Experiment No.8

Name: Gavande Rahil Hussain
Roll no: 412018
Subject: Web Computing Lab

AIM: Implementation of Express and REST API using NodeJS.

Description:

ExpressJS: Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open-source framework developed and maintained by the Node.js foundation.

RESTful APIs: An API is always needed to create mobile applications, and single-page applications, use AJAX calls and provide data to clients. A popular architectural style of how to structure and name these APIs and the endpoints are called REST (Representational Transfer State). HTTP 1.1 was designed keeping REST principles in mind. REST was introduced by Roy Fielding in 2000 in his Paper Fielding Dissertations.

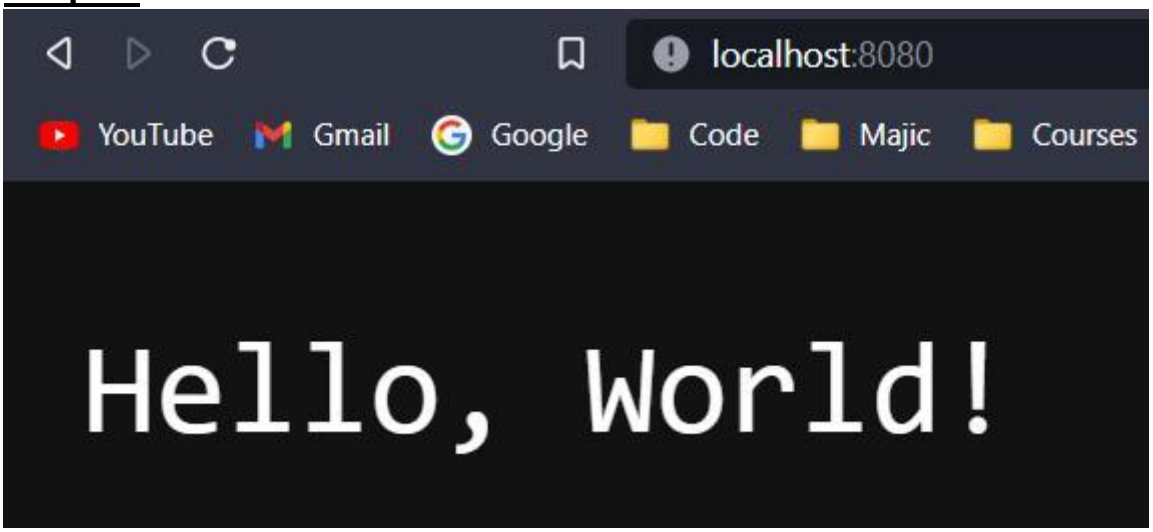
RESTful URIs and methods provide us with almost all the information we need to process a request. The table given below summarizes how the various verbs should be used and how URIs should be named. We will be creating a movies API towards the end; let us now discuss how it will be structured.

a) Implementation of the HTTP server.

Code:

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

const requestListener = function (req, res) { res.writeHead(200);
res.end('Hello, World!');
}
const server = http.createServer(requestListener);
server.listen(8080);
```

Output:



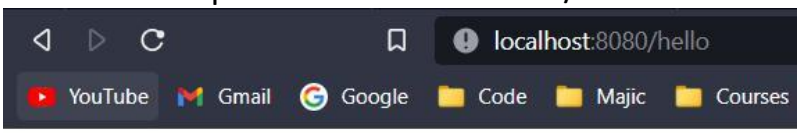
b) Implementation of express with different routes.

Code:

```
var express = require('express');
var app = express();
app.get('/hello', function(req, res){
res.send("Hello Weirdo!");
});
var a=100,b=200,c;
c=a+b;
app.get('/arith', function(req,res){ res.send("Addition of a + b ="+c);
});
app.listen(8080);
```

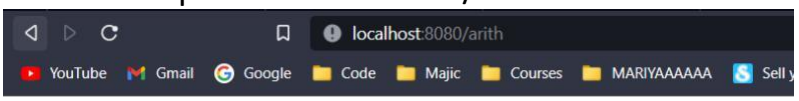
Output:

Hello world implementation on route /hello:



Hello World!

Arithmetic operation on Route /arith:



Addition of a + b =300

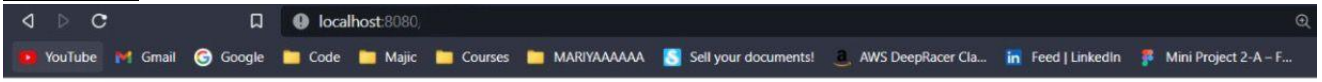
c) Implementation of express with MySQL Connectivity.

Code:

```
const express = require("express");
const app = express();
const mysql = require('mysql');
const connection = mysql.createConnection({
    host     : 'localhost',
    user     : 'root',
    password : 'weirdo',
    database : 'books'
});

connection.connect((err) => { if(err) throw
err; console.log('Connected to MySQL Server!');
});

app.get("/",(req,res) => {
connection.query('SELECT * from book', (err, rows) => { if(err) throw
err; console.log('The data from book table are: \n', rows);
res.write("<h1>"+JSON.stringify(rows)+"</h1>");
```

```
//connection.end();
});
});

app.listen(8080, () => {
console.log('Server is running at port 8080');
});
```

Output:



[{"bookid":1,"title":"wings of fire","price":200},
{"bookid":2,"title":"As a man thinketh","price":100},
{"bookid":3,"title":"Wise and unwise","price":300},
{"bookid":4,"title":"Royale blue","price":250},
{"bookid":101,"title":"wings of fire","price":250},
{"bookid":102,"title":"Royale blue","price":150},
{"bookid":103,"title":"As a man thinketh","price":190}]

Conclusion: Hence, we successfully created http server and performed express using different routes in NodeJS.