Date : 8th - Dec- 2020
Morning Session : 9am – 11pm
By ~ Surya

<mark>Topics:</mark> JSON Web Token (JWT) and page Rendering

Yesterday what is jwt and how jwt works what are the advantages

And we started base app authentication with we are using mongoose

```
1    const mongoose = require("mongoose");
2
3    const UserSchema = mongoose.Schema({
4        email: {
5            type: String,
6            required: true
7        },
8        password: {
9            type: String,
10           required: true
11       },
12       createdAt: {
13           type: Date,
14           default: Date.now()
15       }
16   })
17
18   module.exports = mongoose.model("user", UserSchema);
19
```

Encoding : once we encode we can decode it back using key -- two way
Hasing : once we hash we can not get back the value -- its only way
Loging : password -- createHash(password) --passwordHash --
Bryprt : bcrypt.hash(create a password hash) , bcrypt.compare(hash1,hash2)

Create route for users

Authentication vs authorization :

Authentication is Nothing but we are giving access to application like login and signup

Authorization means a user already authenticated successfully but does not have permission to access that resource.

After creating login and signup we need to validate the token whether the token valid or not

Validation comes under middleware

```javascript
module.exports = mongoose.model("user", UserSchema);

const jwt = require("jsonwebtoken");

module.exports = function(req, res, next) {
    const token = req.header.token;
    if(!token) {
        return res.status(401).json({message: "Auth Error"})
    }
    try{
        //decoding the jwt token
        const decoded = jwt.verify(token, "randomString")
        req.user = decoded.user;
        next();
    } catch(e) {
        console.error(e);
        return res.status(500).json({message: "Error in fetching"})
    }
}
```

Interview Question:

```
1    //this in javascript
2    1.const user = {
3        name: 'surya',
4        getName: function() {
5            console.log("this.name", this.name)
6        }
7    }
8    user.getName();
9
10   const user = {
11       name: 'surya',
12       getName: () => {
13           console.log("this.name", this.name)
14       }
15   }
```

 When we use **user.getname** what would be the output:

Please go through the recorded lecture for code implementation