Task Overview

Build a “Smart Delivery Fee Calculator” Web App

You need to create a small MERN application that calculates **dynamic delivery fees** based on order weight, distance, and delivery type.

Requirements

1. Backend (Node + Express + MongoDB)

- Create an Express server with the following endpoints:

POST /api/calculate-fee

Accepts payload:

```
{
  "weight": 12.5,
  "distance": 7,
  "deliveryType": "express"
}
```

Business Rules for Fee Calculation:

- Base Fee = ₹50
- Weight Charge:
    - Up to 5kg → ₹10 per kg
    - 5–20kg → ₹8 per kg
    - Above 20kg → ₹5 per kg
- Distance Charge:
    - Up to 5km → ₹5 per km
    - 6–20km → ₹3 per km
    - Above 20km → ₹2 per km
- Delivery Type Multiplier:

- standard → x1
- express → x1.5
- priority → x2

Formula:
totalFee = (baseFee + weightCharge + distanceCharge) * multiplier

Return calculated total as:

```
{
 "totalFee": 235
}
```

GET /api/history

- Return last 10 fee calculations stored in MongoDB (with timestamp).

---

2. Frontend (React)

- Create a simple UI with:
  - Input fields: Weight (kg), Distance (km), Delivery Type (dropdown)
  - A **Calculate Fee** button.
  - Display calculated fee clearly on the page.
- Below, show a table of **last 10 calculations** fetched from /api/history.

---

3. Data Storage

- Store every calculation in MongoDB with:
  - weight, distance, deliveryType, totalFee, timestamp.

---

4. Edge Cases

- Validate inputs (no negative weight/distance).

- Round totalFee to 2 decimals.
- Handle case where DB is empty (show "No history yet").

---

5. Code Quality

- Use **modular structure**: separate routes, controllers, services.
- Avoid hardcoding magic numbers — put rates in a config file.
- Use async/await, proper error handling.
- Clean React UI with functional components + hooks.