preparatory steps:

1. **MySQL Server Installation:** 
   1. Begin by installing the MySQL server on your local machine. This database management system will serve as the backend for storing and retrieving the repository's data.
2. **MySQL Workbench Installation:** 
   1. Install MySQL Workbench, which is an essential graphical interface tool. It will aid in visually designing, modeling, generating, and managing databases.
3. **Database Schema Import:** 
   1. Import the provided database schema into your MySQL setup. This schema will define the database's structure, including tables, fields, relationships, and other critical elements.
4. **Endpoint Configuration and Testing:** 
   1. Execute the provided endpoint, ensuring that it is properly configured with your local MySQL password. This step is imperative for establishing a successful connection between your dashboard and the database.
   2. Verify that the endpoint is functioning correctly by testing its ability to communicate and exchange data with the MySQL server.
5. **Frontend Configuration:**
   1. Frontend is built on react, with vanilla CSS. Feel free to change your CSS libraries as per your need. (Only if needed by you.)
   2. It might initially throw some errors, install all the dependencies using npm install
   3. App.js is not configured. So go ahead and structure all your components and get the flow done.