**Backend API Documentation:**

The backend code consists of an Express server connected to a MongoDB database with Mongoose as an ORM. It exposes various RESTful API endpoints for interacting with items and categories.

**Requirements**

* Node.js
* MongoDB
* npm

**Installation & Running the server**

* Clone the repository.
* Open the terminal in the project directory and install the required packages with npm install.
* Run the server with npm start(or by running the items.js file).
* The server will run on <http://localhost:4000>.

**API Endpoints**

All requests and responses are in JSON format.

* POST /categories - Creates a new category.
  + Request body: {“name": "<category name>"}
  + Response: {“\_id": "<id>", "name": "<category name>", "\_v": 0}
  + Error: {“message": "An error occurred"}
* GET /categories - Retrieves all categories. Response: Array of category objects.
* DELETE /categories/:id - Deletes a category with the specified id. Response: "Category deleted"
* POST /items - Creates a new item.
  + Request body: {“name": "<item name>", "category": "<category id>", "description": "<description>", "phone\_number": "<phone number>"}
  + Response: Item object.
  + Error: {“message": "Invalid Category”} or {"message": "Invalid Number"}
* GET /items - Retrieves all items. Response: Array of item objects.
* DELETE /items/:id - Deletes an item with the specified id. Response: "Item deleted"
* PUT /items/:id - Updates an item with the specified id.
  + Request body: {"name": "<item name>", "description": "<description>", "phone\_number": "<phone number>", "category": "<category id>"}
  + Response: "Item updated"
  + Error: {"message": "Category does not exist"} or {"message": "Invalid Number"}

**Frontend Documentation**

The frontend code is built using React and interacts with the backend via the provided API. It allows the user to add, view, and delete items and categories.

**Requirements**

* Node.js
* npm

**Installation & Running the application**

* Clone the repository.
* Open the terminal in the project directory and install the required packages with npm install.
* Run the application with npm start.
* The application will run on <http://localhost:3000>.

**Usage**

* The user can add a new item by clicking the "Add item" button and filling in the details in the popup form.
* The user can add a new category by clicking the "Add category" button and filling in the details in the popup form.
* The user can view all items and their details on the main page.
* Each item can be updated or deleted by clicking the respective buttons.

**Containerization:**

An attempt to containerize the project was made using docker, but both repos had to be merged in a third repo in order to containerize both of them. In the root directory where the backedn and frontend and nginx are, run the following:

**Docker-compose up –build**

Wait a bit… then visit http://localhost:5000