Training Day 19

15th July 2025

TOPICS COVERED

- **Mongoose**

Mongoose is an ODM (Object Data Modeling) library for MongoDB and Node.js. It provides a structured way to interact with MongoDB using schemas and models.

Installation:

npm install mongoose

Basic Connection:

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

mongoose.connect('mongodb://localhost:27017/myDB').then(() => {

  console.log('MongoDB connected');

}).catch(err => console.log(err));
```

- **Schema and Model**

A Schema defines the structure of your documents in a MongoDB collection.

```
const userSchema = new mongoose.Schema({

  name: String,

  age: Number,

  email: String,

});

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

Adding Data:

```
const newUser = new User({ name: 'abc', age: 19, email: 'abc@example.com' });

newUser.save();
```

- **Middleware in Express.js**

Middleware functions run during the request-response cycle.

Inbuilt Middleware

express.json() – parses JSON request bodies

express.urlencoded() – parses form data

express.static() – serves static files

Custom Middleware:

```
app.use((req, res, next) => {

  console.log("Successfully entered in middleware");

  next(); // must call next()

});
```

- **req.params**

Used to capture URL parameters (route variables):

```
app.get('/user/:id', (req, res) => {

  res.send(`User ID: ${req.params.id}`);

});
```
URL: http://localhost:3000/user/123 → req.params.id = 123

- **req.query**

Used to access query parameters (after ? in the URL):

```
app.get('/search', (req, res) => {

  res.send(`You searched for: ${req.query.keyword}`);

});
```

URL: http://localhost:3000/search?keyword=games → req.query.keyword = "games"

- **req.body**

Contains POST/PUT data (requires middleware express.json() or express.urlencoded()):

app.use(express.json());

```
app.post('/register', (req, res) => {
  const { name, email } = req.body;
  res.send(`Registered user: ${name}, Email: ${email}`);
});
```

Use Postman to send JSON:

```
{
  "name": "abc",
  "email": "abc@example.com"
}
```

TOOLS USED

Visual Studio Code (VS Code)

Postman – for testing API requests (GET, POST, etc.)

MongoDB Compass – for viewing database records

Browser Console – for frontend/debug logging

Mongoose – for MongoDB object modeling