

INTERVIEW – Machine Learning Engineer

* -> Must meet these instructions, failure to do so implies immediate disqualification.

1. Assignment Instructions

A. Topic Selection (Choose ONE) *

- **Travel Planner**
 - Features: A chatbot capable of making Hotel reservations (inc. Cost estimates) and destination-specific activity suggestions. Activity implies – visiting tourist spots, hiking or going on a city tour. The chatbot should be able to recommend appropriate actions at the minimum for the user query and keep track of the conversation.
 - Flexibility: Basic functionality as defined above must be met. You are free to come up with additional requirements to design the application.
- **Document Assistant**
 - Features: Analyze a multi-page contract (minimum 5 pages) and provide a Q&A endpoint for the document. Here the chatbot must be capable of answering a single question posed by the user. The application must be able to answer at least 5 questions about the contract and provide a summary of the contract in 100 words.
 - Flexibility: Basic functionality as defined above must be met. You are free to select the contract of your choice and come up with additional requirements to design your application.

B. Codebase & Documentation Requirements

- **GitHub Repository ***
 - Ensure the codebase is public at the end of the 4-day period.
 - Include a note in the README that the application was developed for an interview with CriticalRiver.
- **Documentation ***
 - **README.md:** Must include reproduction instructions, steps to run the entrypoint, and examples of outputs. As long as the application runs locally, we are good.
 - **Architecture Diagram:** Include a diagram in the README showing your overall approach/method.
- **Code Quality**
 - Ensure code is commented, linted, and formatted with Black.
 - Provide all dependencies in a requirements.txt or pyproject.toml.

- We want to see some unit tests on critical functions in your codebase, you do not need to achieve high test coverage, as long as we have 5 unit test's we are good. Use unittest or pytest.

C. Frontend & Backend Tech Stack *

- Use Streamlit (or another simple framework) to build an interactive frontend for demonstration.
- All application code must be written in Python. For LLM models, use the best possible model that you can run locally from HuggingFace, including reasoning models where appropriate. We do not care about latency.

D. Timeline *

- **Duration:** 4 days from the assignment start.
- **Submission:** At the end of the fourth day, make the GitHub repo public and send the link to the HR team.

E. Data *

- Use public data sources only (**no private data**), to develop your application.

F. Additional Instructions

- **Q&A:** You may ask questions but note that extra time will not be provided beyond the 4-day or 96 hours window.
- **Design Flexibility:** While edge cases need not be fully covered, the application must demonstrate basic functionality (as provided in Features).
- We want to see you build the application with at least 2 ablations on architecture design choices. You are free to determine these ablations.

2. General Instructions *

- **Ownership & Rights:** You retain ownership of your code but must include a note at the top of README that the application was designed for an interview with CriticalRiver.
- **No Extra Time:** All work must be completed within the 4-day period or 96 hours from the point this activity is provided to you.