1. Introduction
   1. Node.js
   2. The Callback Pattern
   3. Event Emitter
   4. Error Handling
   5. Buffers
   6. Streams
   7. Async Patterns
2. Architecture of Node.js
   1. Installation dependencies
   2. How to use user-defined call backs effectively
   3. Comparison between events and callbacks
   4. Limitation on the number of blocks executed
3. Socket Handling
   1. All types of protocols (TCP, HTTP, Web Services)
   2. Express.js
   3. SocketIO.js
4. Clustering
   1. Shared memory Usage
   2. Comparison on Master-child process and pm2 related clustering
5. Best Practices in Node.js
   1. Commonly used add-on modules
   2. Suggestions where the synchronous calls are necessary
   3. Products or companies which use Node.js
6. Memory Management
   1. Memory Management concepts
   2. Cache (Redis)
   3. Garbage collection
7. Queue Management
   1. Queue Management concepts
   2. Random access from queue
   3. Queue Maintenance in Redis
8. File Handling (All types of file creations)
9. GUI support in Node.js
10. Database Support
    1. Different types of DB supported
    2. Connection pooling
    3. Redis configuration
11. Process Management & Monitoring
    1. High availability
    2. Fail over
    3. Scalability
    4. Node to node communication
12. Sample Implementation architecture using various add-on modules
    1. Consider different systems (Linux, IBM AIX, SPARC, HP Non-Stop Systems)
    2. Utilization of memory
    3. Node.js parameters at various level to be taken care
    4. How to implement in cloud computing
    5. Framework supported