1. Explain the trade-offs between model size, inference speed, and accuracy in LLMs. How would these factors influence your choice of model for a real-time chatbot application?

Bigger models usually give more accurate and detailed answers because they have learned from more data. But the trade-off is that they’re slower and need more computing power, which can hurt the user experience in real-time chat. Smaller models are faster and cheaper to run, but they may miss context or give less precise answers. For a real-time chatbot, I’d prioritize speed and responsiveness, so I’d likely choose a mid-sized model that balances accuracy with quick replies. If needed, I could use a hybrid setup, using fast model for most chats and a larger model for tricky questions.

2. Describe a scenario where combining multiple LLMs would be beneficial for a complex application. How would you approach integrating these models?

Imagine a global customer support platform. One LLM could handle language translation, another could detect customer sentiment, and a third could generate the actual support response. By combining them, the system can serve users in multiple languages while also tailoring responses based on mood. To integrate these, I’d use a pipeline approach where the user input first goes to the translator, then the sentiment analyzer, and finally the response generator. Clear APIs between each model would make the workflow smooth and scalable.

3. Discuss the potential advantages and challenges of using AI-powered autonomous agents for software development tasks. Provide specific examples to support your answer.

The big advantage is automation. AI agents can handle repetitive tasks like writing boilerplate code, generating unit tests, or even reviewing pull requests. This saves developers time and speeds up delivery. For example, an agent could auto-generate CRUD APIs or test cases based on existing code. The challenge, though, is reliability. AI agents might introduce hidden bugs, produce insecure code, or make choices that don’t fit the team’s coding standards. That means developers still need to review everything carefully.

4. How might the rise of autonomous AI agents impact the role of human developers in the software development process? Consider both short-term and long-term implications.

In the short term, developers will probably shift toward more of a “supervisor” role, guiding AI agents, reviewing their work, and focusing on higher-level design decisions. This could boost productivity without replacing humans. In the long term, as AI agents get more advanced, routine coding might become fully automated. Human developers may focus less on writing code and more on defining problems, designing systems, and making ethical or business decisions. The role won’t disappear, but it will evolve into something more strategic and creative.