Advanced Node.js concepts

- Asynchronous Programming
  - → Event loop
    - Phases :- Timers, I/O callbacks, Poll, check, close callbacks
    - Tasks :- Executes callbacks in queue, manages non blocking I/O operations.
  - → Promises and Async / Await
    - Promises :- '.then()', '.catch()', '.finally()'.
    - Async :- syntactic sugar for Promises, allow writing asynchronous code in a synchronous style.
    - Error Handling :- 'try... catch'
- Streams
  - → Types of streams :-
    - Readable
    - Writable
    - Duplex
    - Transform.
  - → Method and events.
    - Methods :- '.pipe()', '.unpipe()', '.on()'.
    - Events :- 'data', 'end', 'error', 'finish'.
  - → Usage
    - Efficient handling of large data
    - Chaining with '.pipe()' for streaming operations.
- Child processes.
  - → forking
    - Usage :- to create a new node.js process.
    - IPC (inter process communication) :- Communicate between parent and child processes using 'send' and 'on' methods.

→ Exec and spawn.

- Exec :- Execute a command in a shell & buffers O/P.
- Spawn :- Launch new process with given command.

→ Events :- 'exit', 'error', 'message', 'disconnect'.

→ Cluster Module

→ Clustering

- Usage :- create child process that share same server port.
- Load Balancing :- Distribute incoming connections.

→ Events :- 'online', 'listening', 'exit', 'disconnect'.

→ Implementation.

→ Debugging Tools.

- Node inspector :- Applications using chrome DevTools.
- Vs code debugger :- Integrated debugging in Visual studio.

→ Profiling

- CPU profiling :- Identify performance bottleneck.
- Heap snapshots :- Analyze memory usage & leaks.

→ Security :-

- Injection attacks :- SQL, NOSQL.
- Environment variables :- Use '.env' files to manage.
- Secure coding :- Use HTTPS, handle errors properly.

→ Scalability and performance optimization.

- Horizontal scaling and vertical scaling.
- HTTP caching :- Use cache-control headers.
- Asynchronous code :- Avoid blocking event loop, use async
- Tools :- 'morgan', 'winston', 'Prometheus' etc.
- Practices :- Implement logging and real-time monitoring.