S.No	Topic	Status
1	Microservices	Done
2	Graphs, Subgraphs and Gateway	Done
3	Drizzle ORM with Mysql and Nodejs (including connection	Done
4	Nestjs (project structure, DIs)	Done
5	Graphql	Done
6	Apollo Server	Done
7	Code First Approch	Done
8	Authentication	Done
9	Package Management	Done
10	Fastify	Done
11	Typesafety , eslint for linting	Done
12	prettier	Done
13	data class validators nestjs	Done
14	Apis with CRUD, dashboard and user activity monitoring s	Done
	Channels	Done
	Users	Done
	Posts, Likes , Comments , Replies	Done
15	Dashboard functionality	Done
16	User Activity	Done
	users tracking	Done
	Post activities	Done
	Reply activities	Done
	Comment Activity	Done
17	RBAC Authorization	Done
18	Testing of features	Done
	user-service (RBAC) + Features	Done
	Gateway	Done
	Post service features + RBAC	Done
19	Containerization using Docker	Done
	How to generate build	Done
	How to write dockerfile	Done
	How to write docker compose yml file	Done
	Cli commands to work with docker	Done
	Configurations and setup	Done
	Port mapping and concepts like containerization of packag	Done
20	Learning about caching and redis	Done
21	Security features	Done
	CORS for url limiting	Done
	@Fastify/rate-limit (for protection from DDOS attacks)	Done
	class-validators for protection from SQL injection like attac	Done
	JWT tocken for hashing of password + Authenticating use	Done

	Some additional security topics covered theoretically - Throttling (secures from malicious requests and prtects s - Fastify/helmet for secure connection	Done
22	Study on how to make Robust systems -high availability, fault tolerance, security, scalability, and n	Done
23	Need of web sockets and how service like kafka helps in t	Done
	Code management using Github project link - > https://github.com/arthgupta11/ThreadHive	Done
25	Unit testing using jest	Done
	Users CRUD	Done
	User Channels Mapping CRUD	Done
	using nestjs/testing	Done
	resolver testing + rbac testing + CRUD dAO TESTING	Done