Training Day 20

16th July 2025

TOPICS COVERED

- **Exporting in Mongoose**

To keep code modular and clean, schemas and models can be exported and imported in different files.

Example – userModel.js:

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

  name: String,

  email: String,

  password: String,

});


const User = mongoose.model('User', userSchema);

module.exports = User;
```

Import in another file:

```
const User = require('./userModel');
```

- **HTTP Messages**

HTTP messages include both requests (sent from client to server) and responses (sent from server to client).

Request Parts:

Method (GET, POST, PUT, DELETE)

Headers (info about the request)

Body (data, if any)

Response Parts:

Status Code (e.g., 200 OK, 404 Not Found, 500 Server Error)

Headers

Body (the response data)

Example:

res.status(200).json({ message: "Success" });

- **Hashing**

Hashing is a one-way process to convert data (like passwords) into a fixed-length string.

It's irreversible and mainly used to store passwords securely.

- **Encryption**

Encryption converts data into a coded format so that only authorized users can decrypt it using a key.

Unlike hashing, encryption is reversible.

- **bcrypt** (for Hashing Passwords)

bcrypt is a library used to hash passwords in Node.js.

Installation:

npm install bcrypt

- **Salt and Salt Rounds**

Salt adds randomness to passwords before hashing.

Salt Round: Number of times bcrypt will process the data. More rounds = stronger but slower.

- **Hashing Passwords using bcrypt**

```
const bcrypt = require('bcrypt');
const password = "mypassword";
const saltRounds = 10;

bcrypt.hash(password, saltRounds, (err, hash) => {
  if (err) throw err;
  console.log("Hashed Password:", hash);
});
```

- **Comparing Passwords**

```
const enteredPassword = "mypassword";
bcrypt.compare(enteredPassword, hashedPassword, (err, result) => {
  if (result) {
    console.log("Password matched");
  } else {
    console.log("Password incorrect");
  }
});
```

TOOLS USED

VS Code – writing modular JS files

Postman – testing POST requests (e.g., registration with passwords)

MongoDB Compass – verifying hashed passwords stored

bcrypt – for password hashing

Browser Console – debugging messages