Advance Node.js concepts sheet

- Middleware → This is crucial for handling and modifying requests and responses in an express application. They can perform various task such as logging, authentication error handling etc.

- Asynchronous programming
  Node.js employs asynchronous programming to handle multiple operations concurrently without blocking the main thread. This approach allows Node.js to manage I/O operation efficiently.

- Streams → They are used to process data incrementally, which is especially useful for handling large files or data sets. There are 4 type of streams readable writable duplex & transform streams.

- Event loop → The event loop is a mechanism that enable node.js to peeform non-blocking I/O operation. It continuously checks for the processes events or callbacks.

Child processes →

Node.js can spawn processes to execute other scripts or system commands leveraging the underlying operating system capabilities for multiprocessing.

Cluster module → allow you to create child processes that share the same server port. This is useful for taking advantage of multi-core system, enabling better scalability & performance.

Debugging & profiling

↳ They are essential for diagnosing and optimizing Node.JS application. Tools like Node.js built-in debugger, chrome Dev tools provide capabilities.

Security → It involves practices & tools to protect applications from various vulnerabilities. Common security measures include validating & sanitizing user imputs, implementing authentication & authorization.