Green indicate to read theory	yellow indicate to make project on this conccept

Backend Using NodeJS

1	Backend Communication and protocols	A set of formal rules describing how to transmit or exchange data, especially across a network. Web, HTTP, DNS. SSH RPC (Remote Procedure Call) SQL and Database Protocols APIs (REST/SOAP/GraphQL) Implemented in variations
2	Express	A node js web application framework that provides broad features for building web and mobile applications It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes
3	ORMs (Object Relational Mapping)	ORM helps to store the data using object, It's helps to store object to the relational databases
4	Middlewares	A software design pattern where functions are invoked sequentially in a pipeline to handle requests and responses in web applications.
5	Routes	determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on)

6	Status Code	Status codes are a set of universal web development codes utilized to create apps all over the world. Informational responses (100–199) Successful responses (200–299) Redirects (300–399) Client errors (400–499)	
7	Global Catches	In Node.js, "global catches" typically refer to mechanisms for handling errors and exceptions that are not caught by individual try-catch blocks within the application.	process.on ('uncaughtException', (err) => { console.error ('There was an uncaught error', err); process. exit(1); // Mandatory to exit after handling the error });
8	Zod	Zod is a schema declaration and validation library for typescript. It allows us to design schemas for any data, including primitive data types, objects, arrays, and more.	

9	MonoRepos and TurboRepo	A monorepo (short for monolithic repository) is a software development strategy where multiple projects are stored in a single repository. This approach contrasts with using multiple repositories (each containing a single project) and offers several benefits:	Turborepo is a tool designed to optimize the management and performance of monorepos. It provides features that address some of the common challenges associated with monorepos, making it easier to maintain and scale large repositories with multiple projects.
10	Serverless Backend	https://blog.back4app.com/what-is-a-serverless-backend/	
11	OpenAPI Spec	The OpenAPI Specification (OAS) is a standard, language-agnostic format for describing RESTful APIs. It allows developers to define the structure and behavior of their APIs in a way that is both machine-readable and human-readable.	

12	Autogenerated Clients	Autogenerated clients are software libraries or SDKs (Software Development Kits) that are automatically generated from an API specification, such as the OpenAPI Specification. These clients facilitate interaction with the API by providing pre-built functions or methods to perform API operations, abstracting away the complexities of HTTP requests and responses.
13	Authentication using external libraries	
14	Scaling nodejs Performance benchmark	Scaling a Node.js application involves optimizing its performance to handle increased load efficiently. Performance benchmarking is a critical part of this process, providing data to understand how the application performs under different conditions and identifying potential bottlenecks.
15	Deploying PM packages	
16	Advanced Backend Communication, Message Queue and Pubs	Advanced backend communication involves utilizing sophisticated techniques and technologies to enable efficient, scalable, and robust interactions between various backend services, databases, and external systems. This is crucial in modern applications, particularly microservices architectures, where multiple services need to communicate seamlessly. Here are some advanced concepts and techniques: https://dev.to/jay818/advanced-backend-communication-redis-pub-sub-queue-880

17	Proxies and Load Balancer	A proxy server can help you stay anonymous online, while a reverse proxy allows for better control of the content you serve. A load balancer is the best choice when it comes to dealing with high-traffic volumes on a single website or application.	
18	Redis deep dive and Kafka deep dive	Redis (REmote DIctionary Server) is an open source, in-memory, NoSQL key/value store that is used primarily as an application cache or quick-response database.	Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data.
19	Common design patterns in JS	https://www.geeksforgeeks.org/javascript-design-patterns/	
20	Advanced DB concept(Indexing, Normalization)	In a database, normalization involves structuring the data to minimize redundancy and improve data integrity. This helps ensure that the data is organized efficiently and prevents inconsistencies. So, in simpler terms, an index helps with faster data retrieval, like finding information in a book's index.	

21	Rate Limiting	limiting access for people (and bots) to access the API based on the rules/policies set by the API's operator or owner.
22	Sharding, replication, resiliency and horizonta and vertical scaling	**Sharding:** Distributing data across multiple machines to improve performance and scalability. **Replication:** Copying data to multiple servers to ensure redundancy and availability. **Resiliency:** Building systems that can recover from failures and continue to function. **Horizontal Scaling:** Adding more machines to distribute load. **Vertical Scaling:** Adding more resources to a single machine to handle increased load.
23	Captchas and DDoS protection	CAPTCHA is used to mitigate DDoS attacks, as legitimate users are able to pass it, while attacking computers cannot.
24	gRPC	gRPC is a high-performance, open-source Remote Procedure Call (RPC) framework developed by Google. It enables communication between distributed systems or microservices using a binary protocol over HTTP/2. Here’s a brief overview:

25	Capacity Estimation	Capacity Estimation in Systems Design explores predicting how much load a system can handle. Imagine planning a party where you need to estimate how many guests your space can accommodate comfortably without things getting chaotic. Similarly, in technology, like websites or networks, we must estimate how much traffic they can handle before crashing.