

# Express MongoDB Back End Development 01

## Assignment

### Part -1

- 1) **What is client-side and server-side in web development, and what is the main difference between the two?**

**Ans: Client-side development:** Execution of code on the user's device (typically a web browser) using languages like HTML, CSS, and JavaScript. It focuses on the user interface, interactivity, and rendering of content. Examples include React, Angular, and Vue.js. Advantages include a fast user experience, reduced server load, and the ability to create dynamic interfaces.

**Server-side development:** Execution of code on the server that hosts the web application, handling user requests, data processing, and generating dynamic content. It involves languages like Python, Ruby, Java, or JavaScript (with Node.js). Examples include Django, Ruby on Rails, Express.js, and Spring Boot. Advantages include enhanced security, centralized data management, and easier maintenance and updates.

- 2) **What is an HTTP request and what are the different types of HTTP requests?**

**Ans:** An HTTP request is a message that is sent from a client to a server. It is used to request a resource, such as a web page, from the server. The request includes the following information:

- Request method
- Request URI (Uniform Resource Identifier)
- Headers
- Request body

Different types of HTTP requests are:

- GET
- POST
- PUT
- DELETE
- PATCH

- 3) **What is JSON and what is it commonly used for in web development?**

**Ans:** JSON (JavaScript Object Notation) is a lightweight data interchange format used to transmit structured data between a server and a web application.

It is commonly used in web development for the following tasks:

- Sending data from the server to the client
- Storing data in a database
- Serializing and deserializing objects
- Creating APIs

- 4) **What is a middleware in web development, and give an example of how it can be used?**

**Ans:** Middleware in web development is a software component that sits between the server and the application or framework. It acts as a bridge, adding functionalities like request processing, authentication, logging, and error handling. It separates concerns and allows for modular and reusable code.

Here's an example of how middleware can be used in a web application built with a framework like Express.js, which is based on Node.js:

```
const express = require('express');
const app = express();

// Middleware function
const logger = (req, res, next) => {
  console.log(`[${new Date().toISOString()}] ${req.method} ${req.url}`);
  next(); // Calls the next middleware function
};

// Registering middleware globally
app.use(logger);

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

app.get('/users', (req, res) => {
  res.send('User list');
});

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

##### 5) What is a controller in web development, and what is its role in the MVC architecture?

**Ans:** In web development, a controller is a component that handles the logic and behavior of a web application. It acts as an intermediary between the user interface (views) and the data model. In the MVC architecture, the controller receives user input, interacts with the model to perform operations, and updates the view accordingly. Its role is to synchronize and coordinate the application's flow of data and actions. The controller helps separate concerns and promotes code reusability and maintainability.

Here's a simplified overview of the controller's role in the MVC architecture:

- **User interaction:** The user interacts with the view, triggering actions such as submitting a form, clicking a button, or making a request.
- **Controller handling:** The controller receives the user's input from the view and processes it.
- **Model interaction:** The controller interacts with the model to perform operations such as retrieving or updating data.
- **View update:** After performing necessary actions on the model, the controller updates the view to reflect the changes.
- **User feedback:** The updated view is presented to the user, allowing them to see the result of their interaction.