EZeats Fullstack App – Menu & Checkout Flow

Objective:

Build a simple fullstack web application that allows users to browse a restaurant menu, select items, manage a cart, and proceed to checkout via Paymob test integration.

Guidelines:

- Use **React (with TypeScript)** and **Tailwind CSS** for the frontend.
- Use **Node.js with Express** (or any lightweight backend framework) for the backend.
- Paymob integration should work with a **test account**.
- Write **original code**—avoid boilerplate dumping or AI-generated code copy-pasting.
- Prioritize **clean architecture**, **simple UX**, and **modularity**.

Frontend Requirements:

Based on this Figma file:
📎 Figma Link
(https://www.figma.com/design/DwxRKFarNBF2yS3yKLOTCF/Full-Stack-Task?node-id=0-1&p=f&t=LXXLuu55F3AxReAV-0)

1. **Login Screen**
   - Users log in via **email + password**.
   - Token should persist across refreshes using **localStorage** or cookies.
2. **Menu Listing Screen**
   - Fetch and display menu items from the backend.
   - Group items by category if applicable.

   - Each item has image, name, discripotion and price.

3. **Item Details Screen**
   - On selecting a menu item, show a detailed view.
   - User can specify quantity and add to cart.
4. **Menu with Cart Floating Button**
   - After adding to cart, return to menu.
   - A floating button shows **"View Cart"** with total item count and price.
5. **Checkout Screen**
   - Clicking "View Cart" goes to checkout.
   - Show summary of items, total cost.
   - Integrate Paymob test payment API.
   - On successful payment, show confirmation screen.

Backend Requirements:

1. **Authentication API**
   - Login endpoint with JWT issuance.
   - Protect all menu/cart/checkout routes with middleware.
2. **Menu API**
   - Endpoint to get list of menu items.
3. **Cart Logic (Session or In-Memory)**
   - Add/Remove/Update items to cart per user session.

4. **Checkout API**
   - Create an order object with selected items.
   - Integrate **Paymob test checkout API**.
   - Simulate payment confirmation and return success/failure response.

5. **Paymob Setup Instructions**

- Register a test account here: https://accept.paymob.com/portal2/en/register
- Follow this official guide to integrate: https://developers.paymob.com/egypt/getting-started-egypt

---

Submission:

- GitHub repo with clear README
- Include setup steps, `.env.example`, and screenshots
- Deploy frontend and backend (optional but a plus)
- Separate folders for `frontend/` and `backend/`

Evaluation Criteria:

- Code readability and structure
- Frontend & backend integration quality
- UI/UX quality
- Error handling & validation

- Authentication flow
- Clean API design
- Paymob integration working as expected
