Problem

We want to implement a simple full stack application using React, Python and PostgresQL

Part 1: Front End

- Load in the frontend a static JSON file containing the following:
  ```
  [{ "type": "bank-draft", "title": "Bank Draft", "position": 0
  }, { "type": "bill-of-lading", "title": "Bill of Lading",
  "position": 1 }, {"type": "invoice", "title": "Invoice",
  "position": 2}, {"type": "bank-draft-2", "title": "Bank Draft
  2", "position": 3}, {"type": "bill-of-lading-2", "title":
  "Bill of Lading 2", "position": 4}]
  ```
- Display the content as 5 cards in a 3x2 grid. Assign a different thumbnail of your choice to each document type.
- Display a placeholder spinner for each image that is loading.
- Make the application so the cards can be reordered via drag and drop.
- Make so clicking on a card displays the image as an overlay in the middle of the webpage. Make so pressing ESC closes the image.

Add a README file to explain how to run it.

Here's an example of how a row in the grid may look like:

Part 2: Back End

- Create a `PostgreSQL` (or similar) table that can hold the data that was in the static json file from part 1 in a sensible way
- Build a REST API that can fetch the data from this table and add data to this table. Preferably, use `Python >= 3.6` along with `starlette`.

Part 3: Tying it up!

- Call the API from your front end application to display the same grid.
  - Also feel free to allow any domains and ports for CORS. (Do not waste your time on this)
- Create a button on your application that updates the data in the `PostgresQL` table and amends the grid accordingly. Feel free to hard code the change.

Part 4: Deployment

- Create a docker-compose file to start all the components as microservices
- Write some simple documentation that makes it easy for us to understand and use it