General questions

Question 1

As a software engineer, you should provide efficient and clear code practices. Can you

run us through your specific code practices and how you document your code?

A: Before coding, I would take the effort to select appropriate algorithms and data structures based on time and space complexity. After that, I'm used to writing class, variable, and function names that are clear and adhere to style and indentation standards. To enforce this, use formatters and linters. It helps to make the code readable. To make sure the various components of the system function as intended, I write tests for individual code units during the testing process.

I use block comments to illustrate sections of code or complex logic. Describe the purpose and major properties of classes for documenting code.

Question 2

What is your approach to building an application that is talking to many different

services? How would you set up an architecture to support various integrations from

different services (3rd party API, DBs, embedding of Sub Applications)? Can you give us

some best practices you would follow and give us an example of your ideas?

A: I Use different databases suited to different needs.

**Relational Databases (SQL)**: Use for structured data with complex relationships. Examples: PostgreSQL, MySQL.

**Document Databases (NoSQL)**: Use for semi-structured data or documents. Examples: MongoDB, CouchDB.

I design schemas and use normalization in SQL databases to eliminate redundancy, and denormalization in NoSQL databases to optimize read efficiency, depending on each database's database capabilities and access patterns.

Question 3

Effective management of codebases often depends on the Git branching strategy

employed. What is your preferred Git branching model, and why?

A: GitFlow is my preferred Git branching strategy because of its structured approach that allows for concurrent work and continuous delivery. I'm used to using the main branch and committing stuff to it.

Question 4

What are your preferred practices for deploying code to production? Can you explain to us your methodology and give examples on how you would implement a production pipeline?

A: I don’t have much experience in deploying code to production