**Day 11: 13 April 2025**

**Bcryptjs** : it is a type of external module which help to encrypt and decrypt the text password.

Create the folder as **password encrypt example**

Open the terminal inside this folder

npm init -y

npm install **bcryptjs**

**those all rest api. can be access by any user or it can be access publicly.**

**Authentication and authorization**

**Authentication is use to access the resources.**

**Base upon the role they apply some restriction for particular resource to access.**

**Admin role**

**Manager role**

**Guest role**

**Product end point or rest api**

**Store product**

**Delete product**

**Update product**

**View the product**

**Authentication must be there to access these end points.**

**To access these end point you need to do login.**

**Admin role -🡪 Can add, delete, update and view the product**

**Manager role -🡪 update the product price or qty and view the product etc.**

**Customer role 🡪 view the product and place order for that particular product.**

**By default http is a stateless protocol.**

**1st (http/https)---🡪**

**2nd req + sessionId**

**3rd req + sessionId**

**Client Server**

**1st res(http/https) + cookies**

**Cookies : Cookies is a small text file created by server when first time client send the request. This file contains lot of information with sessionId (unique id) for each client. This file send to client machine. 2nd or 3rd or 4th request. This file store in client. Client can disable cookies option in browser. If cookies disable means you are not allowing to store the session id. Cookies is not a secure.**

**JWT (JSON Web Token)**

**Product end point or rest api**

**Store product**

**Delete product**

**Update product**

**View the product**

**We configure middleware module. Before all any endpoint we check token present in your required or not. If not there we will not allow to call the any end point.**

**Token may be there but invalid token. Also we not allow to call.**

**Token is there but expired. Also we not allow to call any end point.**

**Rest api**

**Post method : SignUp --🡪 emailid(unique), password as well as other information**

**Post method : SignIn -🡪 Emailid and password -🡪 we check emailid and password if correct we generate one JWT token with algorithm 256 default and payload (emailid). We provide token to frontend technologies.**

**To generate the token using node js. Node JS provide third party library or modules**

**Jsonwebtoken**

**Create the folder json-web-token-app**

**npm init -y**

**npm install jsonwebtoken**

**creating backend app with login and product entity with token and password encrypt**

**create the folder**

login, product app with token with crud operation

npm init -y package.json

**npm install express mongoose** jsonwebtoken **bcryptjs dotenv**

express 🡪 to create rest api

mongoose 🡪 to connect database mongo db with schema base

jsonwebtoken -🡪 to generate the token

bcryptjs 🡪 hashing password

dotenv 🡪 load environment details.