Junior Full-Stack Developer Task: AI Job Skill Analyzer

Welcome to the next step in the interview process with Botson.ai. This task is designed to showcase your skills by solving real-world problems. We care deeply about your thought process, problem-solving approach, and creativity more than building a perfect, production-ready application.

Please return within 48 hours of receipt

This exercise mirrors a slice of our real internal dashboard. Build two screens to help an Account Manager analyze and extract insights from job-trading logs, and create an AI-powered chat interface for querying the data.

The data is available at:
https://storage.googleapis.com/feeds-json/feeds/transformedFeeds.json

About Our Data

Every day, we receive approximately 20 million new job postings that must be indexed. Each hour, indexing processes run for each client, and logs are generated for each indexing event. This data represents these indexing logs.

transactionSourceName: Anonymized client name (Deal1, Deal2, etc.)
progress object contains:
- **SWITCH_INDEX**: True if indexing switched to a new ElasticSearch index.
- **TOTAL_RECORDS_IN_FEED**: Items received before filtering.
- **TOTAL_JOBS_IN_FEED**: Items after filtering (min CPC, etc.).
- **TOTAL_JOBS_FAIL_INDEXED**: Failed indexing count.
- **TOTAL_JOBS_SENT_TO_ENRICH**: Jobs sent to LLM for metadata enrichment.
- **TOTAL_JOBS_DONT_HAVE_METADATA**: Jobs lacking metadata after enrichment.
- **TOTAL_JOBS_SENT_TO_INDEX**: Successfully indexed jobs.

Example Data Object

```
{
  "_id": "68709db2402cf56cd3813d9e",
  "country_code": "US",
  "currency_code": "USD",
  "progress": {
    "SWITCH_INDEX": true,
    "TOTAL_RECORDS_IN_FEED": 16493,
    "TOTAL_JOBS_FAIL_INDEXED": 1521,
    "TOTAL_JOBS_IN_FEED": 13705,
    "TOTAL_JOBS_SENT_TO_ENRICH": 20,
    "TOTAL_JOBS_DONT_HAVE_METADATA": 2540,
    "TOTAL_JOBS_DONT_HAVE_METADATA_V2": 2568,
    "TOTAL_JOBS_SENT_TO_INDEX": 13686
  },
  "status": "completed",
  "timestamp": "2025-07-11T05:16:20.626Z",
  "transactionSourceName": "Deal4",
  "noCoordinatesCount": 160,
  "recordCount": 11118,
  "uniqueRefNumberCount": 9253
}
```

Screens to Build

Screen A – Operations Dashboard

1 Present tables and/or charts to quickly spot trends and outliers.

2 Support filtering (date range, client, country), sorting, and pagination.

3 Highlight metrics such as averages or deltas to aid decision-making.

Screen B – AI Chat Assistant

1 Embed a chat box for natural-language questions (e.g., 'Average TOTAL_JOBS_SENT_TO_INDEX per client last month?').

2 Backend interprets the query, queries MongoDB, then responds as text, table, or chart.

3 Use an LLM of your choice (OpenAI, Claude, Gemini, etc.) and show your prompt design.

4 Handle ambiguous or unsupported questions gracefully.

Technical Requirements

1 Frontend: React 18 (state library optional).

2 Backend: Node.js + Express OR Next.js App Router.

3 Database: MongoDB (import the provided JSON).

4 Any chart/UI library is acceptable.

5 Document AI API keys in .env.example (do not commit secrets).

Deliverables

1 Public Git repo or ZIP archive.

2 README with setup steps.

3 PROCESS.md explaining:

4 • Architecture and key design choices.

5 • Your AI prompts and any iterations.

6 • How you used AI tools during the task.

Evaluation Criteria

1 Working functionality.

2 Clear system boundaries (frontend ↔ backend ↔ AI).

3 Effective prompt engineering and error handling for AI responses.

4 Code clarity and organization.

5 Transparent, thoughtful use of AI.

Good luck! We look forward to reviewing your submission.