Test Task

**💼 Full Stack Developer Test Task**

**Project:** Create Business Confirmation Flow  
**Company:** Open Mineral  
**Tech Stack:** React + Django (REST API)

**🎯 Objective**

Create **Business Confirmation (BC) Flow** application from wireframes. Traders use this tool to input and validate critical deal information before contract execution.  
The main goal - as full stack developer, build a **multi-step form** application, with stand alone UI and backend. Add suggestions/alerts as part of a mock AI feedback module. All values for drop downs should be stored as dictionary in data base and come from backend,  
  
  
doc file in attachment:   
  
Figma: <https://www.figma.com/deck/eprWs3iqdWbhiXdkhYx9cB/Open-Mineral-%E2%80%93-Business-Confirmation-UX-Flow-by-ROMAN-VOROKHIB-%C2%A9-2025?node-id=1-443&t=olpMFMytewN0VIpJ-1>

**🧩 Architecture Overview**

| Layer | Stack |
| --- | --- |
| Frontend | React |
| Backend | Django |
| Database | SQLite or PostgreSQL |
| Optional | Docker, GitHub Actions |

🟢 Step 1: Deal Basics

| Section | Fields |
| --- | --- |
| Seller & Buyer | Static prefilled seller; dynamic buyer |
| Material & Quantity | Dropdown + Numeric input (±%) |
| Navigation | Button: “Proceed to Terms” |

**Logic to Implement:**

* Field validations: Quantity must be a number, Material required.

**🟡 Step 2: Commercial Terms**

| Section | Fields |
| --- | --- |
| Delivery & Shipment | Delivery Term, Point, Mode, Shipment Period, Packaging |
| Assay / Quality | File upload (mocked), Autofill Assay fields (Zn, Pb, Ag) |
| Pricing & Charges | TC (USD/dmt), RC Ag (USD/toz) with AI Suggestions |
| Additional Terms | Clauses: Transportation Credit, Other Payables, etc. |

📌 **Logic to Implement:**

* Show mock AI feedback like:  
  “Your RC is higher than market avg. Suggest lowering to $4.50”
* If Rails - show “Bulk” and “Big Bags” if Ship - show only “Bulk”
* Allow clause toggles or editable templates.

**🟣 Step 3: Payment Terms**

| Section | Fields |
| --- | --- |
| Payment Method | Method, Currency, Triggering Event |
| Payment Stages | Slider for Prepayment %, text for Provisional & Final |
| WSMD & Surveyor | Final location, cost sharing (slider or fixed 50/50), nominated parties |

📌 **Logic to Implement:**

* Logic pre-fills clause texts with industry standards (editable).
* Dropdowns for surveyor nomination.

**🔵 Step 4: Review & Submit**

| Section | Display Style |
| --- | --- |
| Review Summary | Grouped fields, readonly blocks |
| Final Confirmation | Submit button, Back/Edit link |
| AI Validation | Suggestions like “Surveyor not selected” |

📌 **Logic to Implement:**

* Clear call to action.
* Mock suggestion popup:  
  e.g., “⚠️ Treatment Charge 20% higher than last month. Proceed?”

**Step 5: Summary Page with Graph**   
Implement you own vision.   
  
  
  
**Additional Requirement (Final Step Submit Processing)**  
At the final step of the Business Confirmation Flow, there must be a **Submit** button.  
When the user clicks **Submit**, the application must:

* **Trigger a background task in a Celery queue (using Redis as the broker).**
* This task should simulate processing by **sleeping for 15 seconds**.
* After the task completes, it must mark itself as **completed** in the database.
* The frontend should poll or fetch the task status and **display a message confirming successful processing completion.**
* The task must **actually** execute in the Celery worker and Redis queue (not just a mocked delay on the frontend).

**Deliverables:**

* Public GitHub repositories with:
  + The complete codebase.
  + A fully working **Docker Compose configuration** (including Django, React App, Redis, and Celery).

A clear **README.md** file with step-by-step setup and run instructions.  
  
  
NOTE: For fields that offer multiple options, feel free to use placeholder text like "lorem ipsum" or any sample data of your choice.