Study Case Submission Template

Please use this template to document your solution. Submit it as a **PDF file** along with your project repository.

1. Title: LLM Integration For CV and Project Report Analyzer With RAG Implementation

2. Candidate Information

- **Full Name: M. Fadil Martias**
- **Email Address: fadilmartias26@gmail.com**

3. Repository Link

- https://github.com/fadilmartias/cv-analyzer

4. Approach & Design (Main Section)

Tell the story of how you approached this challenge. We want to understand your thinking process, not just the code. Please include:

- **Initial Plan**
  - After reviewing the requirements, I started by thinking about the tech stack that would let me move fast while keeping things robust. I decided to go with **Go** and the **Fiber framework**. Fiber is built on **fasthttp**, which is super fast, and its goroutines are perfect for handling queues and long-running tasks without blocking. Plus, being a compiled language, Go helps catch bugs early during development, so I can minimize headaches in production.

    For the database, I went with **PostgreSQL**, mainly because I'm familiar with it and it also has the **pgvector extension**, which lets me store embeddings directly in the database.

    For PDF reading, I initially tried **UniPDF**, but the free version couldn't extract text from scanned PDFs. After hitting some roadblocks, I switched to **Tesseract OCR** on my Windows machine, which worked much better for scanned documents.

    For AI/LLM, I started with **OpenRouter** because it's free and has lots of model options. But midway through, I realized OpenRouter doesn't provide embedding endpoints, so I pivoted to **Gemini by Google**, which supports embeddings and fits perfectly for RAG (Retrieval-Augmented Generation) workflows.

    The goal of this mini-project was to allow a user to **upload a CV and project report**, have an AI evaluate them asynchronously, and return detailed scores and feedback, while also leveraging vector databases for intelligent retrieval.
- **System & Database Design**

  I kept the system simple but scalable:
  - **Endpoints**:

    POST /evaluate → for uploading CV and project report

    GET /result/{id} → to fetch AI evaluation results
  - **Database**:

    evaluation_tasks: stores tasks with fields like cv, report, status, detailed scoring (cv_match_rate, cv_feedback, project_score, project_feedback, overall_summary, breakdown).

    jobs: stores job descriptions along with embeddings (title, content, embedding vector) for RAG retrieval.
  - **Long-running tasks**:

    I leveraged Go's **goroutines**. Every call to the LLM runs asynchronously, so /evaluate doesn't block.
  - **Project structure**:

    I followed **clean architecture**:

    usecase → business logic

    repository → database interaction

service → external services (like Gemini)
handler → HTTP endpoints
config → environment variables using godotenv
dto → consistent API responses
utils → helper functions like PDF extraction and format json response for all API Response

- **LLM Integration**
  - I used Gemini because it's free and supports embeddings.
  - The AI acts as a technical recruiter, evaluating both the CV and project report against job requirements.
  - I implemented RAG to fetch relevant job context for scoring, improving the relevance of AI feedback.
  - I created a small set of dummy jobs to populate embeddings and test the pipeline.
- **Prompting Strategy** (examples of your actual prompts)

```
You are an experienced technical recruiter. Analyze the following CV and Project Report against these job requirements:
(here is the context from RAG)
Return your answer STRICTLY in JSON format with this schema:
{
   "cv_match_rate": <float with 2 decimal places, range 0-1 based on cv breakdown score>,
   "cv_feedback": "<feedback about CV>",
   "project_score": <float with 2 decimal places, range 0-10 based on project breakdown score>,
   "project_feedback": "<feedback about Project Report>",
   "overall_summary": "<summary of overall impression, strengths, and areas to improve>",
  "breakdown": {
   "cv": {
   "technical_skills_match": <number 1-5, criteria: backend, databases, APIs, cloud, and AI/LLM exposure>,
   "experience_level": <number 1-5, criteria: years, project complexity>,
   "relevant_achievements": <number 1-5, criteria: impact, scale>,
   "cultural_fit": <number 1-5, criteria: communication, learning attitude>,
   },
   "project_report": {
     "correctness": <number 1-5, criteria: prompt design, chaining, RAG, handling errors>,
     "code_quality": <number 1-5, criteria: clean, modular, testable>,
     "resilience": <number 1-5, criteria: handles failures, retries>,
     "documentation": <number 1-5, criteria: clear README, explanation of trade-offs>,
     "creativity_or_bonus": <number 1-5, criteria: optional improvements like authentication, deployment, dashboards, etc.>
   }
  },

}

CV:
(extracted cv)

Report:
(extracted project_report)
```

- **Resilience & Error Handling**
  I wanted to make sure the system could handle all the usual hiccups:
  - Retry logic with **exponential backoff** for Gemini API calls
  - Timeout configurations for LLM requests

    - Circuit breaker to prevent cascading failures
    - Temperature set to **0.1** for consistent, low-hallucination results
    - Strict JSON validation before storing results
- **Edge Cases Considered**
    - Large, encrypted, or password-protected PDFs
    - Empty or minimal content → validation on CV length
    - Memory leaks in long-running goroutines → monitor goroutine counts
    - Invalid file types → add validation for check file types
    - Rate limiting: /evaluate calls limited to 1 per 4 seconds to stay under Gemini free-tier limits
    - Integration and load testing to ensure stability

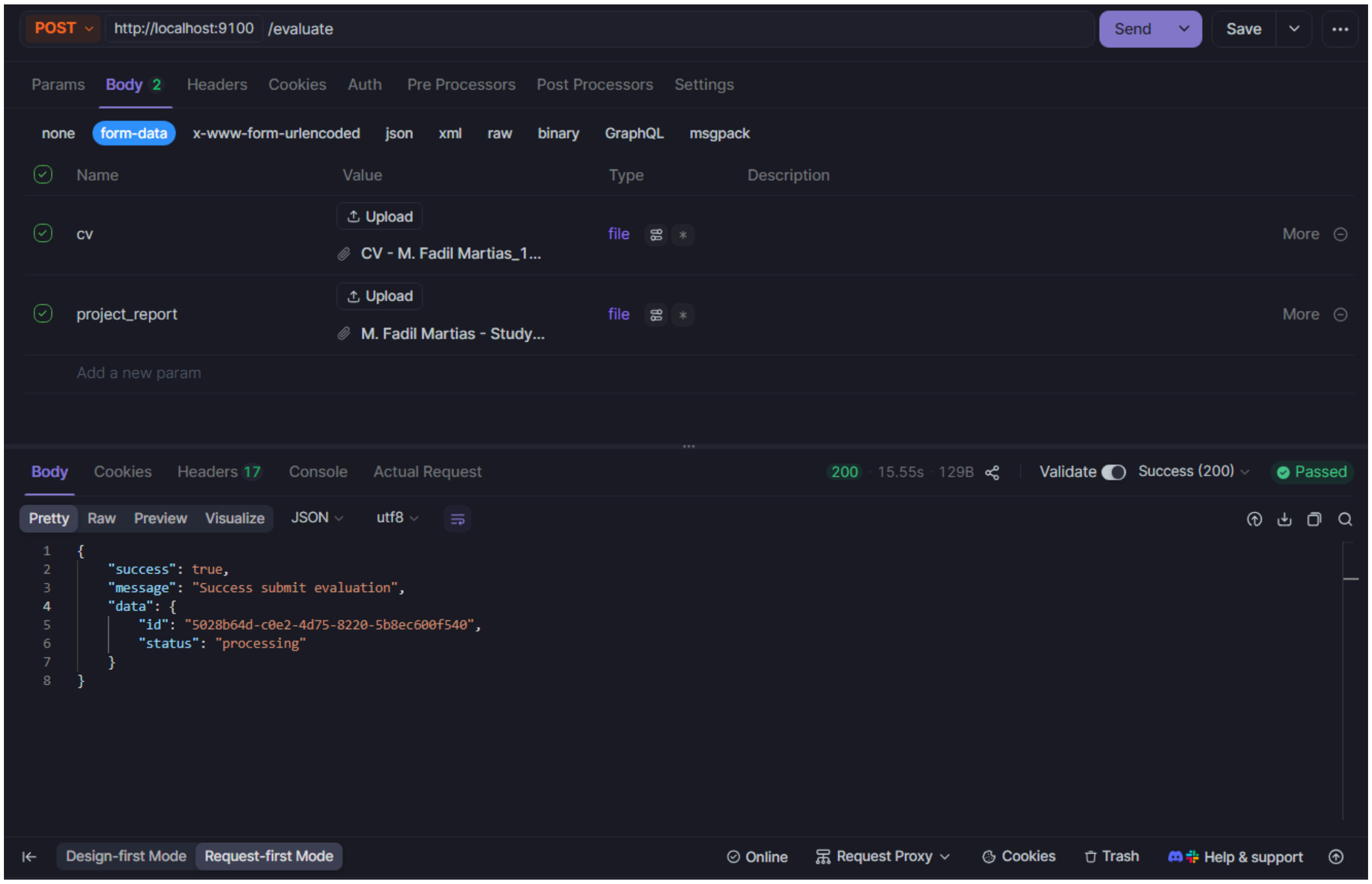
> ✍ This is your chance to be a storyteller. Imagine you're presenting to a CTO, clarity and reasoning matter more than buzzwords.

5. Results & Reflection

- **Outcome**
  - Everything worked as expected. Users can upload CVs and project reports, get async AI evaluations, and store results in Postgres with embeddings for retrieval.
- **Evaluation of Results**
  - AI outputs are consistent thanks to low temperature settings. JSON schema validation ensures structure is reliable.
- **Future Improvements**
  - Handle more error cases for maximum robustness
  - Possibly move to paid-tier LLMs or local embeddings for higher throughput
  - Improve PDF OCR accuracy for tricky scans

6. Screenshots of Real Responses

- GET /evaluate



http://localhost:9100 /evaluate

Name	Value	Type	Description
cv	Upload CV - M. Fadil Martias_1...	file	
project_report	Upload M. Fadil Martias - Study...	file	

Body — 200 15.55s 129B — Validate — Success (200) — Passed

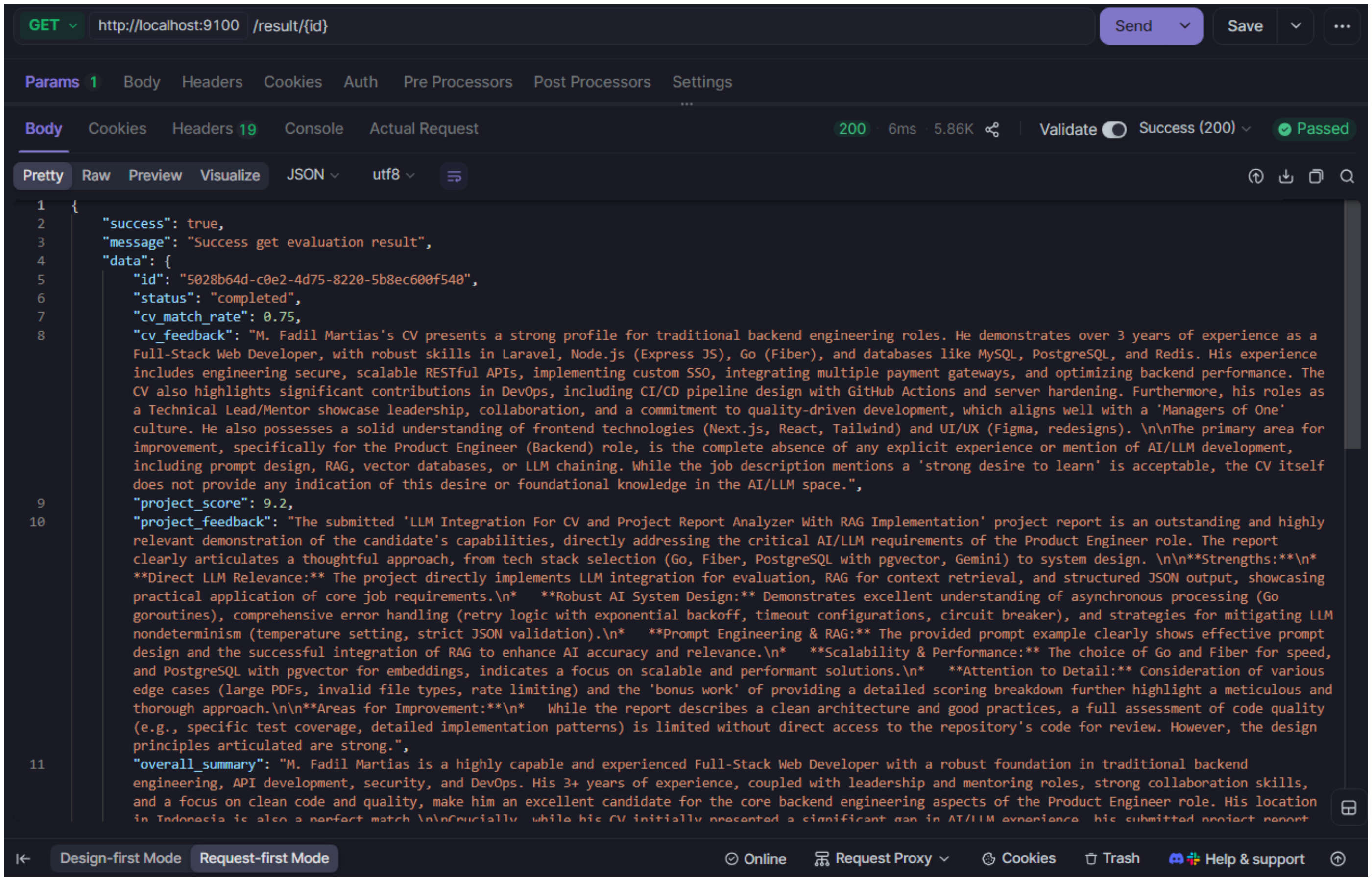
```
{
    "success": true,
    "message": "Success submit evaluation",
    "data": {
        "id": "5028b64d-c0e2-4d75-8220-5b8ec600f540",
        "status": "processing"
    }
}
```

Full Response:

```
{
"success": true,
"message": "Success submit evaluation",
"data": {
   "id": "5028b64d-c0e2-4d75-8220-5b8ec600f540",
   "status": "processing"
}
```

}

POST /result/{id}



```
{
    "success": true,
    "message": "Success get evaluation result",
    "data": {
        "id": "5028b64d-c0e2-4d75-8220-5b8ec600f540",
        "status": "completed",
        "cv_match_rate": 0.75,
        "cv_feedback": "M. Fadil Martias's CV presents a strong profile for traditional backend engineering roles. He demonstrates over 3 years of experience as a Full-Stack Web Developer, with robust skills in Laravel, Node.js (Express JS), Go (Fiber), and databases like MySQL, PostgreSQL, and Redis. His experience includes engineering secure, scalable RESTful APIs, implementing custom SSO, integrating multiple payment gateways, and optimizing backend performance. The CV also highlights significant contributions in DevOps, including CI/CD pipeline design with GitHub Actions and server hardening. Furthermore, his roles as a Technical Lead/Mentor showcase leadership, collaboration, and a commitment to quality-driven development, which aligns well with a 'Managers of One' culture. He also possesses a solid understanding of frontend technologies (Next.js, React, Tailwind) and UI/UX (Figma, redesigns). \n\nThe primary area for improvement, specifically for the Product Engineer (Backend) role, is the complete absence of any explicit experience or mention of AI/LLM development, including prompt design, RAG, vector databases, or LLM chaining. While the job description mentions a 'strong desire to learn' is acceptable, the CV itself does not provide any indication of this desire or foundational knowledge in the AI/LLM space.",
        "project_score": 9.2,
        "project_feedback": "The submitted 'LLM Integration For CV and Project Report Analyzer With RAG Implementation' project report is an outstanding and highly relevant demonstration of the candidate's capabilities, directly addressing the critical AI/LLM requirements of the Product Engineer role. The report clearly articulates a thoughtful approach, from tech stack selection (Go, Fiber, PostgreSQL with pgvector, Gemini) to system design. \n\n**Strengths:**\n* **Direct LLM Relevance:** The project directly implements LLM integration for evaluation, RAG for context retrieval, and structured JSON output, showcasing practical application of core job requirements.\n*   **Robust AI System Design:** Demonstrates excellent understanding of asynchronous processing (Go goroutines), comprehensive error handling (retry logic with exponential backoff, timeout configurations, circuit breaker), and strategies for mitigating LLM nondeterminism (temperature setting, strict JSON validation).\n*   **Prompt Engineering & RAG:** The provided prompt example clearly shows effective prompt design and the successful integration of RAG to enhance AI accuracy and relevance.\n*   **Scalability & Performance:** The choice of Go and Fiber for speed, and PostgreSQL with pgvector for embeddings, indicates a focus on scalable and performant solutions.\n*   **Attention to Detail:** Consideration of various edge cases (large PDFs, invalid file types, rate limiting) and the 'bonus work' of providing a detailed scoring breakdown further highlight a meticulous and thorough approach.\n\n**Areas for Improvement:**\n*   While the report describes a clean architecture and good practices, a full assessment of code quality (e.g., specific test coverage, detailed implementation patterns) is limited without direct access to the repository's code for review. However, the design principles articulated are strong.",
        "overall_summary": "M. Fadil Martias is a highly capable and experienced Full-Stack Web Developer with a robust foundation in traditional backend engineering, API development, security, and DevOps. His 3+ years of experience, coupled with leadership and mentoring roles, strong collaboration skills, and a focus on clean code and quality, make him an excellent candidate for the core backend engineering aspects of the Product Engineer role. His location in Indonesia is also a perfect match \n\nCrucially, while his CV initially presented a significant gap in AI/LLM experience, his submitted project report
```

Full Response:

{

"success": true,

"message": "Success get evaluation result",

"data": {

"id": "5028b64d-c0e2-4d75-8220-5b8ec600f540",

"status": "completed",

"cv_match_rate": 0.75,

"cv_feedback": "M. Fadil Martias's CV presents a strong profile for traditional backend engineering roles. He demonstrates over 3 years of experience as a Full-Stack Web Developer, with robust skills in Laravel, Node.js (Express JS), Go (Fiber), and databases like MySQL, PostgreSQL, and Redis. His experience includes engineering secure, scalable RESTful APIs, implementing custom SSO, integrating multiple payment gateways, and optimizing backend performance. The CV also highlights significant contributions in DevOps, including CI/CD pipeline design with GitHub Actions and server hardening. Furthermore, his roles as a Technical Lead/Mentor showcase leadership, collaboration, and a commitment to quality-driven development, which aligns well with a 'Managers of One' culture. He also possesses a solid understanding of frontend technologies (Next.js, React, Tailwind) and UI/UX (Figma, redesigns). ¥n¥nThe primary area for improvement, specifically for the Product Engineer (Backend) role, is the complete absence of any explicit experience or mention of AI/LLM development, including prompt design, RAG, vector databases, or LLM chaining. While the job description mentions a 'strong desire to learn' is acceptable, the CV itself does not provide any indication of this

desire or foundational knowledge in the AI/LLM space.",

"project_score": 9.2,

"project_feedback": "The submitted 'LLM Integration For CV and Project Report Analyzer With RAG Implementation' project report is an outstanding and highly relevant demonstration of the candidate's capabilities, directly addressing the critical AI/LLM requirements of the Product Engineer role. The report clearly articulates a thoughtful approach, from tech stack selection (Go, Fiber, PostgreSQL with pgvector, Gemini) to system design. ¥n¥n**Strengths:**¥n*   **Direct LLM Relevance:** The project directly implements LLM integration for evaluation, RAG for context retrieval, and structured JSON output, showcasing practical application of core job requirements.¥n*   **Robust AI System Design:** Demonstrates excellent understanding of asynchronous processing (Go goroutines), comprehensive error handling (retry logic with exponential backoff, timeout configurations, circuit breaker), and strategies for mitigating LLM nondeterminism (temperature setting, strict JSON validation).¥n*   **Prompt Engineering & RAG:** The provided prompt example clearly shows effective prompt design and the successful integration of RAG to enhance AI accuracy and relevance.¥n*   **Scalability & Performance:** The choice of Go and Fiber for speed, and PostgreSQL with pgvector for embeddings, indicates a focus on scalable and performant solutions.¥n*   **Attention to Detail:** Consideration of various edge cases (large PDFs, invalid file types, rate limiting) and the 'bonus work' of providing a detailed scoring breakdown further highlight a meticulous and thorough approach.¥n¥n**Areas for Improvement:**¥n*   While the report describes a clean architecture and good practices, a full assessment of code quality (e.g., specific test coverage, detailed implementation patterns) is limited without direct access to the repository's code for review. However, the design principles articulated are strong.",

"overall_summary": "M. Fadil Martias is a highly capable and experienced Full-Stack Web Developer with a robust foundation in traditional backend engineering, API development, security, and DevOps. His 3+ years of experience, coupled with leadership and mentoring roles, strong collaboration skills, and a focus on clean code and quality, make him an excellent candidate for the core backend engineering aspects of the Product Engineer role. His location in Indonesia is also a perfect match.¥n¥nCrucially, while his CV initially presented a significant gap in AI/LLM experience, his submitted project report *completely addresses and exceeds expectations* for the AI-powered systems requirements. This project demonstrates a strong aptitude for learning, practical implementation of RAG, prompt engineering, asynchronous processing, and robust error handling in an LLM context. It shows he can design and orchestrate LLM integrations effectively, making him a very strong contender for the Product Engineer (Backend) role.¥n¥n**Strengths:**¥n*   **Strong Backend Foundation:** Extensive experience with Laravel, Go, Node.js, databases, RESTful APIs, SSO, payment gateways, and CI/CD.¥n*   **Exceptional AI/LLM Aptitude:** The project report showcases a rapid and deep understanding of LLM integration, RAG, prompt design, and resilient AI system architecture, directly addressing the most critical and unique aspects of the Product Engineer role.¥n*   **Full-Stack Capability:** Solid understanding of frontend technologies (Next.js, React, Tailwind) and UI/UX (Figma, redesigns).¥n*   **Leadership & Collaboration:** Proven ability to lead, mentor, collaborate across teams, and drive quality.¥n*   **Robust Engineering Practices:** Demonstrated commitment to clean code, performance, security, and comprehensive error handling.¥n¥n**Areas to Improve:**¥n*   **CV Clarity on AI/LLM:** The CV itself could be updated to explicitly reflect his demonstrated interest and skills in AI/LLM development, as this was a blind spot without the project report.¥n*   **Cloud Provider Specifics:** While CI/CD and server hardening are mentioned, explicit experience with major cloud providers (AWS, GCP, Azure) is not detailed in the CV.",

"breakdown": "{¥"cv¥": {¥"cultural_fit¥": 4, ¥"experience_level¥": 4, ¥"relevant_achievements¥": 4, ¥"technical_skills_match¥": 3}, ¥"project_report¥": {¥"resilience¥": 5, ¥"correctness¥": 5, ¥"code_quality¥": 4, ¥"documentation¥": 4, ¥"creativity_or_bonus¥": 5}}",

"created_at": "2025-10-01T22:57:54.283155+07:00",

"updated_at": "2025-10-01T22:58:50.736339+07:00"

}

```
}
```

7. (Optional) Bonus Work

I added extra feature that can show breakdown of scoring metrics for cv and project_report