Current front end: https://project-17d1.vercel.app/

FREE TO SUGGEST BACKEND AND DB OPTIONS

BUDGET: $200

Here's a structured breakdown of how the backend of the B2B tender matchmaking platform can function, based on your requirements:

Backend Functional Overview

The backend will serve as the engine to manage users, roles, listings, communication, and transactions. It will also ensure data security, role-based access, and seamless communication between buyers and suppliers.

1. User Management

1.1 Signup/Onboarding
• Separate portals for Buyers and Suppliers.
• Buyers: Input requirements such as company name, industry, contact details, and verification info.
• Suppliers: Input requirements such as license numbers, product categories, available stock details, etc.
• Email Verification: Use Resend or SendGrid for confirmation emails during signup.

1.2 Authentication
• Features: Login, Logout, Password Reset.
• Role-based Authentication: Define user roles (Buyer, Supplier, Admin). • JWT-based token system for session management.
• Forgot Password: Trigger email via API.

1.3 Account Approval
• New accounts will require admin approval (optional). • Automatic email notifications post approval/rejection.

2. Role-Based Access Control
• Buyers:
• Access to search for suppliers and their listings.
• Ability to post "Requests for Tenders (RFT)".
• View incoming bids and communicate with suppliers. • Suppliers:
• Access to search for buyer requests.
• Ability to post product/service listings.
• Submit bids for tenders and message buyers.

3. Tender and Listing Management

3.1 Buyer-Side Features
• Post a Tender:
• Define product/service required, quantity, budget range, and delivery time. • Search Suppliers:
• Filter by product category, availability, location, and price.

3.2 Supplier-Side Features
• Create Listings:
• Include product categories, stock levels, pricing, and delivery terms. • View Buyer Requests:
• Search/filter buyer posts based on industry, quantity, and price point.

3.3 Search & Matchmaking
• Implement search algorithms for buyers and suppliers.
• Matching criteria:
• For Buyers: Supplier profiles and listings relevant to their tender. • For Suppliers: Buyer requests that align with their offerings.

4. Communication System

Messaging Tab
• 1:1 Messaging:
• Buyers and suppliers can exchange messages.
• Notifications for new messages via email API.
• Message History:
• Store and display chat history per tender/interaction.

5. Contract & Transaction Management

Contract Creation
• After a successful bid, buyers and suppliers can finalize the terms. • Allow both parties to view contract details.

Transaction Records
• Record all bids, agreements, and completed tenders.
• Optional integration with payment gateways for escrow-like functionality.

6. Admin Panel
• User Management:
• Approve/reject user accounts.
• Tender and Listing Monitoring:
• Oversee all posts for compliance.
• Data Analytics:
• Track activity (tenders posted, bids submitted, etc.).

7. Notifications
• Email Notifications (via Resend/SendGrid):
• Account creation confirmation.
• Updates on tender statuses (e.g., new bid, contract agreement). • Messaging alerts.
• Platform Notifications:
• Dashboard alerts for new interactions.

8. Security Features
• Data Encryption: Encrypt sensitive data such as license numbers.
• Role Validation: Ensure users access only their respective features.
• Rate Limiting: Prevent abuse of search, messaging, and tender posting.

9. Integration with Frontend
• The backend APIs will align with the Figma designs:
• RESTful APIs: For all platform functionalities (signup, search, messaging, etc.). • WebSockets: For real-time messaging.

Tender Matching algorithm:

Inputs Recorded from Listing and Catalogue:

1. Buyer Requirements:
• max_cost: Maximum acceptable cost.
• max_lead_time: Maximum acceptable lead time.
• region: Region(s) where the buyer operates.
• payment_terms: Preferred payment terms.
• warranty_required: Whether the buyer requires a warranty (1 for "Yes", 0 for "No").
2. Supplier Offerings:
• cost: Cost offered by the supplier.
• lead_time: Lead time offered by the supplier.
• region: Region(s) where the supplier operates.
• rebate: Rebate offered by the supplier (1 for "Yes", 0 for "No").
• payment_terms: Payment terms offered by the supplier.
• warranty_offered: Whether the supplier offers a warranty (1 for "Yes", 0 for "No").
3. Weights:
• w1, w2, w3, w4, w5, w6: Weights assigned to cost, lead time, region compatibility, rebates, payment terms, and warranty compatibility, respectively.
• (Sum of weights = 1.)

Steps:
1. Initialise Weights:
Assign normalised weights for each parameter, e.g., w1 = 0.35, w2 = 0.2, w3 = 0.2, w4 = 0.1, w5 = 0.1, w6 = 0.05.
2. Compute Parameter Scores:
For each buyer-supplier-product combination, calculate the following scores:
• Cost Score (C):

- If supplier_cost ≤ max_cost: C = 1.
- If supplier_cost > max_cost:

C = 1 - ((supplier_cost - max_cost) / max_cost).

- Lead Time Score (L):
- If supplier_lead_time ≤ max_lead_time: L = 1.
- If supplier_lead_time > max_lead_time:

L = 1 - ((supplier_lead_time - max_lead_time) / max_lead_time).

- Region Compatibility (R):
- If supplier_region == buyer_region: R = 1.
- Otherwise: R = 0.
- Rebate Score (Reb):
- If supplier_rebate == 1: Reb = 1.
- Otherwise: Reb = 0.
- Payment Terms Score (PT):
- If supplier_payment_terms == buyer_payment_terms: PT = 1.
- Otherwise: PT = 0.
- Warranty Score (W):
- If buyer_warranty_required == supplier_warranty_offered: W = 1.
- Otherwise: W = 0.

3. Combine Scores into Weighted Score:

Weighted Score = w1 * C + w2 * L + w3 * R + w4 * Reb + w5 * PT + w6 * W.

4. Convert to Percentage:

Match Percentage = 100 * Weighted Score.

5. Output:

Generate a matrix of match scores (0–100%) for all buyer-supplier-product combinations.

Pseudocode:

```
# Inputs: Buyers, Suppliers, Products, Weights
buyers = [...]  # Buyer preferences (cost, lead time, region, payment terms, warranty)
suppliers = [...]  # Supplier offerings (cost, lead time, region, rebates, payment terms, warranty)
products = [...]  # Products being matched
weights = [w1, w2, w3, w4, w5, w6]  # Normalised weights

# Function to calculate match score
def calculate_match_score(buyer, supplier, product):
# Cost score
cost_score = 1 if supplier.cost <= buyer.max_cost else max(0, 1 - (supplier.cost -
buyer.max_cost) / buyer.max_cost)

# Lead time score
lead_time_score = 1 if supplier.lead_time <= buyer.max_lead_time else max(0, 1 -
(supplier.lead_time - buyer.max_lead_time) / buyer.max_lead_time)

# Region compatibility
region_score = 1 if supplier.region == buyer.region else 0
```

```
# Rebate score
rebate_score = supplier.rebate  # 1 if rebate is offered, 0 otherwise

# Payment terms score
payment_terms_score = 1 if supplier.payment_terms == buyer.payment_terms else 0

# Warranty score
warranty_score = 1 if buyer.warranty_required == supplier.warranty_offered else 0

# Weighted score
weighted_score = (
weights[0] * cost_score +
weights[1] * lead_time_score +
weights[2] * region_score +
weights[3] * rebate_score +
weights[4] * payment_terms_score +
weights[5] * warranty_score
)

# Match percentage
match_percentage = 100 * weighted_score
return match_percentage

# Generate match matrix
match_matrix = []
for buyer in buyers:
for supplier in suppliers:
for product in products:
match_score = calculate_match_score(buyer, supplier, product)
match_matrix.append((buyer, supplier, product, match_score))

# Output: Match scores for all combinations
print(match_matrix)
```

Low Match Example:

Input Parameters:

1. Buyer Preferences:
   - Max Cost: £100
   - Max Lead Time: 10 days
   - Region: “West Midlands”
   - Payment Terms: “30 days”
   - Warranty Required: Yes

2. Supplier Offerings:
- Cost: £150
- Lead Time: 20 days
- Region: “South East”
- Rebate: No
- Payment Terms: “90 days”
- Warranty Offered: No

3. Weights:
- w1 = 0.35, w2 = 0.2, w3 = 0.2, w4 = 0.1, w5 = 0.1, w6 = 0.05

Calculation:
1. Cost Score (C): 0.5
2. Lead Time Score (L): 0
3. Region Compatibility (R): 0
4. Rebate Score (Reb): 0
5. Payment Terms Score (PT): 0
6. Warranty Score (W): 0

Weighted Score:
(0.35 * 0.5) + (0.2 * 0) + (0.2 * 0) + (0.1 * 0) + (0.1 * 0) + (0.05 * 0) = 0.175

Match Percentage:
100 * 0.175 = 17.5%

Final Match Score: 18%

High Match Example:

Buyer Preferences:
- Max Cost: £100
- Max Lead Time: 10 days
- Region: “West Midlands”
- Payment Terms: “30 days”
- Warranty Required: No

Supplier Offerings:
- Cost: £120 (slightly higher than the buyer’s max cost)
- Lead Time: 10 days (meets the buyer’s max lead time)
- Region: “West Midlands” (matches buyer’s region)
- Rebate: No
- Payment Terms: “30 days” (matches buyer’s preference)
- Warranty Offered: No

Weights:
- w1 = 0.35, w2 = 0.2, w3 = 0.2, w4 = 0.1, w5 = 0.1, w6 = 0.05

Calculation:

1. Cost Score (C): 0.8
2. Lead Time Score (L): 1
3. Region Compatibility (R): 1
4. Rebate Score (Reb): 0
5. Payment Terms Score (PT): 1
6. Warranty Score (W): 1

Weighted Score:
(0.35 * 0.8) + (0.2 * 1) + (0.2 * 1) + (0.1 * 0) + (0.1 * 1) + (0.05 * 1) = 0.895

Match Percentage:
100 * 0.895 = 89.5%

Final Match Score: 90%