Assignment 10
LangChain & LCEL

Part C — Building GenAI Applications

18 questions

★	**Must-know** — asked very frequently in GenAI / ML interviews (7 questions). Prepare these first.
[CODE]	**Hands-on task** — write and run the code (5 tasks).

1. What is LangChain and how does it help build LLM applications? What are its main components? ★
2. Initialize an LLM / chat model (OpenAI or open-source) and ask one question. [CODE]
3. Create a prompt template with two input variables. [CODE]
4. What is LCEL (LangChain Expression Language)? Build a chain: prompt | model | output_parser. (Why did LCEL replace the legacy LLMChain?) ★
5. What is the difference between invoke, stream and batch? ★
6. What are output parsers? Name a few (StrOutputParser, JsonOutputParser, PydanticOutputParser) and their purposes.
7. What is parallelism in LCEL (RunnableParallel)? What are RunnablePassthrough and RunnableLambda?
8. What is a Document in LangChain? What are document loaders? Name a few.
9. Load a .txt, a .pdf and a .csv file as LangChain documents. [CODE]
10. Load all .pdf files and all .txt files in a folder (DirectoryLoader). [CODE]
11. How do you split documents? Explain RecursiveCharacterTextSplitter, chunk_size and chunk_overlap. Split a document into 1000-character chunks. ★
12. How do you summarize a document larger than the context window? Explain stuff, map_reduce and refine. ★
13. How do you make an LLM remember previous conversations? ★
14. Explain the memory types: buffer, buffer-window (what is 'k'?), summary and summary-buffer memory. Write code for one of them.
15. How is conversation memory handled in current LangChain/LangGraph (message history, checkpointers)?
16. How do you use the Hugging Face pipeline inside LangChain?
17. Use one chain as a tool inside another chain or agent. [CODE]

18. How do you handle long conversation memory without exceeding the context window or inflating token cost? ★ · Asked: Wipro