Project Report: AI-Powered Candidate Screening Backend Service

Candidate Information

- **Full Name**: Rafli Ramadhani
- **Email Address**: raflirama7446@gmail.com

Repository Link

- https://github.com/zendParadox/backend-ai-screener

---

1. Approach & System Design

Initial Plan

I divided this challenge into several main components:

1. **API Layer**: Building a clear RESTful API for external interaction.
2. **Asynchronous Processing**: Identifying that AI evaluations take time, so the process must be separated from the main HTTP request-response flow.
3. **RAG (Retrieval-Augmented Generation)**: Designing a workflow for storing and retrieving reference documents (ground truth) to provide accurate context to the LLM.
4. **LLM Integration**: Connecting the system with a generative AI service to perform the final analysis.

System & Database Design

API Endpoints: The system exposes three main endpoints:

- `POST /upload`: Accepts `multipart/form-data` (CV and Report), stores them on the server, and returns the filenames as unique IDs.
- `POST /evaluate`: Accepts file IDs and job details, then enqueues an evaluation task (queue) and immediately returns a `job_id`.
- `GET /result/{id}`: Allows clients to check the job status (`queued`, `processing`, `completed`, `failed`) and retrieve the final result when available.

Job Queue: To handle long-running evaluation processes, I used **BullMQ with Redis** as the backend. This ensures the `/evaluate` endpoint can respond quickly without waiting for the LLM call to complete. It also lays a strong foundation for scalability and future fault-tolerance.

Vector Database: I chose **Qdrant** as the vector database. Reference documents such as Job Descriptions, Case Study Briefs, and Evaluation Rubrics are ingested into a Qdrant collection. This allows efficient semantic search to retrieve the most relevant context for each evaluation.

LLM Integration

LLM Choice: I used **Google AI's gemini-2.5-flash** model, selected for its balance of speed, ability to follow complex instructions (such as producing strict JSON outputs), and cost efficiency.

RAG Strategy: The Retrieval-Augmented Generation pipeline was implemented as follows:

1. When an evaluation starts, the candidate's CV content is converted into vector embeddings using the local model `Xenova/all-MiniLM-L6-v2`.
2. These embeddings are used to query Qdrant for the most relevant reference documents.
3. The retrieved documents (context) are then injected into the prompt sent to Gemini.

Prompting Strategy: The prompt was carefully designed to ensure consistent and structured output:

1. **Persona**: The AI is instructed to act as an *"Expert HR Assistant."*
2. **Strict Instructions**: Explicitly requires the AI to use contextual documents as the sole source of truth.
3. **Output Structure**: Defines a mandatory JSON schema.
4. **Data Injection**: Context from RAG and candidate CV/Report data are provided separately with clear markers.

---

2. Results & Reflection

Results:
The asynchronous architecture with BullMQ worked very well. The system was able to accept jobs, process them in the background, and deliver results without blocking the client. The RAG pipeline successfully retrieved relevant context, significantly improving Gemini's output quality compared to evaluations without context.

Reflection:
The biggest challenge was ensuring version compatibility across all components (Qdrant server, Node.js client, etc.). Additionally, the quality of embeddings strongly influenced the success of RAG. Using a locally running embedding model provided advantages in privacy and cost, though it required additional initialization time during the first run.

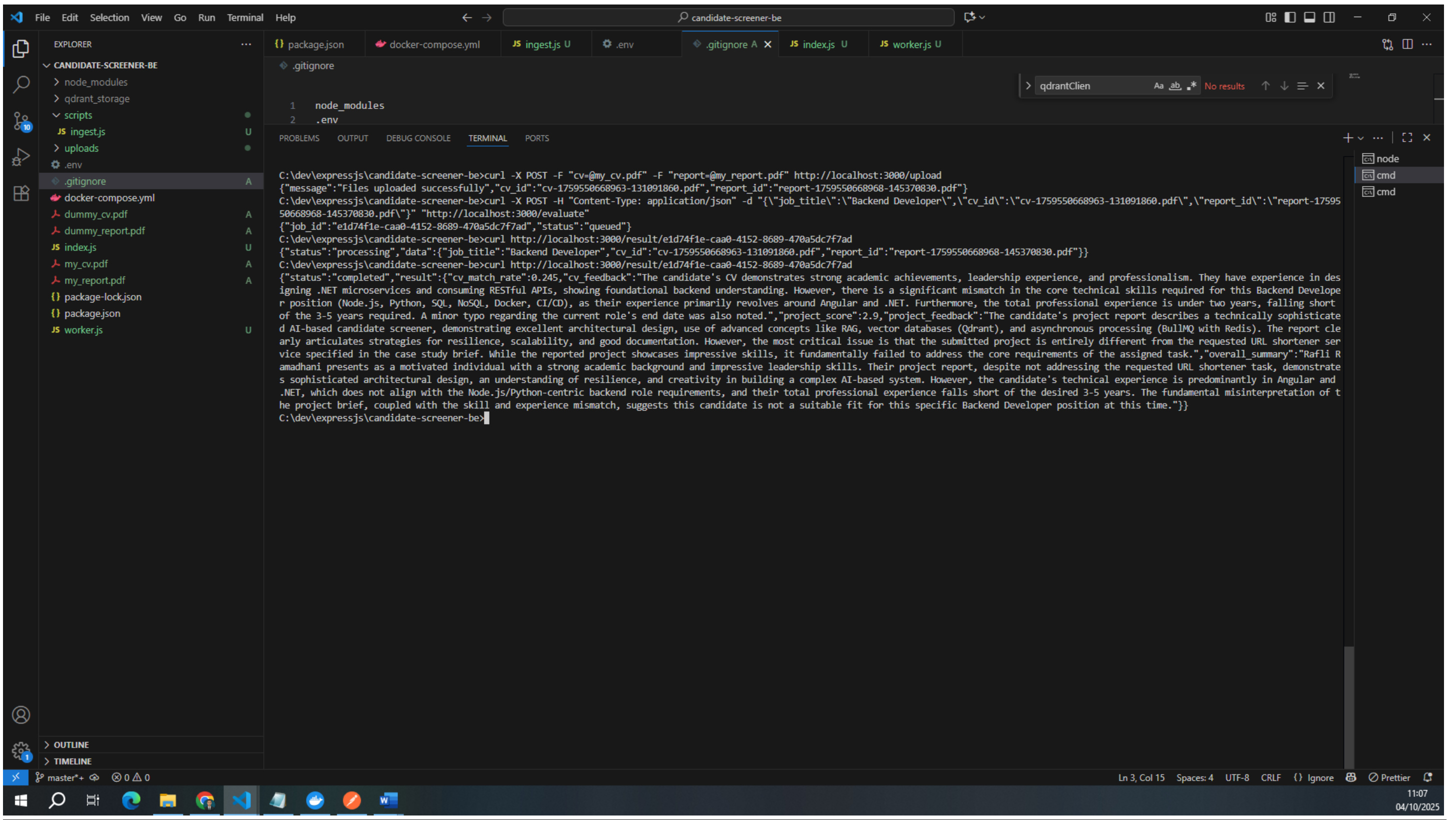
---

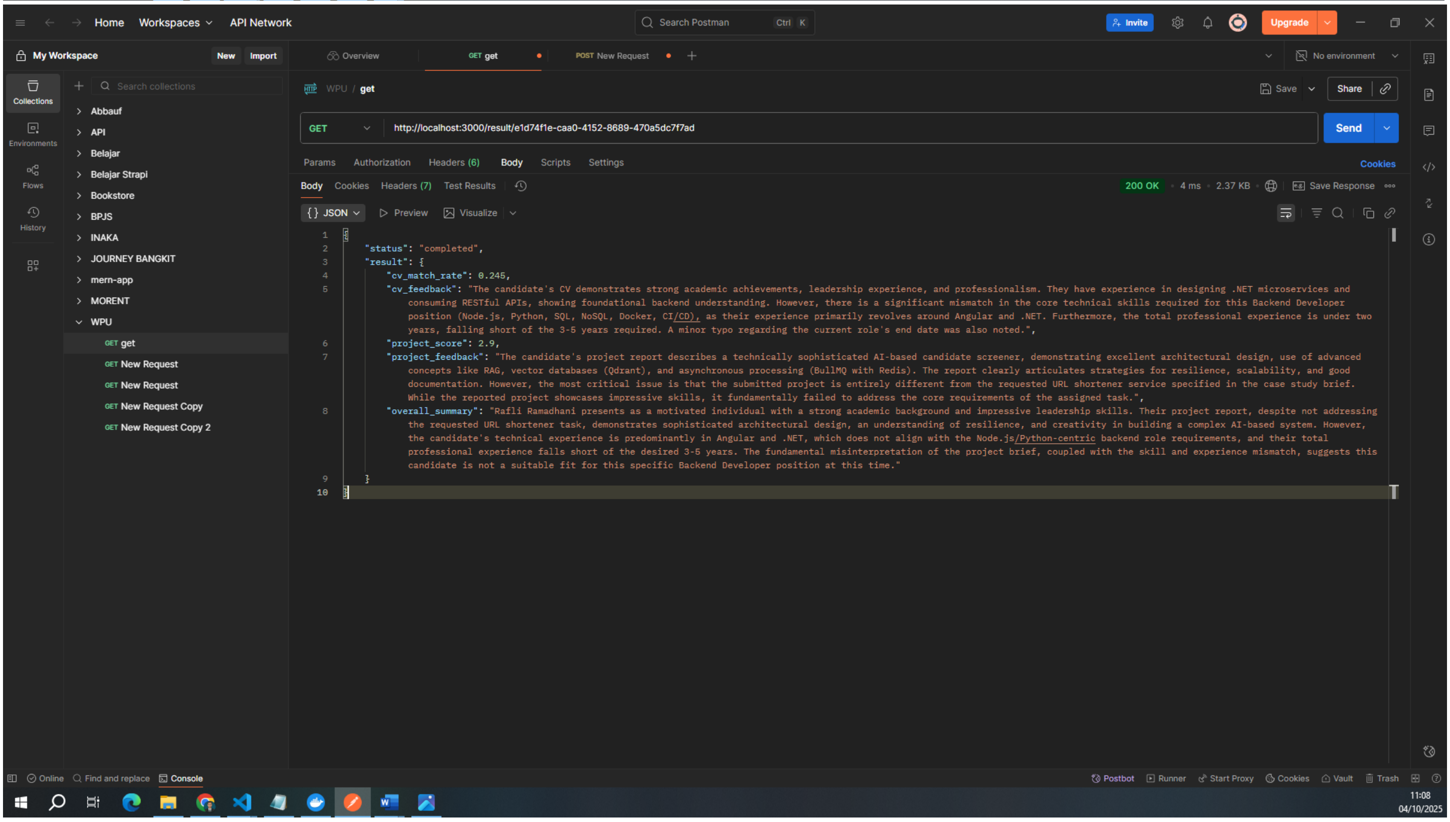
3. Future Improvements

With more time, the following areas could be enhanced:

- **Error Handling**: Implement more advanced retry logic with exponential backoff for Gemini API calls.
- **Testing**: Add unit tests and integration tests to ensure reliability of each component.
- **Frontend**: Build a simple user interface to streamline document upload and result checking.

4. Screenshots of Real Responses


candidate-screener-be
.gitignore
node_modules
.env
qdrantClien
No results
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
C:\dev\expressjs\candidate-screener-be>curl -X POST -F "cv=@my_cv.pdf" -F "report=@my_report.pdf" http://localhost:3000/upload
{"message":"Files uploaded successfully","cv_id":"cv-1759550668963-131091860.pdf","report_id":"report-1759550668968-145370830.pdf"}
C:\dev\expressjs\candidate-screener-be>curl -X POST -H "Content-Type: application/json" -d "{\"job_title\":\"Backend Developer\",\"cv_id\":\"cv-1759550668963-131091860.pdf\",\"report_id\":\"report-1759550668968-145370830.pdf\"}" "http://localhost:3000/evaluate"
{"job_id":"e1d74f1e-caa0-4152-8689-470a5dc7f7ad","status":"queued"}
C:\dev\expressjs\candidate-screener-be>curl http://localhost:3000/result/e1d74f1e-caa0-4152-8689-470a5dc7f7ad
{"status":"processing","data":{"job_title":"Backend Developer","cv_id":"cv-1759550668963-131091860.pdf","report_id":"report-1759550668968-145370830.pdf"}}
C:\dev\expressjs\candidate-screener-be>curl http://localhost:3000/result/e1d74f1e-caa0-4152-8689-470a5dc7f7ad
{"status":"completed","result":{"cv_match_rate":0.245,"cv_feedback":"The candidate's CV demonstrates strong academic achievements, leadership experience, and professionalism. They have experience in designing .NET microservices and consuming RESTful APIs, showing foundational backend understanding. However, there is a significant mismatch in the core technical skills required for this Backend Developer position (Node.js, Python, SQL, NoSQL, Docker, CI/CD), as their experience primarily revolves around Angular and .NET. Furthermore, the total professional experience is under two years, falling short of the 3-5 years required. A minor typo regarding the current role's end date was also noted.","project_score":2.9,"project_feedback":"The candidate's project report describes a technically sophisticated AI-based candidate screener, demonstrating excellent architectural design, use of advanced concepts like RAG, vector databases (Qdrant), and asynchronous processing (BullMQ with Redis). The report clearly articulates strategies for resilience, scalability, and good documentation. However, the most critical issue is that the submitted project is entirely different from the requested URL shortener service specified in the case study brief. While the reported project showcases impressive skills, it fundamentally failed to address the core requirements of the assigned task.","overall_summary":"Rafli Ramadhani presents as a motivated individual with a strong academic background and impressive leadership skills. Their project report, despite not addressing the requested URL shortener task, demonstrates sophisticated architectural design, an understanding of resilience, and creativity in building a complex AI-based system. However, the candidate's technical experience is predominantly in Angular and .NET, which does not align with the Node.js/Python-centric backend role requirements, and their total professional experience falls short of the desired 3-5 years. The fundamental misinterpretation of the project brief, coupled with the skill and experience mismatch, suggests this candidate is not a suitable fit for this specific Backend Developer position at this time."}}
C:\dev\expressjs\candidate-screener-be>
11:07
04/10/2025


My Workspace
WPU / get
GET http://localhost:3000/result/e1d74f1e-caa0-4152-8689-470a5dc7f7ad
Send
200 OK 4 ms 2.37 KB
{
"status": "completed",
"result": {
"cv_match_rate": 0.245,
"cv_feedback": "The candidate's CV demonstrates strong academic achievements, leadership experience, and professionalism. They have experience in designing .NET microservices and consuming RESTful APIs, showing foundational backend understanding. However, there is a significant mismatch in the core technical skills required for this Backend Developer position (Node.js, Python, SQL, NoSQL, Docker, CI/CD), as their experience primarily revolves around Angular and .NET. Furthermore, the total professional experience is under two years, falling short of the 3-5 years required. A minor typo regarding the current role's end date was also noted.",
"project_score": 2.9,
"project_feedback": "The candidate's project report describes a technically sophisticated AI-based candidate screener, demonstrating excellent architectural design, use of advanced concepts like RAG, vector databases (Qdrant), and asynchronous processing (BullMQ with Redis). The report clearly articulates strategies for resilience, scalability, and good documentation. However, the most critical issue is that the submitted project is entirely different from the requested URL shortener service specified in the case study brief. While the reported project showcases impressive skills, it fundamentally failed to address the core requirements of the assigned task.",
"overall_summary": "Rafli Ramadhani presents as a motivated individual with a strong academic background and impressive leadership skills. Their project report, despite not addressing the requested URL shortener task, demonstrates sophisticated architectural design, an understanding of resilience, and creativity in building a complex AI-based system. However, the candidate's technical experience is predominantly in Angular and .NET, which does not align with the Node.js/Python-centric backend role requirements, and their total professional experience falls short of the desired 3-5 years. The fundamental misinterpretation of the project brief, coupled with the skill and experience mismatch, suggests this candidate is not a suitable fit for this specific Backend Developer position at this time."
}
}
11:08
04/10/2025