Please modify my code in the **<!-- Purchase Form Modal -->** section. I have a **Place Order** button. When the user clicks this button:

1. The user should be redirected to the **Order Page**.
2. On the **Order Page**, display the following information:
   * Book cover image
   * Seller name
   * Order ID
   * Total price
   * Date of purchase
   * Payment method
   * MAC address
   * Status (initially set to **"Pending"**)
3. Include two buttons on the Order Page:
   * **Update MAC**: Allows the user to enter a new MAC address, which updates the orders database table.
   * **Cancel Order**: Deletes the order from the database, changes the corresponding book's status in the **New Arrivals Section** back to **"Available"**, and removes the red notification badge.
4. Once the **Place Order** button is clicked:
   * The book's status in the **New Arrivals Section** changes from **"Available"** (green) to **"Sold"** (red), making it non-clickable.
   * The **Orders** navigation button should display a red notification badge with the number "1" to indicate an active order.
   * Clicking the **Orders** button should redirect the user to the **Order Page**.

Here are the files. I will also provide you with my database schema. You don't need to rewrite the code—just tell me where to update for my request. Also, let me know if there are any missing files that I need to provide. The orders.html, orders.css, and orders.js files have not been created yet.

**PART 1**  
  
Create Me a simple websites Project that use Angular.js and Node.js and PostgreSQL.

This is my database used: postgresql://postgres:postgres@localhost:5432/utmebookdb

The Files and structure should only follow my Project Structure (Simple)

First Create Me Register And Login Page By using the data below.  
  username

  email

  password   
  
In Register Page It have OTP services function, After user put username and email, it will send a OTP code to the user’s email and when OTP code entered is correct then user can input the password. After that it will go to the login page. After user login it will go to the dashboard page and in this page will have (Add eBook, Cart, Order, Messages, MyBook, Account) Navigation Bar. And below the Navigation Bar section have Search section, and below search section have Banner Section, and below Banner Section have New Arrivals Section and below New Arrivals have Wishlist section. For The Dashboard page Just Create the interface first do not make it function yet.

**Part 2**

Just do what I request only (Don’t make to complex I will also provide you my database scheme and Project Structure). When clicking on the <i class="fas fa-plus-circle"></i> **Add eBook** button, it will navigate to the **Add Book** page, where users can upload their books. The required information to be entered includes:

* **Title**
* **Category**
* **Price** (in RM)
* **Description**
* **Cover Image** (user can upload an image)
* **Upload Date** (automatically set to the current date), Tells me what folder to create and where
* **Status** (initially set to "Available")
* **Upload Book** (user can upload a PDF file) Tells me what folder to create and where

This information will be stored in the appropriate database table with the correct foreign key relationships.

Once the book is added, two sections will be updated:

1. **New Arrivals Section**
   * This section will display:
     + Cover Image
     + Title
     + Seller Name
     + Status
     + Upload Date
   * Please remove the rating element.
   * The newest book will be shown first.
2. **Order Page**
   * The page will display two sections side by side: **My Purchases** and **My Sales**.
   * When a user adds a book, the **My Sales** section will be updated to show all books uploaded by that user.
   * Each listed book will include an **Edit** button (to modify book information) and a **Remove** button (to delete the book, which will also remove it from the database).

This is the files that currently I have.

**Part 2 Continue…**

In order.html, I have this button:

html

CopyEdit

<button class="btn-edit" ng-click="editBook(book)">

<i class="fas fa-edit"></i> Edit

</button>

When I click this button, I want it to display a form (or modal) that allows me to update the book details, including:

* **Title**
* **Category**
* **Price** (in RM)
* **Description**
* **Cover Image** (user can upload an image)
* **Upload Book** (user can upload a PDF file)

However, the **status** field should be **visible but not editable**.

This method should work the same as the **Add eBook** form, but this one when users want to **edit** an existing book instead of adding a new one.  
  
Don’t make to complex, just do what I request only. I will also provide you my project structure and database scheme.

**Part 3**

I want the following behavior when the **"Buy"** button is clicked on dashboard.html:

html

CopyEdit

<button class="btn-add-buy" ng-click="">

<i class="fas fa-shopping-cart"></i> Buy

</button>

* When this button is clicked, it should navigate to the **"place-order"** page.
* On the "place-order" page, it should display the **book details**, including:
  + Book cover image
  + Title
  + Seller name
  + Price
  + Upload date
  + Book description
* Ask for **payment method**.
* There should also be a **field for the user to enter the MAC address**.

Below the MAC address field:

* Add **two buttons side by side**:
  + **Desktop Instruction**
  + **Mobile Instruction**
* When either button is clicked, a **modal box** should appear showing **instructions** on how to get the MAC address.
* Below the instructions, display a **Download** button:
  + If **Desktop Instruction** is clicked, show a button labeled **"Download Desktop MacReader"**, which downloads a .exe file.
  + If **Mobile Instruction** is clicked, show a button labeled **"Download Mobile MacReader"**, which downloads an .apk file.
* The files should be downloaded from the path: uploads/apps/.

**Lastly**, at the bottom of the page, add a **"Place Order"** button. *(It does not need to be functional yet.). Don’t make to complex, just do what I request only. I will also provide you my project structure and database scheme.*

**Part 3 (Continue…)**

When I click the "Place Order" button in place-order.html:

html

<!-- Place Order Button -->

<div class="order-actions">

<button class="btn-place-order" ng-click="placeOrder()">

Place Order

</button>

</div>

The following actions should take place:

1. The information about the purchased book will be saved to the purchases database table.  
   The stored information should include:
   * title
   * category
   * OrderID ( MUST 10 characters start with UTM)
   * price
   * payment\_method
   * mac\_address
   * cover\_image\_path
   * purchase\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
   * OrderStatus VARCHAR(20) DEFAULT 'Pending'
   * book\_file\_path

Ensure all foreign key relationships are correctly defined.

1. The book status in the books table will be updated from "Available" to "Sold".
2. The "My Purchases" section in order.html will be updated to display the purchased book details similarly to the "My Sales" section.
   * The status will reflect the value from the OrderStatus column in the purchases table.
   * The "Remove" button will be replaced with a "Cancel Order" button.
3. When the user clicks the "Cancel Order" button:
   * The order will be removed from the "My Purchases" section.
   * The book status in the books table will be changed back to "Available".
4. When the user clicks the "Edit" button:
   * They should only be able to update the MAC address.
5. A message will be sent to the admin user in message.html through the following navigation item:

html

<li>

<a href="#" ng-click="navigateTo('messages')">

<i class="fas fa-envelope"></i> Messages

</a>

</li>

* The message will contain details of the purchase based on purchases database table.
* It should be sent only to users with the admin role, based on the users database table.

1. When an admin user clicks the "Messages" tab:

html

<li>

<a href="#" ng-click="setActiveTab('messages')">

<i class="fas fa-envelope"></i> Messages

</a>

</li>

* It will display all purchases made by users.
* When the admin clicks on a specific purchase, the purchase details will be shown.