

ADITI GIRI

Q. Explain blocking and non-blocking operations/approach OR Explain synchronous and asynchronous operations/approach

Synchronous/Blocking Approach/Operations

- **Synchronous** is a **blocking** architecture, so the execution of each operation depends on completing the one before it.
- **Synchronous** execution means the code is executed in sequence line by line (one line at a time). For example, when a function is called, the program execution waits until that function returns before going to the next line of code.
- **Synchronous** is blocking — it will only send the server one request at a time and wait for that request to be answered by the server.
- **Synchronous** is slower and more methodical.
- Eg. Traditional Web Server Model is **synchronous and blocking**

Asynchronous/Non-Blocking Approach/Operations

- **Asynchronous** is a **non-blocking** architecture, so the execution of one task isn't dependent on another. Tasks can run simultaneously.
- **Asynchronous** execution means the code doesn't necessarily run in the sequence. The program doesn't wait for the code block to finish its execution and can move on to the next piece of code.
- **Asynchronous** is non-blocking, which means multiple requests can be processed in parallel without waiting for the earlier request processing to be completed.
- **Asynchronous** increases throughput because multiple operations can run at the same time.
- Eg. Node.js Process Model is Asynchronous and Non-Blocking.