Mcs practical slips programs

Computer Science (G H Raisoni College of Engineering)

1)Slip no 1:
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.

Solution:

```
<html>
<head>
<script type="text/javascript" src="validateform.js"></script>
<style>
ul {list-style-type:none;}
form{
  background-color: #DCDCDC;
}
</style>
</head>

<body>
<form action="#" name="StudenSignupForm" onsubmit="return(validateHTMlform());">

<div cellpadding="2" width="20%" bgcolor="99FFFF" align="center"
cellspacing="2">

<ul>
<li>
<center><font size=4><b>Student Registration Form</b></font></center>
</li>
</ul>

<ul>
<li>First Name</li>
<li><input type=text name=textnames id="textname" size="30"></li>
</ul>

  <ul>
<li>
  Last Name</li>
<li><input type=text name=lastnames id="lastname" size="30"></li>
</ul>

<ul>
<li>Father Name</li>
<li><input type="text" name="full_father_name" id="fathername"
size="30"></li>
```

```
</ul>


<ul>
<li>Address</li>
<li><input type="text" name="personal_address"
id="personaladdress" size="30"></li>
</ul>

<ul>
<li>Gender</li>
<li><input type="radio" name="sex" value="male" size="10">Male
<input type="radio" name="sex" value="Female" size="10">Female</li>
</ul>

<ul>
<li>City</li>
<li><select name="City">
<option value="-1" selected>select..</option>
<option value="KOLKATA">KOLKATA</option>
<option value="CHENNAI">CHENNAI</option>
<option value="PUNE">PUNE</option>
<option value="JAIPUR">JAIPUR</option>
</select></li>
</ul>

<ul>
<li>Course</li>
<li><select name="Course">
<option value="-1" selected>select..</option>
<option value="B.Tech">B.TECH</option>
<option value="MCA">MCA</option>
<option value="MBA">MBA</option>
<option value="BCA">BCA</option>
</select></li>
</ul>



<ul>
<li>State</li>
<li><select Name="State">
<option value="-1" selected>select..</option>
<option value="New Delhi">NEW DELHI</option>
```

```
<option value="Mumbai">MUMBAI</option>
<option value="Goa">GOA</option>
<option value="Bihar">BIHAR</option>
</select></li>
</ul>

  <ul>
<li>District</li>
<li><select name="Disulict">
<option value="-1" selected>select..</option>
<option value="Nalanda">NALANDA</option>
<option value="UP">UP</option>
<option value="Goa">GOA</option>
<option value="Patna">PATNA</option>
</select></li>

</ul>

<ul>
<li>PinCode</li>
<li><input type="text" name="pin_code" id="pincode" size="30"></li>

</ul>
<ul>
<li>student email</li>
<li><input type="text" name="email_id" id="emailid" size="30"></li>
</ul>

<ul>
<li>Date Of Birth</li>
<li><input type="text" name="date_of_birth" id="dob" size="30"></li>
</ul>

<ul>
<li>Mobile Number</li>
<li><input type="text" name="mobilenumber" id="mobile_no" size="30"></li>
</ul>
<ul>
<li><input type="reset"></li>
<li colspan="2"><input type="submit" value="Submit Form" /></li>
</ul>
</div>
</form>
</body>
```

```
</html>
```

Validateform.js

```
function validateHTMlform()
{
  let form = document.StudenSignupForm;
   if( form.textnames.value == "" )
   {
          alert( "Enter Your First Name!" );
          form.textnames.focus() ;
          return;
   }
   if( form.lastnames.value == "" )
   {
          alert( "Enter Your Last Name!" );
          form.textnames.focus() ;
          return;
   }

   if( form.fathername.value == "" )
   {
          alert( "Enter Your Father Name!" );
          form.fathername.focus() ;
          return;
   }

   if( form.paddress.value == "" )
   {
          alert( "Enter Your Postal Address!" );
          form.paddress.focus() ;
          return;
   }
   if( form.personaladdress.value == "" )
{
          alert( "Enter Your Personal Address!" );
          form.personaladdress.focus() ;
          return;
   }
   if ( ( StudenSignupForm.sex[0].checked == false ) && ( StudenSignupForm.sex[1].checked ==
false ) )
   {
   alert ( "Choose Your Gender: Male or Female" );
   return false;
   }
```

```
  if( form.City.value == "-1" )
  {
        alert( "Enter Your City!" );
        form.City.focus() ;
        return;
  }
  if( form.Course.value == "-1" )
  {
        alert( "Enter Your Course!" );

        return;
  }
  if( form.District.value == "-1" )
  {
        alert( "Select Your District!" );

        return;
  }
  if( form.State.value == "-1" )
  {
        alert( "Select Your State!" );

        return;
  }
  if( form.pincode.value == "" ||
        isNaN( form.pincode.value) ||
        form.pincode.value.length != 6 )
  {
        alert( "Enter your pincode in format ######." );
        form.pincode.focus() ;
        return;
  }
var email = form.emailid.value;
  atpos = email.indexOf("@");
  dotpos = email.lastIndexOf(".");
if (email == "" || atpos < 1 || ( dotpos - atpos < 2 ))
{
        alert("Enter your correct email ID")
        form.emailid.focus() ;
        return;
}
  if( form.dob.value == "" )
  {
        alert( "Enter your DOB!" );
```

```
        form.dob.focus() ;
        return;
    }
   if( form.mobileno.value == "" ||
        isNaN( form.mobileno.value) ||
        form.mobileno.value.length != 10 )
    {
        alert( "Enter your Mobile No. in the format 123." );
        form.mobileno.focus() ;
        return;
    }
    return( true );
}
```

```
ıction validateHTMlform()

:t form = document.StudenSignupForm;
f( form.textnames.value == "" )

  alert( "Enter Your First Name!" );
  form.textnames.focus() ;
  return;

f( form.lastnames.value == "" )

  alert( "Enter Your Last Name!" );
  form.textnames.focus() ;
  return;


f( form.fathername.value == "" )

  alert( "Enter Your Father Name!" );
  form.fathername.focus() ;
  return;


f( form.paddress.value == "" )

  alert( "Enter Your Postal Address!" );
```

```
  form.paddress.focus() ;
  return;

f( form.personaladdress.value == "" )

  alert( "Enter Your Personal Address!" );
  form.personaladdress.focus() ;
  return;

f ( ( StudenSignupForm.sex[0].checked == false ) && ( StudenSignupForm.sex[1].checked
 false ) )

lert ( "Choose Your Gender: Male or Female" );
eturn false;

f( form.City.value == "-1" )

  alert( "Enter Your City!" );
  form.City.focus() ;
  return;

f( form.Course.value == "-1" )

  alert( "Enter Your Course!" );

  return;
```

```
f( form.District.value == "-1" )

  alert( "Select Your District!" );

  return;

f( form.State.value == "-1" )

  alert( "Select Your State!" );

  return;

f( form.pincode.value == "" ||
          isNaN( form.pincode.value) ||
          form.pincode.value.length != 6 )

  alert( "Enter your pincode in format ######." );
  form.pincode.focus() ;
  return;

 email = form.emailid.value;
tpos = email.indexOf("@");
otpos = email.lastIndexOf(".");
email == "" || atpos < 1 || ( dotpos - atpos < 2 ))

  alert("Enter your correct email ID")
  form.emailid.focus() ;
```

```
    return;

  ( form.dob.value == "" )

    alert( "Enter your DOB!" );
    form.dob.focus() ;
    return;

  ( form.mobileno.value == "" ||
            isNaN( form.mobileno.value) ||
            form.mobileno.value.length != 10 )

    alert( "Enter your Mobile No. in the format 123." );
    form.mobileno.focus() ;
    return;

  eturn( true );
```

2) Slip no 2:
Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.

Solution:
employee.html

```
<!DOCTYPE html>
<html lang="en"><head>
<meta charset="utf-8">
<title>JavaScript Form Validation using a sample Employee registration
form</title>
```

```
<meta name="keywords" content="example, JavaScript Form Validation, Sample
registration form" />
<meta name="description" content="This document is an example of
JavaScript Form Validation using a sample registration form. " />
<link rel='stylesheet' href='employee.css' type='text/css' />
<script src="employee.js">
</script>
</head>
<body onload="document.registration.userid.focus();"bgcolor="orange">
<h1>Employee Registration Form</h1>
<form name='registration' onSubmit="return formValidation();">
<ul>
<li><label for="first">First Name:</label></li>
<li><input type="text" name="first" size="50" /></li>
<li><label for="last">Last Name:</label></li>
<li><input type="text" name="last" size="50" /></li>
<li><label for="empid">Employee id:</label></li>
<li><input type="text"name="empid" size="50"/></li>
<li><label for="birth">Birth of date:</label></li>
<li><input type="date" id="birth" name="birth"></li>
<li><label for="address">Address:</label></li>
<li><input type="text" name="address" size="50" /></li>
<li><label for="country">Country:</label></li>
<li><select name="country">
<option selected="" value="Default">(Please select a country)</option>
<option value="AF">Australia</option>
<option value="AL">Canada</option>
<option value="DZ">India</option>
<option value="AS">Russia</option>
<option value="AD">USA</option>
</select></li>
<li><label for="no">Contact no:</label></li>
<li><input type="number" id="" name="no"></li>
<li><label for="jdate">Date of joining:</label></li>
<li><input type="date" id="" name="jdate"></li>
<li><label for="email">Email:</label></li>
<li><input type="text" name="email" size="50" /></li>
<li><label id="gender">Gender:</label></li>
<li><input type="radio" name="male" value="Male" /><span>Male</span></li>
```

```
<li><input type="radio" name="female" value="Female"
/><span>Female</span></li>
<li><label for="salary">salary:</label></li>
<li><input type="number" id="salary" name="salary"></li>
<li><input type="submit" name="submit" value="Submit" /></li>
</ul>
</form>
</body>
</html>
```

employee.js

```
function formValidation()
{
var first=document.registration.first;
var last=document.registration.last;
var empid=document.registration.empid;
var birth=document.registration.birth;
var uadd =document.registration.address;
var ucountry =document.registration.country;
var no=document.registration.no;
var jdate=document.registration.jdate;
var uemail = document.registration.email;
var umgen = document.registration.umgen;
var ufgen = document.registration.ufgen;
var salary =document.registration.salary;
if(allLetter(first))
{
if(allLetter(last))
{
if(alphanumeric(empid))
{
if(allb(birth))
{
if(alphanumeric(uadd))
{
if(countryselect(ucountry))
{
```

```
if(allnumeric(no))
{
if(allnumeric(jdate))
{
if(ValidateEmail(uemail))
{
if(validgendor(umgen,ufgen))
{
if(allnumeric(salary))
{
}}}}}}}}}}
return false;
}
function allLetter(first)
{
var letters = /^[A-Za-z]+$/;
if(first.value.match(letters))
{
alert('employee name submitted');
return true;
}
else
{
alert('employee name must have alphabet characters only');
first.focus();
return false;
}
}
function allLetter(last)
{
var letters = /^[A-Za-z]+$/;
if(last.value.match(letters))
{
    alert("employee name submitted");
return true;
}
else
{
alert('employee name must have alphabet characters only');
last.focus();
```

```
return false;
}
}
function alphanumeric(empid)
{
var letters = /^[0-9a-zA-Z]+$/;
if(empid.value.match(letters))
{
    alert("employee id submitted");
return true;
}
else
{
alert('employee id must have alphanumeric characters only');
uadd.focus();
return false;
}
}
function allb(birth)
{
var birth_len = birth.value.length;
if (birth_len == 0)
{
alert("birth date should not be empty");
birth.focus();
return false;
}
alert("birth of date submitted");
return true;
}
function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
    alert("address submitted");
return true;
}
else
{
```

```
alert('address must have alphanumeric characters only');
uadd.focus();
return false;
}
}
function countryselect(ucountry)
{
if(ucountry.value == "Default")
{
alert('Select your country from the list');
ucountry.focus();
return false;
}
else
{
    alert("country submitted");
return true;
}
}
function allnumeric(no)
{
var number = /^[0-9]+$/;
if(no.value.match(number))
{
        alert("Contact Number submitted");
    return true;
    }
    else
    {
    alert('Contact no must have numeric numbers only');
    no.focus();
    return false;
    }
}
function allnumeric(jdate)
{
var jdate_len = jdate.value.length;
if (jdate_len == 0)
{
alert("date of joining should not be empty");
```

```
birthday.focus();
return false;
}
alert("date of joining submitted");
return true;
}
function ValidateEmail(uemail)
{
var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(uemail.value.match(mailformat))
{
    alert("email address is submitted");
return true;
}
else
{
alert("You have entered an invalid email address!");
uemail.focus();
return false;
}
}
function validgender(umgen,ufgen)
{
x=0;
if(umgen.checked)
{
x++;
} if(ufgen.checked)
{
x++;
}
if(x==0)
{
alert('Select Male/Female');
umgen.focus();
return false;
}
else
{
window.location.reload()
```

```
return true;
}
}
function allnumeric(salary)
{
var sal = /^[0-9]+$/;
if(salary.value.match(sal))
{
alert("salary submitted");
return true;
}
else
{
alert('salry is not submitted');
salary.focus();
return false;
}
}
/*function underAgeValidate(births){
    // it will accept two types of format yyyy-mm-dd and yyyy/mm/dd
    var optimizedBirthday = births.replace(/-/g, "/");

    //set date based on birthday at 01:00:00 hours GMT+0100 (CET)
    var myBirthday = new Date(optimizedBirthday);

    // set current day on 01:00:00 hours GMT+0100 (CET)
    var currentDate = new Date().toJSON().slice(0,10)+' 01:00:00';

    // calculate age comparing current date and borthday
    var myAge = ~~((Date.now(currentDate) - myBirthday) / (31557600000));

    if(myAge < 18)
    {
    alert("age is not validate");
    }
    else
    {
    alert("age is validate");
    }
}*/
```

employee.css

```
h1 {
    margin-left: 70px;
    }
    form li {
    list-style: none;
    margin-bottom: 5px;
    }

    form ul li label{
    float: left;
    clear: left;
    width: 100px;
    text-align: right;
    margin-right: 10px;
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:14px;
    }

    form ul li input, select, span {
    float: left;
    margin-bottom: 10px;
    }

    form textarea {
    float: left;
    width: 350px;
    height: 150px;
    }

    [type="submit"] {
    clear: left;
    margin: 20px 0 0 230px;
    font-size:18px
    }
```

```
p {
margin-left: 70px;
font-weight: bold;
}
```

3) Slip no 3:
Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.

Solution:

Validate email ID using Regular Expression

login.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking email</title>
<link rel='stylesheet' href='login.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()'>
<div class="mail">
<h2><font color="red">   Please Enter your
email</font></h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1'/></li>
<li> </li>
<center><li class="submit"><input type="submit" name="submit"
value="submit"
onclick="ValidateEmail(document.form1.text1)"/></li></center>
<li> </li>
```

```
</ul>
</form>
</div>
<script src="login.js"></script>
</body>
</html>
```

login.js

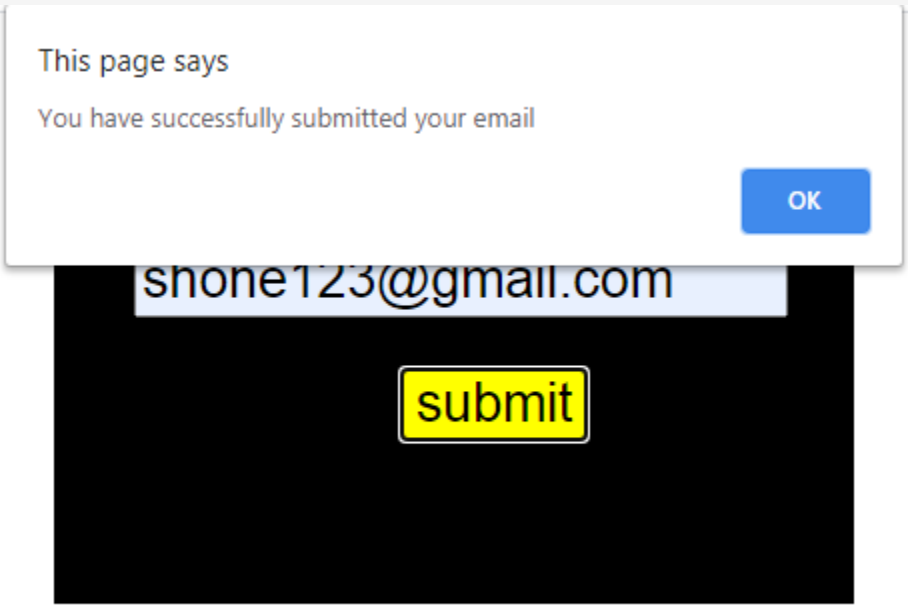
```
function ValidateEmail(inputText)
{
var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(inputText.value.match(mailformat))
{
document.form1.text1.focus();
alert("You have successfully submitted your email");
return true;
}
else
{
alert("You have entered an invalid email address!");
document.form1.text1.focus();
return false;
}
}
```

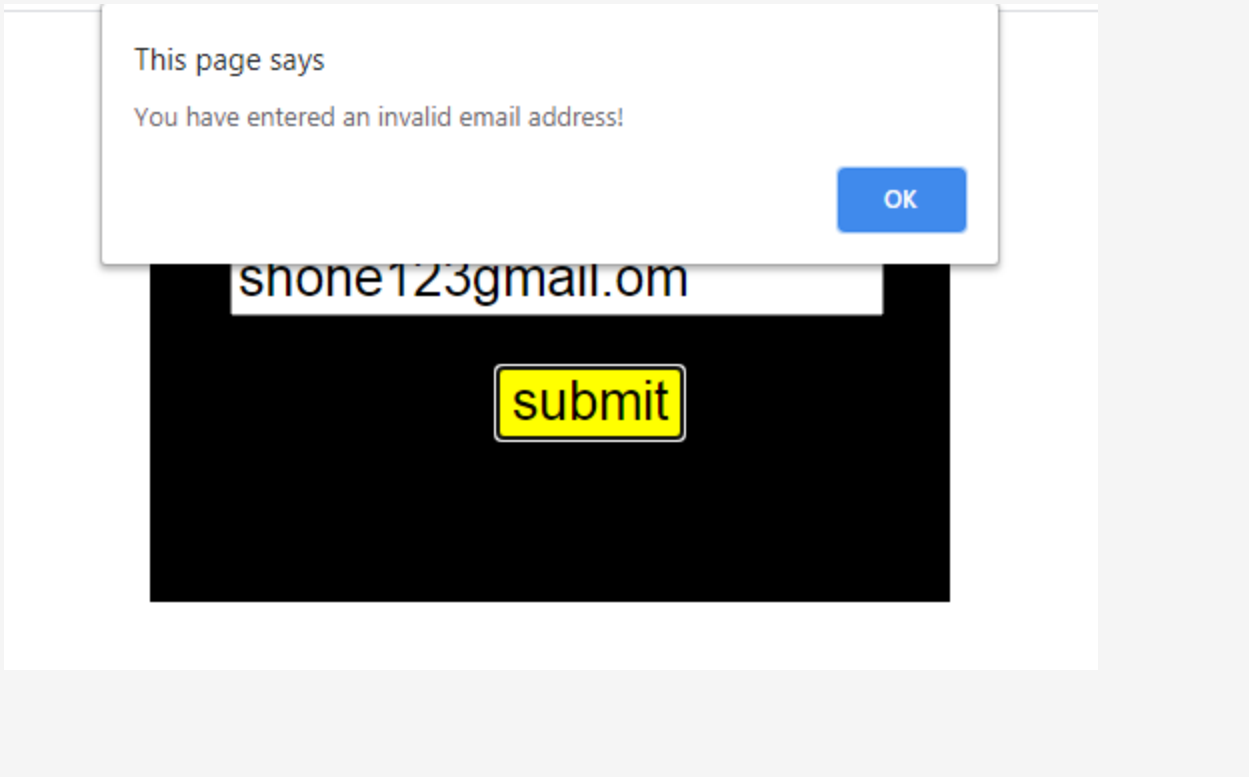
login.css

```
li {list-style-type: none;
    font-size: 16pt;
    }
    .mail {
    margin: auto;
    padding-top: 40px;
    padding-bottom: 40px;
    width: 400px;
    background : black;
    border: 1px soild silver;
    }
```

```
    .mail h2 {
    margin-left: 38px;
    }
    input {
    font-size: 20pt;
    }
    input:focus, textarea:focus{
    background-color: yellow;
    }
    input submit {
    font-size: 12pt;
    }
    .rq {
    color: #FF0000;
    font-size: 10pt;
    }
```

output:





Slip no 4:

Create a Node.js file that will convert the output "Hello World!" into upper-case Letters.

Solution:

```
var http = require('http');
var uc = require('upper-case');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  /*Use our upper-case module to upper case a string:*/
  res.write(uc.upperCase("Hello World!"));
  res.end();
}).listen(8080);
```

Slip no 5:

Using nodejs create a web page to read two file names from user and append contents

of first file into second file.

Solution:

```
// Node.js program to demonstrate the
// fs.appendFile() method

// Import the filesystem module
const fs = require('fs');

// Get the file contents before the append operation
console.log("\nFile Contents of file before append:",
fs.readFileSync("programming.txt", "utf8"));

fs.appendFile("programming.txt", "NOde JS", (err) => {
if (err) {
   console.log(err);
}
else {
   // Get the file contents after the append operation
   console.log("\nFile Contents of file after append:",
   fs.readFileSync("programming.txt", "utf8"));
}
});
```

Slip no 6:

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.

Solution:

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

http.createServer(function (req, res) {
  var q = url.parse(req.url, true);
  var filename = "." + q.pathname;
  fs.readFile(filename, 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);
```

Slip no 7:

Create a Node.js file that writes an HTML form, with an upload field.

Solution:

Step 1: Create an Upload Form

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

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
  res.write('<input type="file" name="filetoupload"><br>');
  res.write('<input type="submit">');
  res.write('</form>');
  return res.end();
}).listen(8080);
```

Step 2: Parse the Uploaded File

Include the Formidable module to be able to parse the uploaded file once it reaches the server.

When the file is uploaded and parsed, it gets placed on a temporary folder on your computer.

Example

The file will be uploaded, and placed on a temporary folder:

```
var http = require('http');
var formidable = require('formidable');
```

```
http.createServer(function (req, res) {
  if (req.url == '/fileupload') {
          var form = new formidable.IncomingForm();
          form.parse(req, function (err, fields, files) {
          res.write('File uploaded');
          res.end();
          });
  } else {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
    res.write('<input type="file" name="filetoupload"><br>');
    res.write('<input type="submit">');
    res.write('</form>');
    return res.end();
  }
}).listen(8080);
```

Slip no 8:

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

Solution:

Creating a Table

To create a table in MySQL, use the "CREATE TABLE" statement.

Make sure you define the name of the database when you create the connection:

Example

Create a table named "customers":

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

var con = mysql.createConnection({
  host: "localhost",
  user: "yourusername",
```

```
  password: "yourpassword",
  database: "mydb"
});

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

Creating a Database

To create a database in MySQL, use the "CREATE DATABASE" statement:

Example

Create a database named "mydb":

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

var con = mysql.createConnection({
  host: "localhost",
  user: "yourusername",
  password: "yourpassword"
});

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

Slip no 9: Create a node.js file that Select all records from the "customers" table, and display the result object on console.

Solution:

```
// Import mysql module
let mysql = require('mysql');

// Setup database connection parameter
let connection = mysql.createConnection({
host: 'localhost',
user: 'root',
password: '',
database: 'mydb2'
});

// Connect with the database
connection.connect(function(e) {
if (e) {

// Show error messaage on failure
return console.error('error: ' + e.message);
}

// Show success message if connected
console.log('\nConnected to the MySQL server...\n');
});

// Set the query message
$query = 'SELECT * from customer1';

// Execute the database query
connection.query($query, function(e, rows) {
if(e){

// Show the error message
console.log("Error ocurred in executing the query.");
return;
}
```

```
/* Display the formatted data retrieved from 'customer' table
using for loop */
console.log("The records of customer table:\n");
console.log("name \t\n");
for(let row of rows) {
 console.log(row['name'],"\t");
}
});

// Close the database connection
connection.end(function(){
console.log('\nConnection closed.\n');
});
```

OR

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

var con = mysql.createConnection({
  host: "localhost",
  user: "myusername",
  password: "mypassword",
  database: "mydb"
});

con.connect(function(err) {
  if (err) throw err;
  //Select all customers and return the result object:
  con.query("SELECT * FROM customers", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
});
```

Slip no 10:

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

Solution:

insert_record.js

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


var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "studentdb"
});


con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO student (rollno,name, percentage) VALUES ?";
var values = [
[1,'abc', 77.6],
[2,'def', 89.6],
[3,'ghi', 91.6]
];
con.query(sql, [values], function (err, result)
 {
    if (err) throw err;
    console.log("Number of records inserted: " + result.affectedRows);
```

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

Slip no 11:
Create a node.js file that Select all records from the "customers" table, and delete the specified record.

Slip no 12:
Create a Simple Web Server using node js.
Solution:

```
var http = require('http'); // 1 - Import Node.js core module

var server = http.createServer(function (req, res) {   // 2 - creating server

   //handle incomming requests here..

});

server.listen(5000); //3 - listen for any incoming requests

console.log('Node.js web server at port 5000 is running..')
```

Slip no 13:
Using node js create a User Login System.

Slip no 14:
Write node js script to interact with the filesystem, and serve a web page from a file .

Slip no 15:
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.

Slip no 16:
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.

Slip no 17:
Write node js application that transfer a file as an attachment on web and enables browser to prompt the user to download file using express js.
Sol:
npm install express

```
var express = require('express');
var app = express();
var PORT = 8080;

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);
});
```

Slip no 18:
Create your Django app in which after running the server, you should see on the browser, the text “Hello! I am learning Django”, which you defined in the index view.

Slip no 19:
Design a Django application that adds web pages with views and templates.

Slip no 26:
Create a Node.js file that demonstrates create database and table in MySQL.

Solution:

```
sudo -i
[sudo] password for pc:srv001
root@pc-HP-Desktop-Pro-G2:~# mysql -u root -p
Enter password:[blank]

mysql> CREATE DATABASE mydb;

mysql> use mydb;
Database changed
mysql> CREATE TABLE book (
        -> id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        -> title VARCHAR(50) NOT NULL,
        -> author VARCHAR(50) NOT NULL,
        -> price int(5));
Query OK, 0 rows affected, 2 warnings (0.65 sec)

mysql> INSERT INTO book values
        -> (NULL,'Learning PHP and MySQL', 'abc', 45),
        -> (NULL,'Learning JQuery', 'pqr', 35),
        -> (NULL,'Angular in Action', 'xyz', 50),
        -> (NULL,'Mastering Laravel', 'lmn', 55);
Query OK, 4 rows affected (0.45 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM book;
+----+-----------------------+-------------+-------+
| id | title                 | author      | price |
+----+-----------------------+-------------+-------+
|  1 | Learning PHP and MySQL | abc |      45 |
|  2 | Learning JQuery       | pqr    |      35 |
|  3 | Angular in Action     |xyz     |      50 |
|  4 | Mastering Laravel     | lmn|  55 |
+----+-----------------------+-------------+-------+
4 rows in set (0.01 sec)

mysql> insert into book values(NULL,'Learning PHP and MySQL', 'abc', 45);
Query OK, 1 row affected (0.16 sec)
```

```
mysql> SELECT * FROM book;
+----+-----------------------+-------------+-------+
| id | title                 | author      | price |
+----+-----------------------+-------------+-------+
|  1 | Learning PHP and MySQL | abc |      45 |
|  2 | Learning JQuery       | pqr    |      35 |
|  3 | Angular in Action     | xyz    |      50 |
|  4 | Mastering Laravel     | lmn | 55 |
|  5 | Learning PHP and MySQL | abc        |         45 |
+----+-----------------------+-------------+-------+
5 rows in set (0.00 sec)

mysql>
```

Connecting to node js program

1) cd nodejs
2) node connect.js

```
Connected to the MySQL server...

The records of book table:

Title                   Author      price

Learning PHP and MySQL          abc     $ 45
Learning JQuery     xyz     $ 35
Angular in Action       pqr     $ 50
Mastering Laravel       lmn     $ 55

Connection closed.
```