**Frontend Documentation: Manager Panel**

I used the react-router-dom library to make navigation and separation of concerns much more clear

**Key Features**

**1. Category Management:**

* **Create/Edit Categories**
  + Enter a name: String.
  + Click "Add Category" or "Update Category".
* **Delete Categories**
  + There is a clickable Delete Button next to the category.

**Note:** Categories with items **CANNOT BE DELETED,** for it is coded like this in the backedn

**2. Item Management**

* **Create/Edit Items**
  + **Name** : String
  + **Description :** String
  + **Mobile Number** (Must be valid by the microservice API)
  + **Category** (MUST)
  + Don’t add validaiton here, let the backend take care of it, so we don’t get bugs for rate limits code 106
* **Delete Items**
  + There is a clickable Delete Button next to the category.

**Types:**

Using TypeScript features for types , I coded:

Id to be a number

Name to be a string

Description to be a string

Mobile Number to be a string

**Component Breakdown**

1. **Category Components:**
2. **CreateCategory**

* **Purpose:** Creates a category using the api service.  
  Just input the name, id generation is handled by spring boot backend

1. **CategoryTable**

* **Purpose:** Displays all categories in a table.
* **Features:**
  + Clickable button for viewing a single category
  + Clickable button for opening EditCategory page in order to edit.
  + Clickable button for deleting the displayed category next to it.

1. **EditCategory**

* **Purpose:** Displays the form in which the user can edit the category

1. **ViewCategory**

* **Purpose:** Viewing the details of a single category

1. **Item Components:**
2. **CreateItem**

* **Purpose:** Creates an item using the api service.  
  Just input the required fields(category is necessary), id generation is handled by spring boot backend  
  Do not use mobile validation here, keep this task for the backend, in order to avoid the rate limit code error 106

1. **CategoryItem**

* **Purpose:** Displays all items in a table.
* **Features:**
  + Clickable button for viewing a single item
  + Clickable button for opening EditItem page in order to edit.
  + Clickable button for deleting the displayed item next to it

1. **EditItem**

* **Purpose:** Displaysn the form in which the user can edit the item

1. **ViewItem**

* **Purpose:** Viewing the details of a single item

1. **Dashboard Component:**

Default home page for our applicatiobn  
It displays the CategoryTable and the ItemTable on one page.

***API Services***

**CategoryService**

Handles API calls for categories: (BASIC CRUD METHODS)

* getAllCategories() : gets all categories.
* createCategory() : Adds a new category.
* updateCategory() : Edits a category.
* deleteCategory() : Removes a category.

**ItemService**

Handles API calls for items:

* getAllItems() : gets all itwms.
* createItem() : Adds a new item.
* updateItem() : Edits an item.
* deleteItem() : Removes an item.
* validateMobileNumber() : Checks if a phone number is valid.

**BACKEND APIs:**

* **Categories API:** http://localhost:8080/api/categories
* **Items API:** http://localhost:8080/api/items
* **Mobile Phone Validation API:** http://localhost:8081/api/mobile/validate