**Q.1) Create an HTML form for Login and write a JavaScript to validate email ID and Password using Regular Expression.(Slip1)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script>

function Validation(){

var emailid=document.getElementById('email').value;

const emailregex = new RegExp(' ^[a-z0-9][+ @ [a-z]+ \. [a-z] {2,3} \. [a-z] {2,3} $](mailto:+@[a-z]+\.%5ba-z%5d%7b2,3%7d\.%5ba-z%5d%7b2,3%7d$) ');

if(!emailregex.test(emailid))

{

alert("invalid ");

}

else{

alert("valid");

}

}

</script>

</head>

<body>

<label for="email">Email:</label>

<input type="text" id="email"><br><br>

<label for="password">Password:</label>

<input type="password" id="password"><br><br>

<input type="button" name="Submit" onclick="Validation()" value="submit">

</body>

</html>

**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.(Slip1,13)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script>

function Validation(){

var FirstName=document.getElementById('fname').value;

var LastName=document.getElementById('lname').value;

var Age=parseInt(document.getElementById('age').value);

const fregex = /^[A-Za-z]+$/;

const lregex = /^[A-Za-z]+$/;

const agerex = /^(1[8-9]|[2-4][0-9]|50)$/;

var isValid = true;

if(FirstName.trim() === "") {

alert("Please enter First Name");

isValid = false;

}

else if(!fregex.test(FirstName)) {

alert('Please Enter a valid first name');

isValid = false;

}

if(LastName.trim() === "")

{

alert("Please enter Last Name");

isValid = false;

}

else if(!lregex.test(LastName)) {

alert('Please Enter a valid Last name');

isValid = false;

}

if(!agerex.test(Age))

{

alert("Please Enter Age between 18 to 50");

isValid = false;

}

if(isValid)

{

alert('Submitted');

}

}

</script>

</head>

<body>

<h3>Student Registration</h3>

<label for="fname">Enter First Name:</label>

<input type="text" name="fname" id="fname"><br><br>

<label for="lname">Enter Last Name:</label>

<input type="text" name="lname" id="lname"><br><br>

<label for="age">Enter your Age:</label>

<input type="text" name="age" id="age"><br><br>

<input type="button" name="Submit" onclick="Validation()" value="submit">

</body>

</html>

**Q.1) Create a Node.js file that will convert the output "Full Stack!" into reverse string.(Slip2)**

const originalstring ="Full Stack!";

function reverseString(str)

{

return str.split('').reverse().join('');

}

const reversedString=reverseString(originalstring);

console.log(reversedString)

//!kcatS lluF

**Q.2) Using node js create a web page to read two file names from user and append contents of first file into second file.(Slip2)**

**1)html**

<html>

<head>

<title>Sample</title>

<style type="text/css">

fieldset

{

width: 300px;

margin-left: 550px;

margin-top: 200px;

}

</style>

</head>

<body>

<form action="fappendFile15" method="post"><!--js file name-->

<fieldset >

<legend align="center">Append File</legend>

First File Name:<input type="text" placeholder="Enter 1st file name" id="f1" name="f1"><br><br>

Second File Name:<input type="text" placeholder="Enter 2nd file name" id="f2" name="f2"><br><br>

<button onclick="appendfile()">Append Files</button>

</fieldset>

</form>

</body>

</html>

**2) js**

const fs=require('fs');

var http=require('http');

var express=require('express');

var path=require('path');

var bodyparser=require('body-parser');

var app=express();

var server=http.createServer(app);

app.use(bodyparser.urlencoded({extended:false}));

app.use(express.static(path.join (\_\_dirname,'./public'))) //html files saved in public directory

app.get('/' ,function(req,res)

{

res.sendFile(path.join(\_ \_dirname ,' ./public/facceptFile15.html'));

})

app.post('/fappendFile15',function(req,res)//html file name

{

var f1=req.body.f1;

var f2=req.body.f2;

fs.readFile(f1,function(err,data)

{

if (err)

{

console.log(err)

}

fs.appendFile(f2,data,(err)=>

{

if(err)

{

console.log(err);

}

else

{

console.log("\n file content of file after append");

console.log(fs.readFileSync(f2,"utf8"));

}

});

});

})

server.listen(3030,function()

{

console.log("Server listening...");

})

**Q.1) Using node js create a User Login System.(Slip3, 4,16, 18, 24)**

**Public**

**1)index.html**

<html>

<head>

<meta charset="UTF-8">

<title>My Form</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>

**2)Registration.html**

<!DOCTYPE html>

<html lang = "en">

<head>

<meta charset = "UTF-8">

<title> My Form </title>

<style>

#mylink{

font-size: 25px;

}

</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>

**3)login.html**

<!DOCTYPE html>

<html lang = "en">

<head>

<meta charset = "UTF-8">

<title> My Form </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>

**4)data.js**

const userDB=[];

module.exports={userDB};

**5)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 usrname = foundUser.username;

res.send(`<div align ='center'><h2>login successful</h2></div><br><br><br><div align ='center'><h3>Hello ${usrname}</h3></div><br><br><div align='center'><a href='./login.html'>logout</a></div>`); //backtick used

}

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");

})

**Q.2) Create a node.js file that Select all records from the "Teacher" table, and find the Teachers whose salary is greater than 20,000.(Slip 3)**

var mongoose = require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db = mongoose.connection;

db.on('error', function (err)

{

console.error('Connection error:', err.message);

});

db.once('open', function ()

{

console.log("Connection Successful!");

var TeacherSchema = mongoose.Schema(

{

name: String,

Department: String,

salary: Number

});

var Teacher = mongoose.model('Teacher', TeacherSchema, 'teacherInfo');

var teachers = [

{ name: 'Sakshi', Department: 'science', salary: 2000 },

{ name: 'Dikshiya', Department: 'CS', salary: 399999 },

{ name: 'Aditi', Department: 'CA', salary: 467654 }

];

Teacher.collection.insertMany(teachers, function (err, docs)

{

if (err) {

return console.error(err);

}

else

{

console.log(teachers);

var query = Teacher.find({ salary: { $gt: 20000 } });

query.select({ name: 1, salary: 1, \_id: 0 });

let doc = query.exec();

doc.then(function (res)

{

console.log(res);

});

}

});

});

**Q.2) Create an HTML form using AngularJS that contain the Student Registration details and validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50 and display greeting message depending on current time using ng-show (e.g. Good Morning, Good Afternoon, etc.)(Use AJAX).(Slip4)**

<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

<meta charset="UTF-8">

<title>Student Registration</title>

<script src="angular.min.js"></script>

</head>

<body>

<div ng-controller="registrationController">

<h2>Student Registration Form</h2>

<form ng-submit="submit()" name="form">

<label>First Name:</label>

<input type="text" ng-model="student.firstName" required><br><br>

<label>Last Name:</label>

<input type="text" ng-model="student.lastName" required><br><br>

<label>Age:</label>

<input type="text" ng-model="student.age" required><br><br>

<button type="submit">Submit</button>

</form>

<div>

<h3>{{ message }}</h3>

</div>

</div>

<script>

angular.module('myApp', [])

.controller('registrationController', function($scope)

{

$scope.student = {};

$scope.submit = function()

{

if (!isAlphabetic($scope.student.firstName) || !isAlphabetic($scope.student.lastName)) {

alert('First name and last name should contain only alphabetic characters.');

}

else if ($scope.student.age < 18 || $scope.student.age > 50)

{

alert('Age should be between 18 and 50.');

}

Else

{

updateMessage();

}

};

function isAlphabetic(str)

{

return /^[a-zA-Z]\*$/.test(str);

}

function updateMessage()

{

var time = new Date().getHours();

if (time < 12)

{

$scope.message = 'Good Morning';

}

else if (time < 18)

{

$scope.message = 'Good Afternoon';

}

else

{

$scope.message = 'Good Evening';

}

}

});

</script>

</body>

</html>

**Q.1) Create a Node.js file that writes an HTML form, with an upload field.(Slip 5)**

**1)html UploadFile.html**

<html>

<head>

<title>Sample</title>

<style type="text/css">

fieldset

{

width: 300px;

margin-left: 550px;

margin-top: 200px;

}

</style>

</head>

<body>

<form action="uploadFile" method="post" enctype="multipart/form-data"><!--js file name-->

<fieldset >

<legend align="center">Upload File</legend>

<input type="file" id="filetoupload" name="filetoupload"><br><br>

<input type="submit" value="upload">

</fieldset>

</form>

</body>

</html>

**2) js**

const fs=require('fs');

var http=require('http');

var express=require('express');

var path=require('path');

var formidable=require('formidable');

var bodyparser=require('body-parser');

var app=express();

var server=http.createServer(app);

app.use(bodyparser.urlencoded({extended:false}));

app.use(express.static(path.join(\_\_dirname,'./public')))

app.get('/', function(req,res)

{

res.sendFile(path.join(\_\_dirname,' ./public/uploadFile.html '));

})

app.post('/uploadFile ',function(req,res)//js file name

{

var form=new formidable.IncomingForm();

form.parse(req,function(err,fields,files)

{

var oldpath=files.filetoupload[0].filepath;

var newpath= 'C:\Users\Administrator\Desktop\FG216'+files.filetoupload[0].originalFilename;

//your folder path

fs.rename(oldpath,newpath,function(err)

{

if (err) throw err;

res.write("File is uploaded!!");

res.end();

})

})

})

server.listen(3030,function()

{

console.log("Server listening...");

})

**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.(Slip 5,20, 24)**

**Html file**

<!DOCTYPE html>

<html ng-app="myApp">

<head>

<title>Username Validation</title>

<script src=" angular.min.js"></script>

</head>

<body>

<div ng-controller="ValidationController">

<label for="username">Username:</label>

<input type="text" id="username" ng-model="username">

<br>

<label for="password">Password:</label>

<input type="password" id="password" ng-model="password">

<br>

<button ng-click="validateFields()">Submit</button>

</div>

<script>

angular.module ('myApp' , [ ])

**.** controller('ValidationController ', function($scope)

{

$scope.validateFields = function()

{

if (!$scope.username)

{

alert("Enter username");

}

else if ($scope.username.length < 3)

{

alert("Username is too short");

}

else if (!$scope.password || $scope.password.length < 8)

{

alert("Password should be minimum 8 characters");

}

else {

alert("Valid username and password");

}

};

});

</script>

</body>

</html>

**Q.1) Write angular JS by using ng-click directive to display an alert message after clicking the element.(slip 6)**

**Html**

<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Click Alert</title>

<script src="angular.min.js"></script>

</head>

<body ng-controller="ClickController">

<button ng-click="showAlert()">Click me</button>

<script>

angular.module('myApp', [])

**.** controller('ClickController', function($scope)

{

$scope.showAlert = function()

{

alert('Button clicked!');

};

});

</script>

</body>

</html>

**Q.2) 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.(Slip6 , 9) js file**

var fs=require('fs');

var http=require('http');

http.createServer(function(req,res)

{

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

fs.readFile('a.html',function(err,data)

{

if(err)

{

res.write("<h1>404 Page not found</h1>");

res.end();

throw err;

}

else

{

res.write(data);

res.end();

}

});

}).listen(3030)

console.log("end");

**Q.1) Create angular JS Application that show the current Date and Time of the System (Use Interval Service)(Slip 7)**

**Html**

<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Current Date and Time</title>

<script src="angular.min.js"></script>

</head>

<body ng-controller="DateTimeController">

<h1>Current Date and Time</h1>

<h3>{{ currentDateTime }}</h3>

<script>

angular.module('myApp', [])

**.**controller('DateTimeController', function($scope, $interval)

{

$interval(function()

{

$scope . currentDateTime = new Date().toLocaleString();

}, 1000);

});

</script>

</body>

</html>

**Q.2) Create a node 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 7, 16, 22,25)**

var events=require('events')

var em=new events.EventEmitter();

function add(a,b)

{

console.log("Addition: "+(a+b));

}

function sub(a,b)

{

console.log("subtraction: "+(a-b));

}

function mul(a,b)

{

console.log("Multiplication: "+(a\*b));

}

function div(a,b)

{

console.log("Division: "+(a/b));

}

em.on('arithmetic',add);

em.on('arithmetic',mul);

em.on('arithmetic',sub);

em.on('arithmetic',div);

em.emit('arithmetic',12,2);

/\*Addition: 14

Multiplication: 24

subtraction: 10

Division: 6\*/

**Q.1) Create a Simple Web Server using node js(Slip8, Slip14)**

**Js**

var http = require('http');

var server = http.createServer(function (req, res)

{

res.write('Hello World!');

res.end();

});

server.listen(8080);

console.log("Done");

**Q.2) Using angular js display the 10 student details in Table format (using ng-repeat directive use Array to store data)(Slip8, 17)**

**Html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Student Details</title>

<script src=" angular.min.js"></script>

<style>

Body

{

margin: 24;

font-size: 120%;

}

th, td

{

padding: 20px;

}

</style>

</head>

<body ng-app="myApp" ng-controller="ListController">

<h1>Students Details</h1>

<table border="1">

<thead>

<tr>

<th>Roll no</th>

<th>Name</th>

<th>Class</th>

<th>Mobile No</th>

<th>Address</th>

</tr>

</thead>

<tr ng-repeat="stud in student">

<td>{{stud.rno}}</td>

<td>{{stud.name}}</td>

<td>{{stud.class}}</td>

<td>{{stud.mno}}</td>

<td>{{stud.address}}</td>

</tr>

</table>

<script>

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

app.controller('ListController', function($scope)

{

$scope.student = [{

rno: 1,

name: 'Radhika',

class: 'MCA',

mno: 5276575507,

address: 'Delhi'

}, {

rno: 2,

name: 'Sakshi',

class: 'MCS',

mno: 4276575507,

address: 'Chakan'

}, {

rno: 3,

name: 'Nandita',

class: 'MCA',

mno: 2276575507,

address: 'Alandi'

}, {

rno: 4,

name: 'Swejal',

class: 'MCA',

mno: 3276575507,

address: 'Delhi'

}, {

rno: 5,

name: 'Mansi',

class: 'MSc(cs)',

mno: 1652146437,

address: 'Pune'

}, {

rno: 6,

name: 'Raj',

class: 'MSc',

mno: 1652146837,

address: 'Pune'

}, {

rno: 7,

name: 'Riya',

class: 'MSc(CA)',

mno: 1652146437,

address: 'Pune'

}, {

rno: 8,

name: 'Priti',

class: 'MCA',

mno: 1652146437,

address: 'Pune'

}, {

rno: 9,

name: 'Spruha',

class: 'MCA',

mno: 1652146437,

address: 'Pune'

}, {

rno: 10,

name: 'Neha',

class: 'MSc',

mno: 1652146437,

address: 'Pune'

}];

});

</script>

</body>

</html>

**Q.1) Create a Node.js file that writes an HTML form, with a concatenate two string.(Slip9)**

**Html**

<!DOCTYPE html>

<html>

<head>

<title>String Concatenation</title>

</head>

<body>

<h1>String Concatenation</h1>

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

<label for="string1">String 1:</label>

<input type="text" id="string1" name="string1"><br><br>

<label for="string2">String 2:</label>

<input type="text" id="string2" name="string2"><br><br>

<button type="submit">Submit</button>

</form>

</body>

</html>

**2) js**

const express = require('express');

const qs = require('querystring');

const http = require('http');

const app =express();

const server=http.createServer(app);

app.get('/', (req, res) =>

{

res.sendFile(\_\_dirname + '/Slip9\_1.html');

});

app.post('/', (req, res) =>

{

let body = ' ' ;

req.on('data', chunk => body += chunk .toString());

req.on('end', () =>

{

const data = qs.parse(body);

const result = data.string1 + data.string2;

res.send(`<h1>Concatenated String:</h1><p>${result}</p>`); //backtick

});

});

server.listen(3333,function()

{

console.log("server is listening on port:3333")

})

**Q.1) Create a Node.js file that demonstrate create college database and table in MySQL(Slip 10)**

var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function()

{

console.log("Connection Successful!");

var CollegeSchema=mongoose.Schema(

{

name:String,

Address:String,

Establish:Number,

Course:[String]

});

var clg = mongoose.model('College', CollegeSchema,'collegeInfo');

var college = [{ name: 'MIT ACSC',Address:'Alandi',Establish:2000,Course:['Computer Science','Computer Application']}];

clg.collection.insertMany(college, function (err, docs)

{

if(err)

{

return console.error(err);

}

else

{

console.log(college);

}

let query= clg.find();

query.select({name:1,Address:1,Establish:1,\_id:0})

const doc=query.exec();

doc.then(function(res)

{

console.log(res);

});

})

});

**Q.2) 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 10, 23)**

**1)module.js**

module.exports=function printDate()

{

var d1=new Date();

return (d1);

}

**2)date.js**

var http=require('http');

var req=require('./date\_module.js');

d=req();

http.createServer(function(request,respose)

{

respose.writeHead(200,{'content-type':'text/html'});

respose.write(d.toString());

respose.end();

}).listen(3030);

Output:

//Sun Apr 07 2024 16:18:25 GMT+0530 (India Standard

Time)...Shows on browser

**Q.1) Create a Node.js file that demonstrates create Movie database and table in MySQL.(Slip 11, 21)**

var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function()

{

console.log("Connection Successful!");

var movieSchema =mongoose.Schema(

{

title: String,

director: String,

year: Number,

genre: [String]

});

var Movie = mongoose.model('Movie', movieSchema,'moviestore');

var newMovie = [{title: "Inception",director: "Christopher Nolan",year: 2010,genre: ["Sci-Fi", "Action", "Thriller"]},

{title: "Rocketry",director: "ABC",year: 2023,genre: ["Sci-Fi", "Real incident"]}];

Movie.collection.insertMany(newMovie,function(err,docs)

{

if(err)

{

return console.error(err);

} else

{

console.log(newMovie);

}

});

});

**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(Slip11,21)**

**1)uploadFile.html(public folder)**

<html>

<head>

<title>Sample</title>

<style type="text/css">

fieldset

{

width: 300px;

margin-left: 550px;

margin-top: 200px;

}

</style>

</head>

<body>

<form action="uploadFile" method="post" enctype="multipart/form-data">

<!—action:link is given -->

<fieldset >

<legend align="center">Upload File</legend>

<input type="file" id="filetoupload" name="filetoupload"><br><br>

<input type="submit" value="upload">

</fieldset>

</form>

</body>

</html>

**2)DownloadFile.html (public folder)**

<html>

<head>

<title>Download file</title>

<style type="text/css">

fieldset

{

width: 300px;

margin-left: 550px;

margin-top: 200px;

}

</style>

</head>

<body>

<form action="downloadFile" method="post" enctype="multipart/form-data>

<fieldset >

<legend align="center">Download File</legend>

<input type="submit" value="Download">

</fieldset>

</form>

</body>

</html>

**3)DownloadFile.js**

const fs=require('fs');

var http=require('http');

var express=require('express');

var path=require('path');

var formidable=require('formidable');

var bodyparser=require('body-parser');

var app=express();

var newpath;

var server=http.createServer(app);

app.use( bodyparser.urlencoded( {extended:false} ) );

app.use(express.static(path.join(\_\_dirname,'./public')))//html files saved in public directory

app.get('/',function(req,res)

{

res.sendFile(path.join(\_\_dirname,'./public/uploadFile.html'));

})

app.post('/uploadFile',function(req,res)//js file name

{

var form=new formidable.IncomingForm();

form.parse(req,function(err,fields,files)

{

var oldpath=files.filetoupload[0].filepath;

newpath='C:\Users\Administrator\Desktop\FG216'+files.filetoupload[0].originalFilename;//y

our folder path

fs.rename(oldpath,newpath,function(err)

{

if (err) throw err;

else

{

res.sendFile(path.join(\_\_dirname,'./public/downloadFile.html'));

}

});

});

});

app.post('/downloadFile',function(req,res)

{

});

res.download(newpath,function(err)

{

});

res.end("Done with download...");

server.listen(3030,function(){

console.log("Server listening...");

})

**Q.1) Create a node.js file that Select all records from the "customers" table, and display the result object on console.(Slip12)**

var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function()

{

console.log("Connection Successful!");

var CustomerSchema=mongoose.Schema(

{

CustomerID:Number,

CustomerName:String,

Address:String,

});

var Customer = mongoose.model('Customer',CustomerSchema,'Customerstore');

var custs=[{CustomerID:7,CustomerName:"Akshay",Address:"Dhayari"},

{CustomerID:8,CustomerName:"Payal",Address:"Alandi"}];

Customer.collection.insertMany(custs,function(err,docs)

{

if(err)

{

return console.error(err);

}

else

{

console.log(custs);

}

let query= Customer.find();

query.select({CustomerID:1,CustomerName:1,Address:1,\_id:0})

const doc=query.exec();

doc.then(function(res)

{

console.log(res);

});

})

})

**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.(Slip 12)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Student Feedback Form</title>

<style>

.error {

color: red;

}

</style>

</head>

<body>

<h2>Student Feedback Form</h2>

<form id="feedbackForm" onsubmit="return validateForm()">

<label for="name">Name:</label><br>

<input type="text" id="name" name="name" required><br>

<span id="nameError" class="error"></span><br>

<label for="email">Email ID:</label><br>

<input type="email" id="email" name="email" required><br>

<span id="emailError" class="error"></span><br>

<label for="mobile">Mobile No.:</label><br>

<input type="tel" id="mobile" name="mobile" pattern="[0-9]{10}" required><br>

<span id="mobileError" class="error"></span><br>

<label for="feedback">Feedback:</label><br>

<select id="feedback" name="feedback" required>

<option value="">Select</option>

<option value="Not good">Not good</option>

<option value="Good">Good</option>

<option value="Very good">Very good</option>

<option value="Excellent">Excellent</option>

</select><br>

<span id="feedbackError" class="error"></span><br>

<input type="submit" value="Submit">

</form>

<script>

function validateForm()

{

const name = document.getElementById('name').value.trim();

const email = document.getElementById('email').value.trim();

const mobile = document.getElementById('mobile').value.trim();

const feedback = document.getElementById('feedback').value.trim();

const namePattern = /^[a-zA-Z\s]+$/;

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

const mobilePattern = /^[0-9]{10}$/;

let valid = true;

if (!namePattern.test(name))

{

document.getElementById('nameError').innerText = 'Invalid name';

valid = false;

} else {

document.getElementById('nameError').innerText = ' ';

}

if (!emailPattern.test(email))

{

document.getElementById('emailError').innerText = 'Invalid email';

valid = false;

}

else

{

document.getElementById('emailError').innerText = ' ';

}

if (!mobilePattern.test(mobile))

{

document.getElementById('mobileError').innerText = 'Invalid mobile number';

valid = false;

} else {

document.getElementById('mobileError').innerText = '';

}

if (feedback === '')

{

document.getElementById('feedbackError').innerText = 'Please select a feedback';

valid = false;

}

else {

document.getElementById('feedbackError').innerText = ' ';

}

return valid;

}

</script>

</body>

</html>

**Q.1) Create a Node.js file that will convert the output "HELLO WORLD!" into lower-case letters.(Slip 13,19)**

**Js**

const inputString = "Hello World!";

const lowercaseString = inputString.toLowerCase();

console.log("Lowercase:", lowercaseString);

const uppercaseString = inputString.toUpperCase();

console.log("Uppercase:", uppercaseString);

**Q.2) Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.(Slip15)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Employee Form</title>

<style>

.error

{

color: red;

}

</style>

</head>

<body>

<h2>Employee Form</h2>

<form id="employeeForm" onsubmit="return validateForm()">

<label for="name">Name:</label><br>

<input type="text" id="name" name="name" required><br>

<span id="nameError" class="error"></span><br>

<label for="email">Email ID:</label><br>

<input type="email" id="email" name="email" required><br>

<span id="emailError" class="error"></span><br>

<label for="mobile">Mobile Number:</label><br>

<input type="tel" id="mobile" name="mobile" pattern="[0-9]{10}" required><br>

<span id="mobileError" class="error"></span><br>

<label for="department">Department:</label><br>

<input type="text" id="department" name="department" required><br>

<span id="departmentError" class="error"></span><br>

<label for="joiningDate">Joining Date:</label><br>

<input type="date" id="joiningDate" name="joiningDate" required><br>

<span id="joiningDateError" class="error"></span><br>

<input type="submit" value="Submit">

</form>

<script>

function validateForm()

{

const name = document.getElementById('name').value.trim();

const email = document.getElementById('email').value.trim();

const mobile = document.getElementById('mobile').value.trim();

const department = document.getElementById('department').value.trim();

const joiningDate = document.getElementById('joiningDate').value.trim();

const namePattern = / ^[a-zA-Z\s] +$ /;

const emailPattern = / ^ [ ^ \s@]+@ [ ^ \ s@]+\ **.** [^\ s @] +$ /;

const mobilePattern = /^[0-9]{10} $/;

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

let valid = true;

if (!namePattern.test(name))

{

document.getElementById('nameError').innerText = 'Invalid name';

valid = false;

}

else

{

document.getElementById('nameError').innerText = ' ';

}

if (!emailPattern.test(email))

{

document.getElementById('emailError').innerText = 'Invalid email';

valid = false;

}

Else

{

document.getElementById('emailError').innerText = ' ';

}

if (!mobilePattern.test(mobile))

{

document.getElementById('mobileError').innerText = 'Invalid mobile number';

valid = false;

}

else

{

document.getElementById('mobileError').innerText = ' ';

}

if (department === '')

{

document.getElementById('departmentError').innerText = 'Department cannot be empty';

valid = false;

}

else {

document.getElementById('departmentError').innerText = '';

}

if (!datePattern.test(joiningDate))

{

document.getElementById('joiningDateError').innerText = 'Invalid joining date';

valid = false;

} else {

document.getElementById('joiningDateError').innerText = '';

}

return valid;

}

</script>

</body>

</html>

**Q.1) Create a node.js file that Select all records from the "students" table, and display the result object on console.(Slip15, 20)**

var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function(){

console.log("Connection Successful!");

var StudentSchema=mongoose.Schema(

{

Studentrollno:Number,

StudentName:String,

Class:String,

Age:Number

});

var Student = mongoose.model('Student',StudentSchema,'Studentstore');

var studs= [{Studentrollno:1,StudentName:"Akshay",Class:"FYMSC",Age:24},

{Studentrollno:2,StudentName:"Kajal",Class:"FYMSC",Age:24}];

Student.collection.insertMany(studs,function(err,docs)

{

if(err)

{

return console.error(err);

}

else

{

console.log(studs);

}

let query= Student.find();

query.select({Studentrollno:1,StudentName:1,StudentName:1,Age:1,\_id:0})

const doc=query.exec();

doc.then(function(res)

{

console.log(res);

});

})

})

**Q.1) Using angular js Create a SPA that show Syllabus content of all subjects of M.Sc (CS) Sem-II (use ng-view)(Slip 17)**

**Copy-paste angular.min.js file in your folder**

<!DOCTYPE html>

<html>

<head>

<script src="angular.min.js"></script>

</head>

<body>

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

<div ng-app="mainApp" ng-controller="MainController">

<div style="display: flex; flex-direction: row; justify-content: space-between; max-width:30%;">

<p><a href="#" ng-click="navigateTo('fsd')">FSD</a></p>

<p><a href="#" ng-click="navigateTo('daa')">DAA</a></p>

<p><a href="#" ng-click="navigateTo('madt')">MADT</a></p>

</div>

<div ng-include="currentTemplate"></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", []);

mainApp.controller("MainController", function ($scope, $location)

{

$scope.navigateTo = function (route)

{

switch (route)

{

case 'fsd':

$scope.currentTemplate = 'FSD.html';

break;

case 'daa':

$scope.currentTemplate = 'DAA.html';

break;

case 'madt':

$scope.currentTemplate = 'MADT.html';

break;

default:

$scope.currentTemplate = ' ';

break;

}

};

});

</script>

</body>

</html>

**Q.2) Create a node.js file that Select all records from the "customers" table, and find the customers whose name starts from ‘A’.(Slip18)**

var mongoose=require('mongoose');

mongoose.connect('mongodb://0.0.0.0:27017/');

var db=mongoose.connection;

db.on('error',console.error.bind(console,'connection error'));

db.once('open',function(){

console.log("Connection Successful!");

var CustomerSchema=mongoose.Schema({

CustomerID:Number,

CustomerName:String,

Address:String,

});

var Customer = mongoose.model('Customer',CustomerSchema,'Customerstore');

var custs=[{CustomerID:7,CustomerName:"Akshay",Address:"Dhayari"},

{CustomerID:8,CustomerName:"Payal",Address:"Alandi"}];

Customer.collection.insertMany(custs,function(err,docs)

{

if(err)

{

return console.error(err);

}

let query= Customer.find({CustomerName : /^A/});

query.select({CustomerID:1,CustomerName:1,Address:1,\_id:0})

const doc=query.exec();

doc.then(function(res)

{

console.log(res);

})

})

})

**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.** **(Slip19)**

<!DOCTYPE html>

<html ng-app="validationApp">

<head>

<title>Details Validation</title>

<script src="angular.min.js"></script>

<script>

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

app.controller('ValidationController', function($scope)

{

$scope.submitForm = function()

{

if ($scope.validateForm())

{

alert('Form submitted successfully!');

}

};

$scope.validateForm = function()

{

if (!/^[a-zA-Z\s]+$/.test($scope.name))

{

alert('Name should contain characters only.');

return false;

}

if (!/^\d{10}$/.test($scope.mobileNumber))

{

alert('Mobile number should contain 10 digits only.');

return false;

}

if (!/^\d{6}$/.test($scope.pinCode))

{

alert('Pin code should contain 6 digits only.');

return false;

}

if (!/^[^\s@][+@[^\s@]+\.[^\s@]+$/.test($scope.email)](mailto:+@[%5e\s@%5d+\.%5b%5e\s@%5d+$/.test($scope.email)))

{

alert('Invalid email address.');

return false;

}

return true;

};

});

</script>

</head>

<body ng-controller="ValidationController">

<h2>Enter Details:</h2>

<form ng-submit="submitForm()">

<label>Name:</label>

<input type="text" ng-model="name" required><br>

<label>Mobile Number:</label>

<input type="tel" ng-model="mobileNumber" required><br>

<label>Pin Code:</label>

<input type="text" ng-model="pinCode" required><br>

<label>Email Address:</label>

<input type="email" ng-model="email" required><br>

<button type="submit">Submit</button>

</form>

</body>

</html>

**Q.1) Using node js create an Employee Registration Form validation.(Slip22)**

EmployeeRegistration(folder)

-app.js

-data.js

-public(folder under EmployeeRegistration)

-index.html

-login.html

-registration.html

**1)data.js**

const userDB=[];

module.exports={userDB};

**2)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);

newUser =

{

};

id: Date.now(),

username: req.body.username,

date:req.body.date,

addr:req.body.addr,

no:req.body.no,

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 usrname = foundUser.username;

res.send(`<div align ='center'><h2>login

successful</h2></div><br><br><br><div align

='center'><h3>Hello ${usrname}</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");

**3)index.html**

<html>

<head>

<meta charset="UTF-8">

<title>My Form</title>

<style>

a{

}

font-size: 40px;

</style>

</head>

<body align='center'>

<form>

<fieldset>

<h1>Employee Registration</h1>

<a href="./registration.html">Register</a>

<br>

<a href="./login.html">Login</a>

</fieldset>

</form>

</body>

</html>

**4)registration.html**

<!DOCTYPE html>

<html lang = "en">

<head>

<meta charset = "UTF-8">

<title> My Form </title>

<style>

#mylink{

font-size: 25px;

}

</style>

</head>

<body align='center'>

<header>

<h1>Employee Registration </h1>

</header>

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

<fieldset>

<label>Employee Name</label>

<input type ="text" id = 'username'

name="username" placeholder="maverick" required>

<br><br>

<label>Date Of birth</label>

<input type ="date" id = 'date' name="date"

required>

<br><br>

<label>Address</label>

<input type ="text" id = 'addr' name="addr"

placeholder="abc" required>

<br><br>

<label>Contact Number</label>

<input type ="text" id = 'no' name="no"

placeholder="+91" 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>

**5)login.html**

<!DOCTYPE html>

<html lang = "en">

<head>

<meta charset = "UTF-8">

<title> My Form </title>

<style>

#mylink{

font-size: 25px;

}

</style>

</head>

<body align='center'>

<header>

<h1>Employee 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>

**Q.1) Write node js script to interact with the file system, and serve a web page from a File(Slip23)**

1)data.html

<html>

<head>

<title>Sample</title>

<style type="text/css">

fieldset

{

width: 300px;

margin-left: 550px;

}

</style>

</head>

<body>

<form>

<fieldset >

<legend align="center">Registration Form</legend>

Student Name:<input type="text" placeholder="Enter your name"><br><br>

Class:<select>

<option value="Fymsc">FYMSC</option>

<option value="Fymsc">SYMSC</option>

<option value="Fymsc">TYMSC</option>

</select><br><br>

age:<input type="number" placeholder="Enter age"><br><br>

address:<textarea rows="5" cols="20">here</textarea><br><br>

<input type="submit" value="submit">&nbsp;

<input type="reset" value="clear"

</fieldset>

</form>

</body>

</html>

**2)data.js**

var http=require('http');

var fs=require('fs');

var data=fs.readFileSync('data.html');

http.createServer(function(req,res)

{

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

res.write(data);

res.end();

}).listen(3030)

console.log("end");

**Q.1) Create an angular JS Application that shows the location of the current web page.(slip25)**

<!DOCTYPE html>

<html ng-app="locationApp">

<head>

<title>Current Page Location</title>

<script src="angular.min.js"></script>

<script>

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

locationApp.controller('mainController', function ($scope,

$location)

{

$scope.currentLocation = $location.absUrl();

});

</script>

</head>

<body ng-controller="mainController">

<div>

<h2>Current Page Location</h2>

<p>The current page location is: {{ currentLocation }}</p>

</div>

</body>

</html>

<!--The current page location is:

<file:///C:/Users/hp/OneDrive/Desktop/JS/Slip25-1.html#!/-->>

**Q.2) Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.(Slip14)**

<html>

<head>

<script type="text/javascript">

function display()

{

let first=document.getElementById("i1").value;

let last=document.getElementById("i2").value;

let age=document.getElementById("a").value;

let dob=document.getElementById("dob").value;

let jd=document.getElementById("jd").value;

let salary=document.getElementById("s").value;

let dobpatt=/^\d{4}-\d{2}-\d{2}$/;

let salpatt=/^\d[0-9]+$/;

if(dob.match(dobpatt) && jd.match(dobpatt) && salary.match(salpatt))

{

document.write("First Name:"+first+"<br>");

document.write("LastFirst Name:"+last+"<br>");

document.write("Age:"+age+"<br>");

document.write("Date of birth:"+dob+"<br>");

document.write("Date of joining:"+jd+"<br>");

document.write("Salary:"+salary+"<br>");

}

else

{

alert("Please enter valid data!!!");

}

}

</script>

</head>

<body>

<h4>First Name:</h4><input type="text" placeholder="Enter first name" id="i1">

<h4>Last Name:</h4><input type="text" placeholder="Enter last name" id="i2">

<h4>Age:</h4><input type="text" placeholder="Enter Age" id="a">

<h4>Date Of birth:</h4><input type="date" placeholder="Enter Birth Date" id="dob">

<h4>Joining Date:</h4><input type="date" placeholder="Enter Joining date" id="jd">

<h4>Salary:</h4><input type="text" placeholder="Enter Salary" id="s">

<button onclick="display()">submit</button>

</body>

</html>