Case Title: AI Candidate Evaluation System

Objective:

You are asked to design and implement an AI-assisted backend evaluation system that analyzes uploaded CVs and project reports using retrieval and deterministic scoring logic.

Tasks:

1. Build a Django-based backend with RESTful APIs.

2. Implement endpoints for:

- /api/upload (upload CVs and reports)

- /api/evaluate (start an evaluation job)

- /api/result/ (fetch the evaluation result)

3. Extract text from uploaded PDFs (CVs, reports).

4. Implement a basic vector retrieval system (can be in-memory).

5. Compare extracted keywords to job descriptions to determine match rates.

6. Apply deterministic rules to calculate:

- CV match rate (% overlap with backend-related skills)

- Project score (degree of alignment with RAG/LLM implementation criteria).

7. Generate structured feedback text (simulating LLM-based summarization).

Expected Output:

A REST API that can process and evaluate candidate submissions.

Final result JSON example:

```
{

"cv_match_rate": 0.83,

"project_score": 0.85,

"cv_feedback": "Strong backend and cloud experience...",

"project_feedback": "Implements RAG structure partially...",

"overall_summary": "Good backend fit; strengthen RAG guardrails."

}
```

Evaluation Criteria:

- Correct implementation of endpoints.

- Functional deterministic scoring.
- Modular, maintainable code structure.
- Use of feature-based retrieval and vector similarity.