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

varhttp = 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.

varhttp = 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

varhttp = require('http');

varuc = 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 .

varhttp = require('http');

varfs = require('fs');

http.createServer(function (req, res) {

  fs.readFile('input1.txt', function(err, data) {

    if (err) {

      res.writeHead(404, {'Content-Type':'text/html'});

      returnres.end("404 Not Found");

    }

    res.writeHead(200, {'Content-Type':'text/html'});

    res.write(data);

    returnres.end();

  });

}).listen(8080);

Open browser=>

<http://localhost:8080/>

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

varfs = require('fs');

fs.appendFile('input.txt', 'Hello contentnew!', function (err) {

  if (err) throwerr;

  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.

varfs = require('fs');

fs.readFile('first.txt', function(err, data){

fs.appendFile('second.txt', data, function (err) {

  if (err) throwerr;

  console.log('Saved!');

});

});

Open browser=>

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

varmysql = require('mysql');

varcon = mysql.createConnection({

  host:"127.0.0.1",

  user:"root",

  password:"indira"

});

con.connect(function(err) {

  if (err) throwerr;

  console.log("Connected!");

  con.query("CREATE DATABASE college", function (err, result) {

    if (err) throwerr;

    console.log("Database created");

  });

});

varmysql = require('mysql');

varcon = mysql.createConnection({

  host:"127.0.0.1",

  user:"root",

  password:"indira",

  database:"college"

});

con.connect(function(err) {

  if (err) throwerr;

  console.log("Connected!");

  varsql = "CREATE TABLE student (name VARCHAR(255), class VARCHAR(255))";

  con.query(sql, function (err, result) {

    if (err) throwerr;

    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

varmysql = require('mysql');

varcon = mysql.createConnection({

    host:"127.0.0.1",

    user:"root",

    password:"indira",

  database:"mydb"

});

con.connect(function(err) {

  if (err) throwerr;

  con.query("SELECT \* FROM customers", function (err, result, fields) {

    if (err) throwerr;

    console.log(result);

  });

});

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

varmysql = require('mysql');

varcon = mysql.createConnection({

    host:"127.0.0.1",

    user:"root",

    password:"indira",

    database:"college"

});

con.connect(function(err) {

  if (err) throwerr;

  console.log("Connected!");

  varsql = "INSERT INTO student (name, class) VALUES ('Dip Shaha', 'MSc II'), ('Geet Kale', 'MSc I')";

  con.query(sql, function (err, result) {

    if (err) throwerr;

    console.log(" record inserted");

  });

});

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

varmysql = require('mysql');

varcon = mysql.createConnection({

    host:"127.0.0.1",

    user:"root",

    password:"indira",

  database:"mydb"

});

con.connect(function(err) {

  if (err) throwerr;

  con.query("SELECT \* FROM customers", function (err, result, fields) {

    if (err) throwerr;

    console.log(result);

    varsql = "DELETE FROM customers WHERE address = 'wakad'";

  con.query(sql, function (err, result) {

    if (err) throwerr;

    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 () {

  returnDate();

};

DemoModules.js

varhttp = require('http');

vardt = 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 ​

varevents= require('events');

// Create an eventEmitter object​

 vareventEmitter = newevents.EventEmitter();

// Create an event handler as follows ​

varconnectHandler = functionconnected() { 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.

varexpress = require('express');

varapp = express();

varPORT = 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);

});

<html>

<head>

<script type="text/javascript">

function create\_account()

{

var fn=document.getElementById("fn").value;

var ln=document.getElementById("ln").value;

var emailid=document.getElementById("emailid").value;

var pn=document.getElementById("pn").value;

var age=document.getElementById("age").value;

var pass=document.getElementById("pass").value;

var cpass = document.getElementById("cpass").value;

var letters = /^[A-Za-z]+$/;

var email = /^([a-zA-Z0-9\_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

var phoneno= /^[0-9]+$/;

if(fn=='' || ln=='' || emailid=='' || pn=='' || pass=='' || cpass=='' || age=='')

{

alert("please fill each detail");

}

else if(!letters.test(fn))

{

alert("Enter only alphabets");

}

else if(!letters.test(ln))

{

alert("Enter only alphabets");

}

else if(!email.test(emailid))

{

alert("Enter valid email");

}

else if(!phoneno.test(pn))

{

alert("Enter valid phone");

}

else if(age<18 || age>50)

{

alert("Enter age between 18 and 50");

}

else if(pass != cpass)

{

alert("Enter valid password");

}

else if(document.getElementById("pass").value.length > 12)

{

alert("Password maximum length is 12");

}

else if(document.getElementById("pass").value.length < 6)

{

alert("Password minimum length is 6");

}

else{

alert("Your account has been created successfully...");

window.location="success.html";

}

}

</script>

</head>

<body bgcolor="pink">

<table bgcolor="pink" align="center"/>

<tr>

<td>First Name</td>

<td><input type="text" id="fn"/></td>

</tr>

<tr>

<td>Last Name</td>

<td><input type="text" id="ln"/></td>

</tr>

<tr>

<td>Email\_Id</td>

<td><input type="text" id="emailid"/></td>

</tr>

<tr>

<td>Phone no</td>

<td><input type="text" id="pn"/></td>

</tr>

<tr>

<td>Age</td>

<td><input type="text" id="age"/></td>

</tr>

<tr>

<td>Password</td>

<td><input type="text" id="pass"/></td>

</tr>

<tr>

<td>confirm password</td>

<td><input type="text" id="cpass"/></td>

</tr>

<tr>

<td></td>

<td><input type="submit" value="Submit" onClick="create\_account()"/></td>

</tr>

</table>

</body>

</html>