1. What are the request methods in http?

1.GET

2.POST

3.PUT

4.DELETE

5.PATCH

6.HEAD

7.OPTIONS

8.TRACE

9.CONNECT

2. Find the out the difference between GET and POST methods?

GET:GET is used to request data from a specified resource.

Get is used for retrieving data from the server.

GET requests can be cached

GET requests remain in the browser history

GET requests can be bookmarked

GET requests should never be used when dealing with sensitive data

GET requests have length restrictions

GET requests are only used to request data (not modify)

GET Data is visible to everyone in the URL

POST:POST is used to send data to a server to create/update a resource.

POST requests are never cached

POST requests do not remain in the browser history

POST requests cannot be bookmarked

POST requests have no restrictions on data length

POST Data is not displayed in the URL

3. How to send the files as response in express js?

Transfers the file at the given path. Sets the Content-Type response HTTP header field based on the filename's extension. Unless the root option is set in the options object, path must be an absolute path to the file.

it is a send() function sends the HTTP response. The body parameter can be a String or a Buffer object or an object or an Array. Parameter: This function accepts a single parameter body that describes the body to be sent in the response. Returns: It returns an Object

4. Create express js program to display employee details as follows (Use HardCoded Values):

Employee Id : 10256

Employee Name : Narasimha Rao

Employee Job : Manager

Employee Deptno : 10

Employee EmailId : tnrao.trainer@gmail.com

var express = require("express");

var app = express();

app.get("/", function(req, res){

let str ="<h3 align ='center'>Employee Details</h1>";

str += "<hr>";

str += "<div>";

str += "Employee Id : 2584548 </br>";

str += "Employee Name : Mounika </br>";

str += "Employee Job : Trainer </br>";

str += "Employee dep : Computers</br>";

str += "Employee email : mounikapoola958@gmail.com</br>";

str +="</div>";

res.send(str);

});

var server = app.listen(3005,function(){});

console .log("Express server Application is started. Browser at the URL: <http://localhost:3005/>");



5. Create another program to update the above program with variables and also use template literals to format the strings.

Hint : Apply following Styles as per your convenience .

var express = require("express");

var app = express();

app.get("/", function(req, res){

let empId = 2584548;

let empName ='Mounika';

let empJob = 'Trainer';

let empdep = 'Computers';

let empemail='mounikapoola958@.com';

let str ="<h3 align ='center'>Employee Details</h3><hr>";

str += `<div style="color:blue;font-family:georgia,sans,sarif;">

Employee Id : ${empId} <br>

Employee Name : ${empName}<br>

Employee Job : ${empJob}<br>

Employee dep : ${empdep}<br>

Employee email : ${empemail}<br>

</div>`;

res.send(str);

});

var server = app.listen(3005,function(){});

console .log("Express server Application is started. Browser at the URL: <http://localhost:3005/>");



6. Create express js program to display product details as follows (Use variables)

Product Id : 10256

Product Name : LG Printer

Unit Price : 2500

Quantity : 3

Total Amount : 7500

Hint : Use html table to present the above data

const Express = require("express");

var app = Express();

app.get("/", function (req, res) {

let productId = 10256;

let productName = "LG Printer";

let unitPrice = 2500;

let quantity = 3;

// let totalAmount = 7500;

let totalAmount = unitPrice \* quantity;

let html = `

<style>

h2{

font-family:roboto;

}

table {

font-family:roboto;

border-collapse: collapse;

width: 50%;

margin: auto;

}

th, td {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h2 align='center'>Product Details</h2>

<table>

<tr>

<th>Product Id</th>

<td>${productId}</td>

</tr>

<tr>

<th>Product Name</th>

<td>${productName}</td>

</tr>

<tr>

<th>Unit Price</th>

<td>$ ${unitPrice}</td>

</tr>

<tr>

<th>Quantity</th>

<td>${quantity}</td>

</tr>

<tr>

<th>Total Amount</th>

<td>$ ${totalAmount}</td>

</tr>

</table>

`;

res.send(html);

});

var server = app.listen(3005, function () {});

console.log("This is Express App browse it at : <http://localhost:3005/>");

