CBCS : 2020-21 M.Sc.-II   Computer Science

Web Frameworks Practical’s # Practical’s

Note : Install node js and visual studio code on your machine

Q. Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.

Q. Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.

Q. Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.

Q.5. Create a Simple Web Server using node js

```
var http = require("http");

http.createServer(function (request, response) {
   // Send the HTTP header
   // HTTP Status: 200 : OK
   // Content Type: text/plain
   response.writeHead(200, {'Content-Type': 'text/plain'});

   // Send the response body as "Hello World"
   response.end('Welcome\n');
}).listen(8081);


// Console will print the message
console.log('Server running at http://127.0.0.1:8081/');
```

Open browser=>

http://localhost:8081/

O/P: Hello World

Q.4. Create a Simple Web Server using node js to print today’s date on the browser.

```
var http = require("http");

http.createServer(function (request, response) {
   // Send the HTTP header
   // HTTP Status: 200 : OK
   // Content Type: text/plain
   response.writeHead(200, {'Content-Type': 'text/plain'});

   // Send the response body as "Hello World"
   response.end('Todays Date: \n'+Date());
}).listen(8081);

// Console will print the message
console.log('Server running at http://127.0.0.1:8081/');
```

Open browser=>

http://localhost:8081/

O/P: Todays Date:

```
Thu Jan 12 2023 16:00:52 GMT+0530 (India Standard Time)
```

Q 6. Create a Node.js file that will convert the output "Hello World!" into upper-case letters:

On terminal:

npm install upper-case

Upper.js

```
var http = require('http');
var uc = require('upper-case');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(uc.upperCase("Hello World!"));
    res.end();
}).listen(8081);
```

http://localhost:8081

Q 7. Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error .

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('input1.txt', function(err, data) {
    if (err) {
      res.writeHead(404, {'Content-Type': 'text/html'});
      return res.end("404 Not Found");
    }
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

Open browser=>

http://localhost:8080/

Q. 8. Create a Node.js file that opens the requested file and appends the given content to the file.

```
 var fs = require('fs');


fs.appendFile('input.txt', 'Hello contentnew!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Q 9. Using nodejs create a web page to read two file names from user and append contents of first file into second file.

```
var fs = require('fs');
fs.readFile('first.txt', function(err, data){

fs.appendFile('second.txt', data, function (err) {
  if (err) throw err;
  console.log('Saved!');
});
});
```

Open browser=>

Q. Create a Node.js file that demonstrate create database and table in MySQL

```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "127.0.0.1",
  user: "root",
  password: "indira"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE college", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });

});
```

```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "127.0.0.1",
  user: "root",
  password: "indira",
  database: "college"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE student (name VARCHAR(255), class VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

Q 10. Create a node.js file that Select all records from the "customers" table, and display the result object on console

```
var mysql = require('mysql');

var con = mysql.createConnection({
    host: "127.0.0.1",
    user: "root",
    password: "indira",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
});
```

Q 11. Create a node.js file that Insert Multiple Records in "student" table, and display the result object on console

```
var mysql = require('mysql');

var con = mysql.createConnection({
    host: "127.0.0.1",
    user: "root",
    password: "indira",
    database: "college"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO student (name, class) VALUES ('Dip Shaha', 'MSc II'), ('Geet Kale', 'MSc I')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(" record inserted");
  });
});
```

Q 12 . Create a node.js file that Select all records from the "customers" table, and delete the specified record.

```
var mysql = require('mysql');

var con = mysql.createConnection({
    host: "127.0.0.1",
    user: "root",
    password: "indira",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
    var sql = "DELETE FROM customers WHERE address = 'wakad'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Number of records deleted: " + result.affectedRows);
  });
});
});
```

Q 13. Write node js script to build Your Own Node.js Module. Use require (‘http’) module is a built-in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, “modules.js” and add this function to return today’s date and time.

Solu:

Modules.js

```
exports.myDateTime = function () {
  return Date();
};
```

DemoModules.js

```
var http = require('http');
var dt = require('./modules');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("The date and time are currently: " + dt.myDateTime());
  res.end();
}).listen(8080);
```

Open browser=>

http://localhost:8081/

o/p:

```
The date and time are currently: Wed Jan 26 2022 05:00:12 GMT+0530 (India Standard Time)
```

Q 14. Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those events is detected.

```
// Import events module

var events= require('events');

// Create an eventEmitter object

 var eventEmitter = new events.EventEmitter();

// Create an event handler as follows

var connectHandler = function connected() { console.log('connection
succesful.');

// Fire the data_received event
eventEmitter.emit('data_received'); }

// Bind the connection event with the handler
 eventEmitter.on('connection', connectHandler);

// Bind the data_received event with the anonymous function
eventEmitter.on('data_received', function() {
```

```
console.log('data received succesfully.'); });

// Fire the connection

 eventEmitter.emit('connection');

 console.log("Program Ended.");
```

Q. 15 Write node js application that enables the user to download file and save it locally.

```
var express = require('express');
var app = express();
var PORT = 3000;

app.get('/', function(req, res){
    res.download('Hello.txt');
});

app.listen(PORT, function(err){
    if (err) console.log(err);
    console.log("Server listening on PORT", PORT);
});
```