EXERCISE: FOOD DELIVERY

1. DESCRIPTION

Build a backend API for a food delivery platform with the following capabilities:

- Restaurants register and add menu items.
- Customers register and place orders.
- Orders contain multiple menu items and belong to a single restaurant.
- Each restaurant has a daily order limit.
- Orders go through a status workflow (PENDING, CONFIRMED, DELIVERED, CANCELLED).

2. FUNCTIONAL REQUIREMENTS

Create Restaurants - Each restaurant has a name and a daily order limit.

Add Menu Items - Each menu item belongs to a restaurant and has a name and price.

Create Customers - Customers register with name and email.

Place Orders - A customer places an order to a single restaurant with one or more menu items and quantities.

- The total price is the sum of the price × quantity for each item.
- The system must enforce the restaurant’s daily order limit (only count orders made that day with status PENDING or CONFIRMED).
- An order cannot be placed if the limit is reached.

Update Order Status — Orders can be updated following this flow:

- PENDING → CONFIRMED
- CONFIRMED → DELIVERED or CANCELLED

- Once DELIVERED or CANCELLED, no further updates allowed.

Get Order Details — Retrieve a complete order, including customer info, items ordered with quantities, and total price.

3. API ENDPOINTS

POST (/restaurants) - Create a restaurant

POST (/restaurants/:id/menu-items) - Add menu items to a restaurant

POST (/customers) - Register a new customer

POST (/orders) - Place an order (with items)

PATCH (/orders/:id/status) - Update the status of an order

GET (/orders/:id) - Retrieve order details

4. DELIVERABLES

- Source code with instructions to run.
- Postman or curl commands to test the endpoints.

5. BONUS FEATURES

- Input validation and error handling.
- Transaction handling for order creation and status updates.
- Unit tests for key business logic (order creation, limit enforcement).
- Basic authentication (optional).