

# 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

## 2. Candidate Information

- **Full Name:**
- **Email Address:**


## 3. Repository Link

- Provide a link to your GitHub repository.
- **▲ Important:** Do **not** use the word *Rakamin* anywhere in your repository name, commits, or documentation. This is to reduce plagiarism risk.
- Example: *github.com/username/ai-cv-evaluator*

## 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**
  - How you broke down the requirements.
  - Key assumptions or scope boundaries.
- **System & Database Design**
  - API endpoints design.
  - Database schema (diagram or explanation).
  - Job queue / long-running task handling.
- **LLM Integration**
  - Why you chose a specific LLM or provider.
  - Prompt design decisions.
  - Chaining logic (if any).
  - RAG (retrieval, embeddings, vector DB) strategy.
- **Prompting Strategy** (examples of your actual prompts)
- **Resilience & Error Handling**
  - How you handled API failures, timeouts, or randomness.
  - Any retry, backoff, or fallback logic.
- **Edge Cases Considered**
  - What unusual inputs or scenarios you thought about.
  - How you tested them.

 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**
  - What worked well in your implementation?
  - What didn't work as expected?
- **Evaluation of Results**
  - If the evaluation scores/outputs were bad or inconsistent, explain why.
  - If they were good, explain what made them stable.
- **Future Improvements**
  - What would you do differently with more time?
  - What constraints (time, tools, API limits) affected your solution?

6. Screenshots of Real Responses

- Show **real JSON response** from your API using your own **CV + Project Report**.
- Minimum:
  - `/evaluate` → returns job\_id + status
  - `/result/:id` → returns final evaluation (scores + feedback)
- Paste screenshots or Postman/terminal logs.

---

7. (Optional) Bonus Work

If you added extra features, describe them here.