Express MongoDB Back End Development 01

Module: 07

Assignment: 01

Part 1: Short Answer Questions:

Q.1: What is client-side and server-side in web development, and what is the main difference between the two?

Ans: Client-side:

The Client-side is everything in a web application displayed or occurring on a client's (end user's) device, including what they see – images, text, and the rest of the UI. It also involves any actions performed by the app within the user's browser.

Server-side:

The part of the web application that does not interact with the user but works on the back end of the web application is called the server side.

The main difference Between Client-side and Server-side is:

Client-side does not interaction with the server, But Server-side requires interaction with the server.

Q.2: What is an HTTP request and what are the different types of HTTP requests?

Ans: An HTTP request is **made by a client, to a named host, which is located on a server**. The aim of the request is to access a resource on the server.

The different types of HTTP requests are:

Get, Post, Put, Patch, Delete, Copy, Head, Options, Link, Unlink, Purge, Lock, Unlock, Propfind and View. ***(Collected from postman)***

Q.3: What is JSON and What is it commonly used for in web development?

Ans: The full from of JSON is JavaScript Object Notation. JSON is a text-based, human-readable data interchange format used to exchange data between web clients and web servers. The format defines a set of structuring rules for the representation of structured data.

In web development, JSON is commonly used in the following ways:

Data transmission, API responses, Configuration files and Data storage.

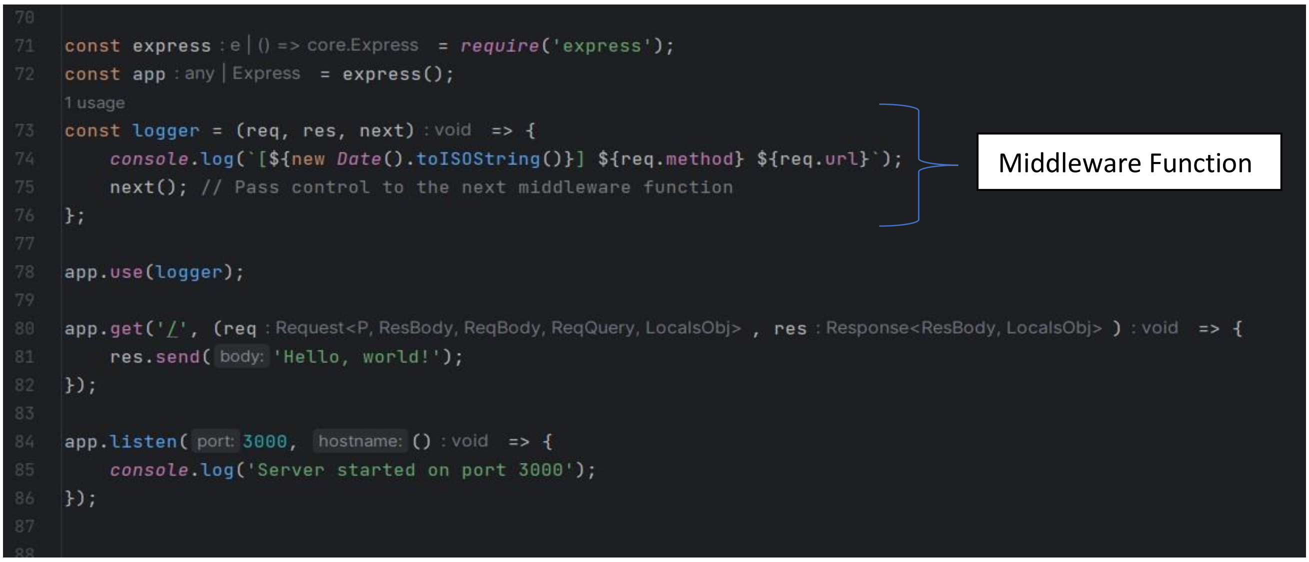
Overall, JSON plays a crucial role in web development by providing a standardized format for data exchange, enabling interoperability between different systems and applications.

Q.4: What is a middleware in web development, and give an example of how it can be used?

Ans: **Middleware:**

In web development, Middleware refers to a software component or layer that sits between the web application and the server or framework it runs on. It acts as a bridge, intercepting and processing requests and responses, often performing tasks such as authentication, logging, input validation, and request modification.

Here is an Example how it can be used:



```javascript
const express = require('express');
const app = express();
const logger = (req, res, next) => {
    console.log(`[${new Date().toISOString()}] ${req.method} ${req.url}`);
    next(); // Pass control to the next middleware function
};

app.use(logger);

app.get('/', (req, res) => {
    res.send('Hello, world!');
});

app.listen(3000, () => {
    console.log('Server started on port 3000');
});
```

Middleware Function

Q.5: What is a controller in web development, and what is its role in the MVC architecture?

Ans: **Controller:**

In web development, a controller is a component or module that handles the logic and behavior of a particular request and response.

In the Model-View-Controller (MVC) architecture, the Controller is one of the three core components responsible for handling user interactions and coordinating the overall flow of the application. The main role of the Controller is to receive input from the user and translate it into actions to be performed by the Model or the View.