TORCH PROGRAMMING EXERCISE – AI/ML ENGINEER (PE #4)

**Programming Exercise Objectives:** The objective of this programming exercise is to design and implement an AI assistant capable of processing technical documents, extracting key information, and answering complex reasoning-based questions.

**Expected Completion Time:** Most candidates complete the exercise within 1-3 hours. We ask that you put as much or as little time into the exercise as necessary to best complete the objectives within reason; however, we do not expect an extensive effort nor a perfect solution. If there are clear gaps in your submission, please clearly explain in your documentation the additional steps you’d take if you had more time.

***Please note that candidates should not use ChatGPT or similar tools to complete any part of the exercise, nor should any provided information be entered into open online tools. This ensures the integrity and confidentiality of the assessment.***

**Submission Instructions:**

Once you complete the exercise, please package all relevant files in an organized .zip folder and submit to Lindsey Murphy [lmurphy@eccoselect.com](mailto:lmurphy@eccoselect.com) at least 48 hours prior to your scheduled interview.

1. The completed exercise will be reviewed by our engineering team and used for discussion during your second-round interview.
2. Please be prepared to discuss:
3. Your understanding of the programming exercise, including its objectives.
4. Any gaps in your understanding of the exercise or questions you have that may have assisted you in completing the exercise.
5. Your overall comfort and familiarity with performing this type of work.
6. Common challenges associated with this type of work.
7. The approach you took to solve these challenges.
8. The approach you would take if you had more time and resources available to you.

**Problem Statement:**

You are building an AI assistant for a company that processes technical documents (e.g., research papers, API documentation, or regulatory guidelines). The assistant should:

1. Extract key information from a set of documents.
2. Answer complex reasoning-based questions using both the provided documents and general knowledge.
3. Format responses in a structured way (e.g., JSON output for integration into another system).

Task 1: Document Processing & Retrieval-Augmented Generation (RAG)

* Given a set of technical documents (provided in PDF or plain text), create a system that:
* Parses and indexes the content efficiently.
* Allows for semantic search to retrieve relevant sections.
* Uses an LLM to summarize key concepts.

Task 2: LLM-Based Reasoning & Structured Response Generation

* Implement a solution where the LLM can:
* Answer complex questions that require reasoning across multiple documents.
* Generate responses in a structured format, such as JSON.
* Explain its reasoning process when answering ambiguous queries.

**Extra Credit:**

Task 3: System Design & Optimization

* Optimize the workflow by considering:
* Latency vs. accuracy trade-offs (e.g., caching, prompt tuning).
* Fine-tuning vs. API calls (if applicable).
* Handling of ambiguous or conflicting information.

Deliverables:

* A GitHub repository containing:
* Code & README explaining how to run the solution.
* Example input/output for different queries.
* A brief write-up on design choices and trade-offs.

*Disclaimer: Torch.AI will not use any code submitted as part of programming exercises in support of existing products or client solutions. Submissions are intended solely to evaluate a candidate’s technical abilities based on similar work the candidate may be expected to perform as a full-time employee. No actual client data or requirements are used in the design of the programming exercises.*