BACKEND NOTES

- **A Load Balancer is a system that sits in front of your backend servers and distributes incoming requests among them.**

- **Node.js is single-threaded, yet it can handle multiple I/O operations asynchronously without blocking execution. This is possible because of the Event Loop. the api calls etc are delecated to web apis**

- **A thread is the smallest unit of execution within a process. It represents a single sequence of instructions that a CPU can execute independently.**

- **A Thread Pool is a collection of worker threads that execute tasks concurrently in the background. It is mainly used for handling CPU-intensive and I/O-bound tasks efficiently in Node.js. Thread can be considererd as a worker.**

- **Event Loop:**
    1. **Event Loop watches event/task queue and move it to task stack**
    2. **Call Stack manages the execution**
    3. **Task Queue holds web api callback to execute after certain period**
    4. **Event loop moves task from task queue to call stack to**
    5. **MicroTask queue is for async await .then,catch,finally**

- **NODE JS Architecture**
    1. **Firstly request goes to request queue**
    2. **Event loop then checks if it is a blocking or non blocking operation. If it is blocking, it goes to thread pool to look for a worker**
    3. **After that it sends back the response**

- **Both async/await and promise are used to handle asynchronous operations**

- **High Order Function Takes function as argument and returns function. Callback function are passed as argument**

- **Middleware acts as a bridge between client and server. Whenever client requests the server, it first passess through middleware.**

- **Streams in Node.js are a way to handle reading/writing data efficiently. Instead of loading the entire data at once (which can be memory-intensive), streams process data in chunks, making them ideal for handling large files, network requests, and real-time data.**

- **To build scalable backend, you can do horizontal or vertical scaling, implement load balancer and also can use micro services.**

- **Development To Production:**
    1. **Push your local code to cloud services like github.**

2. **Best practise is that instead of pushing to main branch, it is first pushed to dev branch, If everything is okey in dev branch, we move to staging branch and test. Lastly we move to production branch**
3. **Once different branch are merged, CI tool (GitHub Actions, GitLab CI, Jenkins) runs tests & builds the application.**
4. **CI/CD automates the process of building, testing, and deploying code**
5. **Continuous Integration (CI):**
   Developers frequently merge code changes into a shared repository, and automated builds and tests are run to ensure the reliability of the merged code.
6. **Continuous Delivery/Deployment (CD):**
   After successful integration and testing, the code is automatically deployed to production or a staging environment, either continuously (deployment) or with manual approval (delivery).

- **Payload is what client sends to server**

- **Backend Security**
  1. **Size of payload**
  2. **Check size of payload**
  3. **Content**
  4. **Rate limit**
  5. **Encrypt Data**
  6. **Use helmet, becrypt, Joi, cors, express-rate-limiter npm packages**

- **Webhook is like notification system between apps. For example: If payment is done like through stripe. It sends notification to our backend.**

- **Content negotiation is a mechanism in HTTP that allows a client and a server to agree on the best content format for a response.**

- **A RESTful API (Representational State Transfer API) is an API that follows the principles of REST (Representational State Transfer), a software architectural style used for building web services:**
  1. **Uses HTTP method**
  2. **Stateless**
  3. **each endpoint corresponds to a resource.**

- **SOAP is a protocol used for exchanging structured information in web services communication. Used XML only. Slower but best for security. Used in Financial Banking etc**

- **GraphQL is a query language and runtime for APIs, developed by Facebook in 2015. Unlike REST, which requires multiple endpoints for different resources, GraphQL allows clients to request exactly the data they need in a single query. Clients can request exactly the fields they need, avoiding over-fetching or under-fetching.**

GIT AND GITHUB NOTES

- **Git helps you keep track of changes in your project, so you can go back to an earlier version if something goes wrong.**
- **Github helps to store project in cloud**
- **Stagging is like adding to the cart while shopping**
- **Only after staging you commit**
- **Use reset cmd to remove the done commits**
- **Commit is like a checkpoint**
- **Use revert to just go back in the commit. It maintains the history**
-

DATABASE NOTES

- **DBMS helps to manage and create databases. Eg: MySQL, Mongodb**

- **Indexing in MongoDB is a way to improve the speed and efficiency of queries by creating special data structures that store a small portion of the collection's data in an organized manner. It works similarly to an index in a book, allowing MongoDB to find documents faster instead of scanning the entire collection.**

- **A transaction in a database is a sequence of one or more operations that are executed as a single unit of work. It ensures that either all operations succeed or none of them are applied, maintaining data integrity.**

- **Models are responsible for creating and reading documents from the underlying MongoDB database.**

- **Mongoose is a library**

- **A Schema in Mongoose defines the structure (blueprint) of documents in a collection. It specifies field names, data types, validation rules, and default values.**

- **Model provides an interface to query, create, update, and delete documents in MongoDB.**

- **Mongoose discriminators are a feature that allows you to create multiple models that share the same schema structure but have some variations.**
- **Middleware is used to perform certain task if crud operation occurs**