Initial Brainstorming for personal purposes, not documentation

Overview:

Make a full-stack app called Teebay. A marketplace for buying, selling, and renting products.

Tech-Stack:

- React.js
    - Material UI
- Express.js
    - GraphQL
    - Prisma ORM
- PostgresQL

Requirement Breakdown:

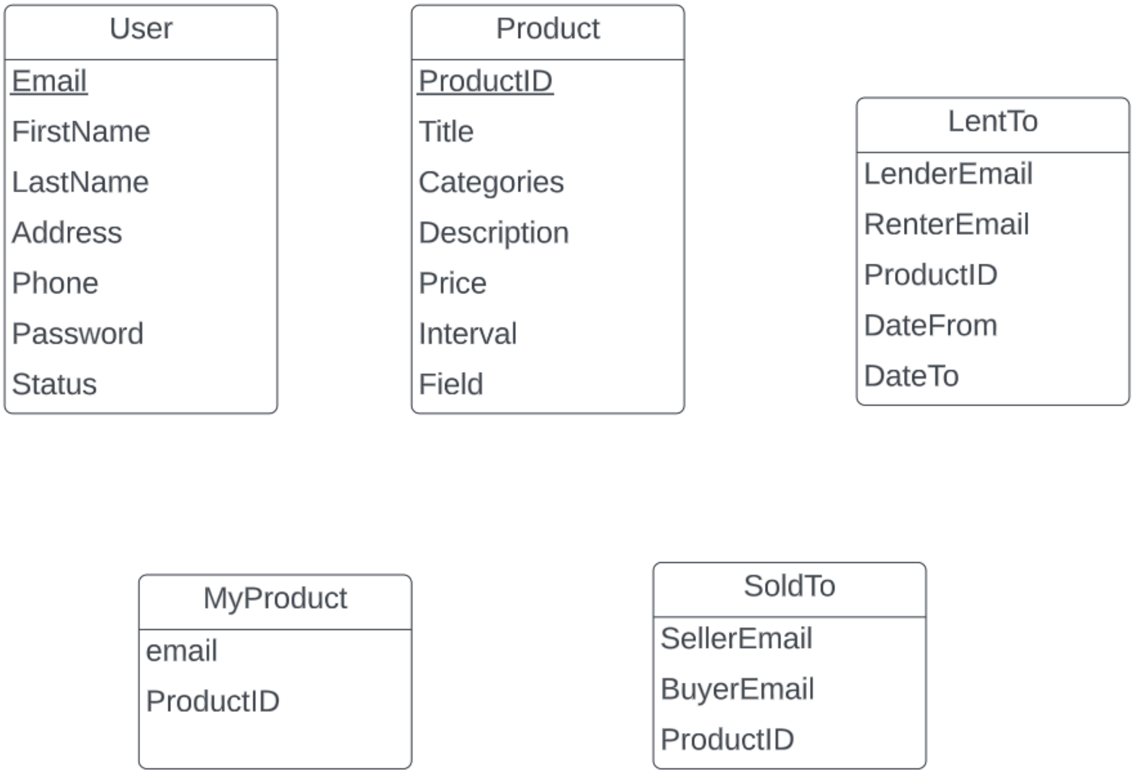
- FE routing
- Error cases
- Readability and comments

sl.	Feature List	Possible Execution
1	User Registration	Simple React Hook form on the FE: 1. Input: First Name, Last Name, Address, Email, phone, Password, Confirm Password (can think of more later) In the BE: 1. Check if primary key (email) exists 2. If so, send an error message 3. If not, hash password and create a new tuple in the user table a. Set user as deactivated b. Send email auth JWT c. Tell user in FE to check their email 4. Send a completion message
2	Login	FE: 1. Input: email and password 2. If passed, save JWT in client to remember login BE: 1. Presence check, Look up check 2. Validate 3. If valid, send a JWT to user

3.	Add product	FE: 1. When button is pressed 2. Open a multi-page forum. Pages: a. Title, check presence and length b. Select categories, multiple can be selected c. Description in a text box d. Enter purchase price e. Enter rent price and rent interval f. Show a summary BE: 1. Send form data in JSON format to BE 2. Clean and check all data a. Length check b. Prescence check c. Range check 3. If it passes check, add as new table to product table 4. If not send an error message indicating which aspect of the code went wrong
4	Delete product	FE: 1. Trash can icon 2. Show a dialog with a confirmation message 3. If no, do nothing and go back 4. If yes, send product ID to backend BE: 1. Delete from database 2. Delete from cache
5	View MY product	FE: 1. Show everything in a new page a. Title b. Categories c. Description d. Price e. Rent f. Option g. Button to edit
6	Edit My product	FE: 1. Edit in all the same places BE: 1. Update the tuple and cache
6	List ALL products	FE: 1. A list of products in the DB BE: 1. Fetch all the products from DB

7	View All products	FE: 1. Similar to viewing MY product but without edit 2. Rent and buy option
8	Rent product	FE: 1. Two inputs, from and to 2. Check of To>From 3. Confirm Rent submits the form BE: 1. Check if the dates match 2. If so, assign the product to user in rent table
9	Buy product	FE: 1. Buy yes and no BE: 1. If bought, assign item to buyer 2. Set item as bought
10	Check stock	FE: 1. Four tabs where different fetchs are made a. Bought b. Sold c. Borrowed d. Lent

Rough Database Design



User
Email
FirstName
LastName
Address
Phone
Password
Status

Product
ProductID
Title
Categories
Description
Price
Interval
Field

LentTo
LenderEmail
RenterEmail
ProductID
DateFrom
DateTo

MyProduct
email
ProductID

SoldTo
SellerEmail
BuyerEmail
ProductID