Authentication And Authroization

Q1. What is authentication? and use case of it.

Ans=> In simple words, Authentication is the process of verifying who a user is (who you are).or
In authrntication process the identity of user are checked
for provinding the access to the system.or
In authentication process, users or persons
are verified.Authentication determines whether the person is user or not.

Q2. How can we authenticate data with NodeJS?

Ans=> Authentication Options in Node. js.
Build a Simple Node. js App with Authentication.
Initialize the Node. ...
Install Node.js Dependencies.
Define Node.js Controllers.
Create a Simple Web Server in Node.
Test the Application Using Postman.
Implement Token-Based Authentication in Your Node.
==>In authentication we use jwt joken to vrfy a user
When sucessfullt login attempt we create the jwt token and sen back the user and where authentication needed we decode the token and check users identity.

Example=>.

A common example is entering a username and password when you log in to a website.
Entering the correct login information lets the website know
1) Who you are and
2)decoded the jwt token if token decoded that means you are the the real user not face user

Q3. how is authentication different from authorization.

Ans=> Simply put, authentication is the process of verifying who someone is, whereas authorization is the process of verifying what specific applications, files, and data a user has access to.

Q4. what should be perform before authentication or authorization.

Ans=>authentication bcoz once we vrefiyd who the person is only than we can allow and give access to that person to update(put or delete). So the conclusion Is that first we can perform the authentication the go to the

authorization process.the flow is that first we need to verify the user login or not then we check the the user have the access or not to do the particuler task like edit,update,delete,change etc.

Q5. What is jwt token and explain in code.

Ans=>JWTs are a good way of securely transmitting information between parties because they can be signed, which means you can be sure that the senders are who they say they are. Additionally, the structure of a JWT allows you to verify that the content hasn't been tampered with.
what are three parts of jwt.

Or jwt is npm package who provide us the functionality to craete a token and varify thr token basically jwt has two main functionality the first one is create a token
Jwt.sign =>

```
jwt.sign(payload, secretOrPrivateKey, expierttime)
```

With the help of jwt.sign we can create a token to user and send the user
Payload- payload conatin the key value paire of any user identtty like name phone anything we have.

Jwt.verify =>

```
jwt.verify(token, secretOrPublicKey, [options, callback])
```

With the help of jwt .verify we can decode the token after decoded it gives the user identty
Token- token which we want to ecoded.

SecretOrPublicKey-secretOrPublicKey is the key which we will given the toekn create time. When token is created or login api

Q6. What are the three parts of JWT.

Ans=>a header, payload, and signature.

HEADER
The information contained in the header describes the algorithm used to generate the signature. The decoded version of the header from the above example looks like:

```
{
"alg": "HS256",
"typ": "JWT"
}
```

PAYLOAD

All the claims within JWT authentication are stored in this part.The decoded version of the payload from the JWT example provided above looks like:

```
{
"sub": "1234567890",
"name": "John Doe",
"iat": 1516239022
}
```

The ‘name’ field is used to identify the user to whom the token was issued to. The ‘sub’ and ‘iat’ are examples of registered claims and are short for ‘subject’ and ‘issued at’.

SIGNATURE

The signature part of a JWT is derived from the header and payload fields.

1. Combine the base64url encoded representations of header and payload with a dot (.)

base64UrlEncode(header) + “.” + base64UrlEncode(payload)

2. Hash the above data with a secret-key only known to the server issuing the token. The hashing algorithm is the one described inside the header.

hash_value = hash([base64UrlEncode(header) + “.” + base64UrlEncode(payload)], secret-key)

3. Base64Url encode the hash value obtained from the step above

Signature = base64UrlEncode(hash_value)

Because the ‘secret-key’ is only known to the server, only it can issue new tokens with a valid signature. Users can not forge tokens as producing a valid Signature for the token requires the knowledge of the ‘secret-key’.

Q7. why do we write auth code in middleware in nodejs.

Ans=> When creating protected routes in Express, you need to know the user's authentication status before executing the logic of route controllers. Thus, authentication in Express is a step in the request-response cycle, which you can implement as middleware.

Q8. Why should we write auth in middleware.

Q9. jwt is it application level or route level middleware.

Ans=> both it will. as global and route specific

Q10. What is Authorization? and use case of it.

Ans=> In simple words,Authorization is the process of verifying what they have access to (what you are allowed to do).

in authorization process, person’s or user’s authorities are checked for accessing the resources or in this process, users or persons are validated.it determines What permission do user have?