1. API Name

Tender Recommendation API

2. API Endpoint and Method

The primary method for retrieving recommendations will be a `GET` request, which is suitable for fetching a ranked list of items.

- **Endpoint:** `/api/v1/recommendations/tenders`
- **Method:** `GET`

3. Request Process

The API will leverage the vector embeddings generated for both bidder profiles [1] and tenders [22] to find the most relevant opportunities for a specific bidder.

1. **Authentication:** The client makes a request to the endpoint, including a user authentication token (e.g., JWT) in the header. The backend uses this token to identify the `auth_user_id`[3].
2. **Fetch Bidder Profile Vector:** The system queries the `users` collection in MongoDB using the `auth_user_id`. It retrieves the `profile_embedding` vector for the authenticated bidder [4]. If the embedding is null or does not exist, the API should return an error or an empty list, indicating that the bidder profile is incomplete.
3. **Vector Similarity Search:** The bidder's `profile_embedding` vector is used as the query input for a vector search on the `tenders` collection. This search is performed against the `vectorEmbedding` field of the tenders[555].
    - The database (e.g., MongoDB Atlas using a vector index [6]) calculates a similarity score (e.g., cosine similarity) between the bidder's vector and every tender's vector.
4. **Pre-filtering:** Any query parameters included in the `GET` request (such as `country`, `state`, or `min_value`) are applied as pre-filters during the vector search. This ensures that the similarity search is only run on the subset of tenders that already meet the bidder's explicit criteria.
5. **Ranking and Pagination:** The search results are returned ranked by their similarity score in descending order. The system then applies pagination to this ranked list based on the

page and limit query parameters[7].

6. **Format and Respond:** The system constructs a JSON response containing the ranked and paginated list of tenders, mirroring the output format of the GlobalTender API[8] but with the addition of a
7. match_score for each tender.

4. GET Request Format

The request uses query parameters for basic filtering and pagination, building upon the parameters available in the

GlobalTender API[9].

Endpoint:

https://api.yourdomain.com/api/v1/recommendations/tenders

Authentication Header:

- Authorization: Bearer <user_jwt_token>

Query Parameters (Optional):

Parameter	Type	Description	Example
country	String	ISO 3166-1 alpha-2 country code to filter tenders[10].	IN
state	String	State or province to filter tenders[11].	Tamil Nadu
min_value	Integer	Minimum tender value to	500000

		consider[12].		
currency	String	ISO 4217 currency code for the	min_value[13].	INR
page	Integer	Page number for pagination. Default: 1[14].	1	
limit	Integer	Number of results per page. Default: 10, Max: 100[15].	20	

Example GET Request:

```
https://api.yourdomain.com/api/v1/recommendations/tenders?country=IN&min_value=10000000&currency=INR&limit=15
```

5. API Output Format

The output format is designed for consistency with the existing

GlobalTender API[16], with the key addition of a

match_score field in each tender object to indicate the relevance of the recommendation.

Output (Success 200 OK):

JSON

```
{
  "status": "success",
  "results": 15,
  "pagination": {
    "currentPage": 1,
    "totalPages": 12,
    "totalResults": 178,
    "limit": 15
  },
  "data": [
    {
      "match_score": 0.925,
      "tenderId": "tend_f4a8b1c9",
      "sourceUrl": "https://example.gov.in/tenders/2025/infra/001",
      "scrapedTimestamp": "2025-08-29T16:30:00Z",
      "country": "IN",
      "state": "Tamil Nadu",
      "tenderDetails": {
        "referenceNumber": "TN/INFRA/2025/001",
        "title": "Construction of Smart City Command Centre",
        "issuingAuthority": "Greater Chennai Corporation",
        "procurementSummary": "This project involves the design, development, and construction of a centralized command centre...",
        "category": ["Infrastructure", "Construction", "IT"],
        "tenderValue": 50000000,
        "currency": "INR",
        "dates": {
          "publishedDate": "2025-08-15T12:00:00Z",
          "closingDate": "2025-09-30T15:00:00Z"
        }
      }
      // ... other tender fields as defined in the Tender Scraping module
    },
    {
      "match_score": 0.891,
      "tenderId": "tend_x7y8z9a0",
      // ... other tender details
    }
  ]
}
```

Output (Error 404 Not Found):

This error would be returned if the user's profile or embedding is not found.

JSON

```
{
  "status": "error",
  "message": "Bidder profile not found or is incomplete. Please complete your profile to receive recommendations."
}
```