

Event driven Architecture

- Utilizes an event driven arch, enabling non-blocking, asynchronous handling of I/O operation
- Events are triggered by various action such as incoming req

Asynchronous Programming

- Handle I/O bound operations efficiently
- Call back fun are commonly used to manage asynchronous task
- Promises and `async/await` → alt for handling code

NPM

- It is default Package manager for Node.js
- Vast ecosystem of packages
- `package.json` serves as a manifest file for npm projects

Modules

- Node.js follows the commonJS module system
- Module encapsulates code and data
- `require()` is used to import module

Error handling

- Crucial to ensure robustness and ~~code~~ reliability
- Errors are handled using `try-catch` block, `error` first callback, promises or `catch` method

Stream
Handle large volume of data efficiently

- Types include readable, writable, duplex and transform
- allow data to be processed in chunks

Web Frameworks

Express Js, ~~Koa~~ Koa.js, Hapi.js and Next.js

Testing

- ensures reliability & correctness of node.js app
- Mocha, Jest & Jasmine are powerful ~~too~~ frameworks for running test
- Common methodology are - Test driven development (TDD) and behaviour driven development (BDD)

Caching

Implement Caching mechanism to store frequently accessed data and reduce response time

Optimised code

Write efficient code by minimizing synchronous operation, avoiding unnecessary I/O and optimizing algorithms