

Slip 11- 20

----- Slip 11-----

Q.1) Create a Node.js file that demonstrates create Movie database and table in MySQL. [10 Marks]

```
//npm i mysql

var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "root", // by default root
  password: "", // by default root has no Password
  database: "Movie"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");

  //create database
  con.query("CREATE DATABASE Movie", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });

  //create table
  var sql = "CREATE TABLE Movie (Mname VARCHAR(255), No_ofActors INTEGER)";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });

  //insert record
  var sql = "INSERT INTO Movie (Mname, No_ofActors) VALUES ('ASA', 24)";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });

  //select record
  con.query("SELECT * FROM Movie", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
  con.end();
});
```

Q.2) 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 [20 Marks]

```
hierarchy for storing files =>
// Programs
// : :-public
// :   :-fileDownload.html
// :   :-fileUpload.html
// :   :-uploadDownload.js
```

fileDownload.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Download File</title>
  </head>
  <body>
    <form action="/fileDownload" method="POST"
enctype="multipart/form-data">
      <input type="submit" value="Download">

    </form>
  </body>
</html>
```

fileUpload.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Upload File</title>
  </head>
  <body>
    <form action="/fileUpload" method="POST"
enctype="multipart/form-data">
      <input type="file" placeholder="Choose File" id="fUpload"
name="fUpload">
      <input type="submit" value="Upload">
    </form>
  </body>
</html>
```

UploadDownload.js

```
var express = require('express')
var fs = require('fs')
var path = require('path')
var http = require('http')
```

```

var formidable = require('formidable') // npm i formidable
var bodyParser = require('body-parser')
var newPath;

var app = express();
var server = http.createServer(app);

app.use(bodyParser.urlencoded({extended:false}));
app.use(express.static(path.join(__dirname, './public'))); //__dirname is
current dir name and ./public is dir name of out html folder

app.get('/',function(req,res){ //here html file is opened for getting input
    res.sendFile(path.join(__dirname, './public/fileUpload.html'));
});

//below match action and post 1st arg if those are same then below code get
executed
app.post('/fileUpload',function(req,res){
    var form = new formidable.IncomingForm();
    form.parse(req,function(err,fields,files){

        var oldPath = files.fUpload[0].filepath; //fUpload[0] is id of my
input Tag and zero represents 1st file choosen from multiple files
        console.log("\n oldPath : "+oldPath)

        newPath = 'C:/Users/Mehta/' +files.fUpload[0].originalFilename;
        console.log("\n newPath : "+newPath);

        fs.rename(oldPath,newPath,function(err){
            if(err) throw err;
            else{
                //res.write('file Uploaded and Moved! ');
                //res.end();
            }
        });
        res.sendFile(path.join(__dirname, './public/fileDownload.html'));
    });
});

app.post('/fileDownload',function(req,res){ //downloading same file which we
uploaded it access that file by index so....
    res.download(newPath,function(err){
        res.write("write Succesfully");
    })
});

server.listen(3000,function(){
    console.log("Running on port : 3000");
});

```

----- Slip 12 -----

Q.1) Create a node.js file that Select all records from the "customers" table, and display the result object on console. [10 Marks]

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root", // by default root
  password: "", // by default root has no Password
  database: "customers"
});
con.connect(function(err) {
```

```

if (err) throw err;
console.log("Connected!");

//create database
con.query("CREATE DATABASE customers", function (err, result) {
  if (err) throw err;
  console.log("Database created");
});

//create table
var sql = "CREATE TABLE customer(Cname VARCHAR(255), mno INTEGER, addr VARCHAR(255))";
con.query(sql, function (err, result) {
  if (err) throw err;
  console.log("Table created");
});

//insert record
var sql = "INSERT INTO customer VALUES ('Akash', 7249184378, 'Dighi')";
con.query(sql, function (err, result) {
  if (err) throw err;
  console.log("1 record inserted");
});

//select All record
con.query("SELECT * FROM customer", function (err, result, fields) {
  if (err) throw err;
  console.log(result);
});
con.end();
});

```

Q.2) Create an HTML form for Student Feedback Form with Name, Email ID, Mobile No., feedback (Not good, good, very good, excellent) and write a JavaScript to validate all field using Regular Expression. [20 Marks]

```

<!DOCTYPE html>
<html>
  <body>
    <div>
      <center><h1>Student Feedback Form</h1></center>
      <label>Name:</label> <input type="text" id="name" /><br />
      <label>Email:</label> <input type="email" id="email" /><br />
      <label>Mobile:</label> <input type="text" id="mobile" /><br />
      <label>Feedback:</label>
      <select id="feedback">
        <option value="Not good">Not good</option>
        <option value="Good">Good</option>
      </select>
    </div>
  </body>
</html>

```

```

        <option value="Very good">Very good</option>
        <option value="Excellent">Excellent</option></select>
    <br />
    <button onclick="validateForm()">Submit</button>
</div>

<script>
function validateForm() {
    let name = document.getElementById("name").value;
    let email = document.getElementById("email").value;
    let mobile = document.getElementById("mobile").value;
    let feedback = document.getElementById("feedback").value;

    const namePattern = /^[A-Za-z[ ]*$/;
    const emailPattern =
/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
    const mobilePattern = /^[0-9]{10}$/;

    if (!namePattern.test(name)) {
        alert("Invalid name. Please enter a valid name.");
    } else if (!emailPattern.test(email)) {
        alert("Invalid email address. Please enter a valid email.");
    } else if (!mobilePattern.test(mobile)) {
        alert(
            "Invalid mobile number. Please enter a 10-digit mobile number."
        );
    } else {
        alert("Form submitted successfully!");
    }
}
</script>
</body>
</html>

```

----- Slip 13 -----

Q.1) Create a Node.js file that will convert the output "HELLO WORLD!" into lower-case letters. [10 Marks]

```

var http = require("http")
http.createServer(async function(req,res){
    var msg = 'HELLO WORLD!';
    res.write(msg.toLowerCase())
    res.end()
}).listen(3001)
console.log("running on localhost:3001 ...")

// ..OR console.log("HELLO WORLD!".toLowerCase());

```

Q.2) 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. [20 Marks]

```
<!DOCTYPE html>
<html>
<body>
  <div>
    <center><h1>Student Registration</h1></center>
    <label>First Name:</label> <input type="text" id="firstName" /><br />
    <label>Last Name:</label> <input type="text" id="lastName" /><br />
    <label>Age:</label> <input type="number" id="age"
placeholder="18-50"/><br />
    <button onclick="validateForm()">Register</button>
  </div>

  <script>
    function validateForm() {

      let firstName = document.getElementById("firstName").value.trim();
      let lastName = document.getElementById("lastName").value.trim();
      let age = parseInt(document.getElementById("age").value.trim());

      const namePattern = new RegExp('^[A-Za-z]+$');

      if (namePattern.test(firstName) && namePattern.test(lastName) &&
(!isNaN(age) && age < 51 && age > 17)){
        alert("Form submitted successfully!");
      }
      else
        alert("Check all the details")
    }
  </script>
</body>
</html>
```

----- Slip 14 -----

Q.1) Create a Simple Web Server using node js. [10 Marks]

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

const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World!\n');
}).listen(3001);

console.log('server listening on port 3001');
```

Q.2) Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary. [20 Marks]

```
<!DOCTYPE html>
<html>
  <head>Pge</head>
```



```
<body>
  <div>
    <center><h1>Employee Registration</h1></center>
    Name: <input type="text" id="name" /><br />
    Date of Birth: <input type="date" id="dob" /><br />
    Joining Date: <input type="date" id="joiningDate" /><br />
    Salary: <input type="number" id="salary" /><br />
    <button onclick="validateForm()">Register</button>
  </div>

  <script>
    function validateForm() {

      let name = document.getElementById("name").value.trim();
      let dob = document.getElementById("dob").value.trim();
      let joiningDate = document.getElementById("joiningDate").value.trim();
      let salary = parseInt(document.getElementById("salary").value.trim());

      const datePattern = /^\\d{4}-\\d{2}-\\d{2}$/;

      if (name === "") {
        alert("Please enter a name.");
      } else if (!datePattern.test(dob)) {
        alert("Invalid Date of Birth format");
      } else if (!datePattern.test(joiningDate)) {
        alert("Invalid Joining Date format");
      } else if (isNaN(salary) || salary <= 0) {
        alert("Invalid salary. Please enter a valid positive number.");
      } else {
        alert("Form submitted successfully!");
      }
    }
  </script>
</body>
</html>
```

----- Slip 15 -----

Q.1) Create a node.js file that Select all records from the "students" table, and display the result object on console. [10 Marks]

```
//15a create students DB, Table and insert some record in it.
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root", // by default root
  password: "", // by default root has no Password
  //database: "students"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");

  //create database
  con.query("CREATE DATABASE students", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });

  //create table
```

```

    var sql = "CREATE TABLE stud(rno INTEGER, name VARCHAR(255), mno
INTEGER, addr VARCHAR(255))";
    con.query(sql, function (err, result) {
        if (err) throw err;
        console.log("Table created");
    });

    //insert record
    var sql = "INSERT INTO stud VALUES (3,'aditi', 9579740156 , 'Sant
Nagar')";
    con.query(sql, function (err, result) {
        if (err) throw err;
        console.log("1 record inserted");
    });

    //select All record
    con.query("SELECT * FROM stud", function (err, result, fields) {
        if (err) throw err;
        console.log(result);
    });
    con.end();
});

//1st comment database:students and then run code after that remove comment of
it and comment create Database section

```

Q.2) Create an HTML form for Employee and write a JavaScript to validate name, email ID, mobile number, department, joining date using Regular Expression. [20 Marks]

```

<!DOCTYPE html>
<html>
  <head> </head>
  <body>
    <div>
      <center><h1>Employee Registration</h1></center>
      <input type="text" id="name" placeholder="Name" /><br />
      <input type="text" id="email" placeholder="Email ID" /><br />
      <input type="text" id="mobile" placeholder="Mobile Number" /><br />
      <input type="text" id="department" placeholder="Department" /><br />
      <input type="date" id="joiningDate" /><br />
      <button onclick="validateForm()">Register</button>
    </div>

    <script>
      function validateForm() {
        let name = document.getElementById("name").value;
        let email = document.getElementById("email").value;
        let mobile = document.getElementById("mobile").value;
        let department = document.getElementById("department").value;
        let joiningDate = document.getElementById("joiningDate").value;
      }
    </script>
  </body>
</html>

```

```
const namePattern = /^[A-Za-z]+$/;
const emailPattern =
/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
const mobilePattern = /^[0-9]{10}$/;
const datePattern = /^\d{4}-\d{2}-\d{2}$/;

if (!namePattern.test(name)) {
  alert(
    "Invalid name. Please enter a valid name containing only
alphabets."
  );
} else if (!emailPattern.test(email)) {
  alert("Invalid email address. Please enter a valid email.");
} else if (!mobilePattern.test(mobile)) {
  alert(
    "Invalid mobile number. Please enter a 10-digit mobile number."
  );
} else if (department === "") {
  alert("Please enter a department.");
} else if (!datePattern.test(joiningDate)) {
  alert(
    "Invalid Joining Date format. Please enter a valid date in the
format YYYY-MM-DD."
  );
} else {
  alert("Form submitted successfully!");
  // Additional logic to submit the form can be added here
}
</script>
</body>
</html>
```

----- Slip 16 -----

Q.1) Using node js create a Recipe Book. [10 Marks]

Q.2) 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 call-back function when one of those events is detected. [20 Marks]

```
const event = require('events');
const em = new event.EventEmitter();

em.on('eventOne', () => {
  console.log('Event One occurred');
});

em.on('eventTwo', () => {
  console.log('Event Two occurred');
});

console.log('Main loop started. Listening for events...');

setTimeout(() => {
  em.emit('eventOne');
}, 2000);

setTimeout(() => {
  em.emit('eventTwo');
```

```
}, 4000);

setTimeout(() => {
  em.emit('eventOne');
}, 6000);
```

----- Slip 17 -----

Q.1) Using angular js Create a SPA that show Syllabus content of all subjects of M.Sc (CS) Sem-II (use ng-view) [10 Marks]

```
<!DOCTYPE html>
<html>
  <head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></
script>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular-route.min.
js"></script>
  </head>

  <body >
    <h1>M.Sc. CS Syllabus</h1>

    <div ng-app="mainApp">
      <div style="display: flex; flex-direction: row; justify-content:
space-between; max-width:30%;">
        <p><a href="#fsd">FSD</a></p>

        <p><a href="#daa">DAA</a></p>

        <p><a href="#madt">MADT</a></p>
      </div>

      <div ng-view></div>
```

```

<script type="text/ng-template" id="FSD.html">

    <h2> FSD </h2>

    <p>
        Unit 1: Introduction to HTML <br>
        Unit 2: Introduction to node.js <br>
        Unit 3: Introduction to angular.JS <br>
    </p>

    {{message}}
</script>

<script type="text/ng-template" id="DAA.html">

    <h2> DAA </h2>

    <p>
        Unit 1: Introduction to Data Algorithum <br>
        Unit 2: Greed Method <br>
        Unit 3: Dynamic Programming <br>
    </p>

    {{message}}
</script>

<script type="text/ng-template" id="MADT.html">

    <h2> MADT </h2>

    <p>
        Unit 1: Introduction to Mobile Application <br>
        Unit 2: Android Activity <br>
        Unit 3: Phone Gap <br>
    </p>

    {{message}}
</script>
</div>

<script>
    var mainApp = angular.module("mainApp", ["ngRoute"]);

    mainApp.config([
        "$routeProvider",
        function ($routeProvider) {
            $routeProvider

                .when("/fsd", {
                    templateUrl: "FSD.html",

                    controller: "dspController",

```

```

    })

    .when("/daa", {
        templateUrl: "DAA.html",

        controller: "mlController",
    })

    .when("/madt", {
        templateUrl: "MADT.html",

        controller: "adbsController",
    });
},
]);

mainApp.controller("dspController", function ($scope) {
    $scope.message = "----end----";
});

mainApp.controller("mlController", function ($scope) {
    $scope.message = "----end----";
});

mainApp.controller("adbsController", function ($scope) {
    $scope.message = "----end----";
});
</script>
</body>
</html>

```

Q.2) Using angular js display the 10 student details in Table format (using ng-repeat directive use Array to store data) [20 Marks]

Index.html

```

<!DOCTYPE html>
<html lang="en" ng-app="StudentApp" >
<head>
    <meta charset="UTF-8">
    <title>Student Details</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <script src="app.js"></script>

</head>
<body>

<div ng-controller="StudentController">
    <h2>Student Details</h2>

```



```

<table cellpadding="10px" border="1px">
  <tr >
    <th>Name</th>
    <th>Roll Number</th>
    <th>Grade</th>
  </tr>
  <tr ng-repeat="student in students">
    <td>{{ student.name }}</td>
    <td>{{ student.rollNumber }}</td>
    <td>{{ student.grade }}</td>
  </tr>
</table>
</div>

</body>
</html>

```

app.js

```

var app = angular.module('StudentApp', []);

app.controller('StudentController', function($scope) {
  $scope.students = [
    { name: 'Ajay jaid', rollNumber: '101', grade: 'A+' },
    { name: 'Rushikesh Kashid', rollNumber: '102', grade: 'B+' },
    { name: 'Aniket Galande', rollNumber: '103', grade: 'A-' },
    { name: 'Tushar P', rollNumber: '104', grade: 'B' },
    { name: 'Ishan P', rollNumber: '105', grade: 'B' },
    { name: 'Shubham Morade', rollNumber: '106', grade: 'A' },
    { name: 'sujit Sable', rollNumber: '107', grade: 'B-' },
    { name: 'Rohit Sharma', rollNumber: '108', grade: 'C' },
    { name: 'Virat Kolhi', rollNumber: '109', grade: 'A' },
    { name: 'Messi', rollNumber: '110', grade: 'A-' }
  ];
});

```

Q.1) Using node js create a User Login System [10 Marks]

```
//hierarchy
//  userlogin
//  |  |-public
//  |    |-index.html
//  |    |-login.html
//  |    |-registration.html
//  |  |-app.js
//  |  |-data.js
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    a{font-size: 40px;}
  </style>
</head>
<body align="center">
  <a href="./registration.html">Register</a>
  <br>
  <a href="./login.html">Login</a>
</body>
</html>
```

registration.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    #mylink{
      font-size: 25px;
    }
  </style>
</head>
<body>
  <div>
    <h1>Registration</h1>
    <div>
      <input type="text" value="Name" />
      <input type="password" value="Password" />
      <input type="password" value="Confirm Password" />
      <input type="button" value="Register" />
    </div>
  </div>
</body>
</html>
```

```

    </style>
</head>
<body align="center">
    <header>
        <h1>Register</h1>
    </header>

    <form action="/register" method="POST">
        <fieldset>
            <label>Username</label>
            <input type="text" id="username" name="username"
placeholder="maverick" required>
            <br><br>

            <label>Email ID</label>
            <input type="email" id="email" name="email"
placeholder="abc@example.com" required>
            <br><br>

            <label>Password</label>
            <input type="password" id="password" name="password" required>
            <br><br>

            <button type="reset">Reset</button>
            <button type="submit">Submit</button>
        </fieldset>
    </form>
    <br><br>
    <a id="mylink" href="./login.html">login</a>
</body>
</html>

```

login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #mylink{
            font-size: 25px;
        }
    </style>
</head>
<body align="center">
    <header>
        <h1>Login</h1>
    </header>

    <form action="/login" method="POST">

```

```

    <fieldset>

        <label>Email ID</label>
        <input type ="email" id = 'email' name="email"
placeholder="abc@example.com" required>
        <br><br>

        <label>Password</label>
        <input type="password" id = "password" name="password" required>
        <br><br>

        <button type ="reset">Reset</button>
        <button type ="submit">Submit</button>
    </fieldset>
</form>
<br><br>
<a id="mylink" href="./registration.html">register</a>
</body>
</html>

```

app.js

```

const express=require('express');
const http=require('http')
const bcrypt=require('bcrypt')
const path=require('path')
const bodyParser=require('body-parser')
const users=require('./data').userDB;

const app=express();
const server=http.createServer(app);

app.use(bodyParser.urlencoded({extended:false}));
app.use(express.static(path.join(__dirname,'./public')));

app.get('/',(req,res)=>{
    res.sendFile(path.join(__dirname,'./public/index.html'));
})

app.post('/register', async (req, res) => {
    try{
        let foundUser = users.find((data) => req.body.email === data.email);
        if (!foundUser) {

            let hashPassword = await bcrypt.hash(req.body.password, 10);

            let newUser = {
                id: Date.now(),
                username: req.body.username,
                email: req.body.email,
                password: hashPassword,
            };

```

```

        users.push(newUser);
        console.log('User list', users);

        res.send("<div align = 'center'><h2>Registration  
successful</h2></div><br><br><div align='center'><a  
href='./login.html'>login</a></div><br><br><div align='center'><a  
href='./registration.html'>Register another user</a></div>");
    } else {
        res.send("<div align = 'center'><h2>Email already  
used</h2></div><br><br><div align='center'><a  
href='./registration.html'>Register again</a></div>");
    }
} catch{
    res.send("Internal server error");
}
});

app.post('/login', async (req, res) => {
    try{
        let foundUser = users.find((data) => req.body.email === data.email);
        if (foundUser) {

            let submittedPass = req.body.password;
            let storedPass = foundUser.password;

            const passwordMatch = await bcrypt.compare(submittedPass,
storedPass);
            if (passwordMatch) {
                let username = foundUser.username;
                res.send(`<div align = 'center'><h2>login  
successful</h2></div><br><br><br><div align = 'center'><h3>Hello  
${username}</h3></div><br><br><div align='center'><a  
href='./login.html'>logout</a></div>`);
            } else {
                res.send("<div align = 'center'><h2>Invalid  
password</h2></div><br><br><div align = 'center'><a href='./login.html'>login  
again</a></div>");
            }
        }
        else {
            res.send("<div align = 'center'><h2>Invalid  
email</h2></div><br><br><div align='center'><a href='./login.html'>login  
again<a></div>");
        }
    } catch{
        res.send("Internal server error");
    }
});

server.listen(3000,function(){
    console.log("Server is listening on port:3000");
});

```

data.js

```
const userDB=[];  
module.exports={userDB};
```

Q.2) Create a node.js file that Select all records from the "customers" table, and find the customers whose name starts from 'A'. [20 Marks]

```
var mysql = require('mysql');  
var con = mysql.createConnection({  
  host: "localhost",  
  user: "root", // by default root  
  password: "", // by default root has no Password  
  database: "customers"  
});  
con.connect(function(err) {  
  if (err) throw err;  
  console.log("Connected!");  
  
  //create database  
  con.query("CREATE DATABASE customers", function (err, result) {  
    if (err) throw err;  
    console.log("Database created");  
  });  
  
  //create table  
  var sql = "CREATE TABLE customer(Cname VARCHAR(255), mno INTEGER, addr  
  VARCHAR(255))";  
  con.query(sql, function (err, result) {  
    if (err) throw err;  
    console.log("Table created");  
  });  
  
  //insert record  
  var sql = "INSERT INTO customer VALUES ('Akash', 7249184378, 'Dighi')";  
  con.query(sql, function (err, result) {  
    if (err) throw err;  
    console.log("1 record inserted");  
  });  
  
  //select record  
  con.query("SELECT * FROM customer WHERE Cname LIKE 'A'", function (err,  
  result, fields) {  
    if (err) throw err;  
    console.log("customer whose name start with A are : \n"+result);  
  });  
  con.end();  
});
```

----- Slip 19 -----

Q.1) Create a Node.js file that will convert the output "Hello World!" into upper-case letters. [10 Marks]

```
var http = require("http")
http.createServer(async function(req,res){
  var msg = 'hello world!';
  res.write(msg.toUpperCase())
```

```

        res.end()
    }).listen(3001)
    console.log("running on localhost:3001 ...")

    // ..OR console.log("hello world!".toUpperCase());

```

Q.2) Using angular js create a SPA to accept the details such as name, mobile number, pin code and email address and make validation. Name should contain character only, address should contain SPPU M.Sc. Computer Science Syllabus 2023-24, mobile number should contain only 10 digit, Pin code should contain only 6 digit, email id should contain only one @, . Symbol. [20 Marks]

```

<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <title>AngularJS SPA</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body>

<div ng-controller="FormController">
    <h2>Enter Your Details</h2>
    <form name="myForm">
        <label>Name:</label> <input type="text" ng-model="formData.name"
required pattern="[A-Za-z]+"><br>
        <label>Mobile Number:</label> <input type="text"
ng-model="formData.mobile" required pattern="[0-9]{10}"><br>
        <label>Pin Code:</label> <input type="text"
ng-model="formData.pincod" required pattern="[0-9]{6}"><br>
        <label>Email Address:</label> <input type="email"
ng-model="formData.email" required><br>
        <button ng-click="submitForm()">Submit</button>
    </form>
    <div ng-if="formSubmitted">
        <h3>Form Submitted Successfully!</h3>
        <p>Name: {{formData.name}}</p>
        <p>Mobile Number: {{formData.mobile}}</p>
        <p>Pin Code: {{formData.pincod}}</p>
        <p>Email Address: {{formData.email}}</p>
    </div>
</div>

<script>
var app = angular.module('myApp', []);

```



```

app.controller('FormController', function($scope) {
    $scope.formData = {};
    $scope.formSubmitted = false;

    $scope.submitForm = function() {
        if ($scope.myForm.$valid) {
            if ($scope.formData.name &&
                $scope.formData.name.match(/^[A-Za-z]+$/) &&
                $scope.formData.mobile &&
                $scope.formData.mobile.match(/^[0-9]{10}$/) &&
                $scope.formData.pincode &&
                $scope.formData.pincode.match(/^[0-9]{6}$/) &&
                $scope.formData.email &&
                $scope.formData.email.match(/^[^\s@]+@[^\s@]+\.[^\s@]+$/)) {
                $scope.formSubmitted = true;
            } else {
                alert("Please enter valid details according to the provided requirements.");
            }
        } else {
            alert("Please fill out all the required fields.");
        }
    };
});
</script>

</body>
</html>

```

----- Slip 20 -----

Q.1) Create a Node.js file that demonstrate create student database and table in MySQL [10 Marks]

```

var mysql = require('mysql');
var con = mysql.createConnection({
    host: "localhost",
    user: "root", // by default root
    password: "", // by default root has no Password
    database: "students" //
});
con.connect(function(err) {
    if (err) throw err;
    console.log("Connected!");
});

```

```

//create database
con.query("CREATE DATABASE students", function (err, result) {
    if (err) throw err;
    console.log("Database created");
});

//create table
var sql = "CREATE TABLE stud(rno INTEGER, name VARCHAR(255), mno
INTEGER, addr VARCHAR(255))";
con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
});

//insert record
var sql = "INSERT INTO stud VALUES (4,'ASA', 9579740156 , 'Dighi')";
con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
});

//select All record
con.query("SELECT * FROM stud", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
});
con.end();
});

//1st comment database:students and then run code after that remove comment of
it and comment create Database section

```

Q.2) Using angular js create a SPA to carry out validation for a username entered in a textbox. If the textbox is blank, alert "Enter username". If the number of characters is less than three, alert " Username is too short". If value entered is appropriate the print "Valid username" and password should be minimum 8 characters [20 Marks]

```

<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <title>AngularJS SPA</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body>

<div ng-controller="ValidationController">
    <h2>Username Validation</h2>
    <label>Username:</label> <input type="text" ng-model="username" /><br>

```

```
<label>Password:</label> <input type="password" ng-model="password" /><br>
<button ng-click="validate()">Validate</button>
</div>

<script>
var app = angular.module('myApp', []);

app.controller('ValidationController', function($scope) {
    $scope.validate = function() {
        if (!$scope.username) {
            alert("Enter username.");
        } else if ($scope.username.length < 3) {
            alert("Username is too short.");
        } else if ($scope.password.length < 8 || !$scope.password) {
            alert("Password should be minimum 8 characters.");
        } else {
            alert("Valid username.");
        }
    };
});
</script>

</body>
</html>
```