------------------------ Slip 1 -------------------

Q.1) Create an HTML form for Login and write a JavaScript to validate email ID and Password using Regular Expression.
[10 Marks]

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]

---------------------- Slip 2 ------------------

Q.1) Create a Node.js file that will convert the output "Full Stack!" into reverse string.
[10 Marks]

```
var http = require('http');
var uc = require('reverse');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  /*Use our reverse module to reverse a string:*/
  res.write(uc.reverse("Hello World!"));
  res.end();
}).listen(8081);
```

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.
[20 Marks]

Node.js
```
//ass 15
// Using nodejs create a web page to read two file names
from user and append contents of first file into second file
```

```javascript
// 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", "Welcome to 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"));
}
});
```

Programming.txt
node

------------------------ Slip 6 -------------------

Q.1) Write angular JS by using ng-click directive to display an alert message after clicking the Element.   [10 Marks]

```html
<html lang="en" ng-app="clickApp">
<head>
    <title>AngularJS Click Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="clickController">

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

<script>
    angular.module('clickApp', [])
    .controller('clickController', ($scope) =>
$scope.showAlert = () => alert('Button Clicked!'));
```

```
</script>

</body>
</html>
```

```
<!--   We define an AngularJS module named 'clickApp'.
    We create a controller named 'clickController' within
the module.
    Inside the controller, we define a function
$scope.showAlert that displays an alert when called.
    The button in the HTML has the ng-click directive
set to showAlert(), meaning that the showAlert function
will be executed when the button is clicked.

When you open this HTML file in a browser and click the
"Click Me" button, it will trigger the showAlert function,
and you'll see an alert with the message "Button
Clicked!".*/→
```

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.   [20 Marks]

```javascript
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 = "./text.txt" + 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(8081);
```

------------------------ Slip 7 -------------------

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

```
<html ng-app="dateTimeApp">
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/
angular.min.js"></script>
</head>
<body ng-controller="dateTimeCtrl">
    <h2>Current Date and Time:</h2>
    <p>{{ currentDateTime | date:'medium' }}</p>
    <script>
    var app = angular.module('dateTimeApp', []);
    app.controller('dateTimeCtrl', function ($scope,
$interval) {
        // Function to update the current date and time
        function updateDateTime() {
        $scope.currentDateTime = new Date();
        }
        // Update the date and time every second (1000
milliseconds)
```

```
        var intervalPromise = $interval(updateDateTime,
1000);
        // Cancel the interval when the controller is
destroyed
        $scope.$on('$destroy', function () {
        $interval.cancel(intervalPromise);
        });
        // Initial update
        updateDateTime();
    });
    </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. [20 Marks]

```
var events=require('events');
var eventEmitter=new events.EventEmitter();

var connectHandler=function connected(s){
```

```
console.log('its',s);
}
eventEmitter.on('data_received',function(name){
console.log(name,"understood event driven");
});
eventEmitter.emit('data_received',"NOde");

eventEmitter.on('connection',connectHandler);
eventEmitter.emit('connection',"simplesolution");
 console.log("program ended");
```

------------------------- Slip 8 -------------------

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

```
//Assignment 22
//Create a Simple Web Server using node js

var http = require('http'); // 1 - Import Node.js core module
var server = http.createServer(function (req, res) {   // - creating server
//handle incomming requests here..
});
server.listen(8081); //3 - listen for any incoming requests
console.log('Node.js web server at port 8081 is running..')
```

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

```
<html lang="en" ng-app="studentApp">
<head>
    <meta charset="UTF-8">
```

```html
<title>Student Details</title>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/
angular.min.js"></script>
<style>
 body {
        margin: 2%;
        font-size: 100%;
 }

    th,
    td {
        padding: 10px;
    }
  </style>
</head>
<body>

<div ng-controller="StudentController as ctrl">
    <table>
    <table border=1>
        <tr>
        <th>ID</th>
        <th>Name</th>
```

```html
        <th>class</th>
        </tr>
    </thead>
    <tbody>
        <tr ng-repeat="student in ctrl.students">
        <td>{{ student.id }}</td>
        <td>{{ student.name }}</td>
        <td>{{ student.class }}</td>
        </tr>
    </tbody>
    </table>
</div>

<script>
    angular.module('studentApp', [])
    .controller('StudentController', function() {
    var ctrl = this;
    ctrl.students = [
        { id: 1, name: 'payal', class: 'FY' },
        { id: 2, name: 'ROHIT', class: 'SY' },
        { id: 3, name: 'RONIT', class: 'TY' },
        { id: 4, name: 'RAHUL', class: 'MCS' },
        { id: 5, name: 'RONIT', class: 'FY' },
        { id: 6, name: 'KARAN', class: 'SY' },
```

```javascript
            { id: 7, name: 'VIRAT', class: 'SY' },
            { id: 8, name: 'RAHUL', class: 'FY' },
            { id: 9, name: 'SUMIT', class: 'MCS' },
            { id: 10, name: 'SUMEDH', class: 'TY' }
        ];
    });
</script>

</body>
</html>
```

------------------------ Slip 9 -------------------

Q.1) Create a Node.js file that writes an HTML form, with a concatenate two string. [10 Marks]

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 [20 Marks]

```
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 = "./text.txt" + 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(8081);
```

--------------------------- Slip 10 -------------------

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

Create database

```
var mysql=require('mysql');
var con=mysql.createConnection({
host:"localhost",
user:"root",
password:"srv001"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 con.query("create database college",function(err,result){
 if(err) throw err;
 console.log("Database Created!");
 });
 });
```

Create Table

```
var mysql=require('mysql');
var con=mysql.createConnection({
```

```
host:"localhost",
user:"root",
password:"srv001",
database:"college"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 var sql="create table node(id int primary key,name
varchar(20))";
 con.query(sql,function(err,result){
 if(err) throw err;
 console.log("Table Created!");
 });
 });
```

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.   [20 Marks]

modules.js

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

slip10.js

```
var http = require('http');
var dt = require('./modules');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("The date and time is currently: " +
dt.mydatetime());
  res.end();
}).listen(8082);
```

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

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

Create database

```
var mysql=require('mysql');
var con=mysql.createConnection({
host:"localhost",
user:"root",
password:"srv001"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 con.query("create database movie",function(err,result){
 if(err) throw err;
 console.log("Database Created!");
 });
 });
```

Create Table

```
var mysql=require('mysql');
var con=mysql.createConnection({
```

```
host:"localhost",
user:"root",
password:"srv001",
database:"movie"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 var sql="create table node(id int primary key,name
varchar(20))";
 con.query(sql,function(err,result){
 if(err) throw err;
 console.log("Table Created!");
 });
 });
```

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]

```javascript
var express = require('express');
var app = express();
var PORT = 8082;

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);
});
```

Create hello.txt

------------------------ 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",
  password: "srv001",
  database: "customer"
});
con.connect(function(err) {
  if(err) throw err;
  //Select all customers and return the result object:
  con.query("SELECT * FROM customer1", function (err, result, fields) {
      if(err) throw err;
      console.log(result);
  });
});
```

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]

```html
<!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()">
```

```html
<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="text" id="mobile" name="mobile"
required><br>
<span id="mobileError" class="error"></span><br>

<label for="feedback">Feedback:</label><br>
<select id="feedback" name="feedback" required>
    <option value="">Select feedback</option>
    <option value="notgood">Not Good</option>
    <option value="good">Good</option>
    <option value="verygood">Very Good</option>
    <option value="excellent">Excellent</option>
</select><br>
<span id="feedbackError" class="error"></span><br>
```

```html
    <input type="submit" value="Submit">
</form>

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

      // Regular expressions for validation
      const nameRegex = /^[a-zA-Z\s]+$/;
      const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
      const mobileRegex = /^\d{10}$/;

      let isValid = true;

      // Name validation
      if (!nameRegex.test(name)) {
```

```javascript
        document.getElementById('nameError').innerHTML
= 'Please enter a valid name';
        isValid = false;
        } else {
        document.getElementById('nameError').innerHTML
= '';
        }

        // Email validation
        if (!emailRegex.test(email)) {
        document.getElementById('emailError').innerHTML
= 'Please enter a valid email address';
        isValid = false;
        } else {
        document.getElementById('emailError').innerHTML
= '';
        }

        // Mobile validation
        if (!mobileRegex.test(mobile)) {

        document.getElementById('mobileError').innerHTM
L = 'Please enter a valid 10-digit mobile number';
        isValid = false;
```

```
        } else {

        document.getElementById('mobileError').innerHTM
L = '';
        }

        // Feedback validation
        if (feedback === '') {

        document.getElementById('feedbackError').innerHT
ML = 'Please select feedback';
        isValid = false;
        } else {

        document.getElementById('feedbackError').innerHT
ML = '';
        }

        return isValid;
    }
</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');
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(8081);
```

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]

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

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

```
var http = require('http'); // 1 - Import Node.js core module
var server = http.createServer(function (req, res) {   // - creating server
//handle incomming requests here..
});
server.listen(8081); //3 - listen for any incoming requests
console.log('Node.js web server at port 8081 is running..')
```

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]

----------------------- 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]

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "srv001",
  database: "student "
});
con.connect(function(err) {
  if(err) throw err;
  //Select all customers and return the result object:
  con.query("SELECT * FROM students", function (err, result, fields) {
      if(err) throw err;
      console.log(result);
  });
});
```

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]

---------------------- 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]

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

```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"></script>
    <style>
     body {
         margin: 2%;
         font-size: 100%;
     }

     th,
```

```
        td {
                padding: 10px;
        }
    </style>
</head>
<body>

<div ng-controller="StudentController as ctrl">
    <table>
    <table border=1>
        <tr>
        <th>ID</th>
        <th>Name</th>
        <th>class</th>
        </tr>
    </thead>
    <tbody>
        <tr ng-repeat="student in ctrl.students">
        <td>{{ student.id }}</td>
        <td>{{ student.name }}</td>
        <td>{{ student.class }}</td>
        </tr>
    </tbody>
    </table>
```

```html
</div>
<script>
    angular.module('studentApp', [])
    .controller('StudentController', function() {
    var ctrl = this;
    ctrl.students = [
        { id: 1, name: 'payal', class: 'FY' },
        { id: 2, name: 'ROHIT', class: 'SY' },
        { id: 3, name: 'RONIT', class: 'TY' },
        { id: 4, name: 'RAHUL', class: 'MCS' },
        { id: 5, name: 'RONIT', class: 'FY' },
        { id: 6, name: 'KARAN', class: 'SY' },
        { id: 7, name: 'VIRAT', class: 'SY' },
        { id: 8, name: 'RAHUL', class: 'FY' },
        { id: 9, name: 'SUMIT', class: 'MCS' },
        { id: 10, name: 'SUMEDH', class: 'TY' }
    ];
    });
</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",
  password: "srv001",
  database: "student "
});
con.connect(function(err) {
  if(err) throw err;
  //Select all customers and return the result object:
  con.query("SELECT * FROM students", function (err, result, fields) {
      if(err) throw err;
      console.log(result);
  });
});
```

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]

```html
<html>
<html ng-app="validationApp">
<head>
    <title>AngularJS Validation</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/
angular.min.js"></script>
</head>
<body>

<div ng-controller="validationController">

    <label for="username">Username:</label>
    <input type="text" id="username" ng-model="user.username" ng-change="validateUsername()">
```

```html
    <br>

    <label for="password">Password:</label>
    <input type="password" id="password" ng-
model="user.password" ng-
change="validatePassword()">
    <br>

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

</div>

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

    app.controller('validationController', function
($scope) {
    $scope.user = {
        username: '',
        password: ''
    };

    $scope.validateUsername = function () {
        if (!$scope.user.username) {
```

```
            alert('Enter username');
            } else if ($scope.user.username.length < 3) {
            alert('Username is too short');
            } else {
            alert('Valid username');
            }
      };

      $scope.validatePassword = function () {
            if ($scope.user.password.length < 8) {
            alert('Password should be minimum 8
characters');
            }
      };

      $scope.submitForm = function () {
            // Additional logic to submit the form if needed
            // You can perform any further actions here.
      };
      });
</script>

</body>
</html>
```

---------------------- Slip 21 ------------------

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

Create database

```
var mysql=require('mysql');
var con=mysql.createConnection({
host:"localhost",
user:"root",
password:"srv001"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 con.query("create database movie",function(err,result){
 if(err) throw err;
 console.log("Database Created!");
 });
 });
```

Create Table

```
var mysql=require('mysql');
var con=mysql.createConnection({
```

```
host:"localhost",
user:"root",
password:"srv001",
database:"movie"
});
 con.connect(function(err){
 if(err) throw err;
 console.log("Connected!");
 var sql="create table node(id int primary key,name
varchar(20))";
 con.query(sql,function(err,result){
 if(err) throw err;
 console.log("Table Created!");
 });
 });
```

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]

```
var express = require('express');
var app = express();
var PORT = 8082;

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);
});
```

Create hello.txt file

------------------------ Slip 23 ------------------

Q.1) Write node js script to interact with the file system, and serve a web page from a File
[10 Marks]

slip23.html
```html
<html>
<body>
<h1>Welcome</h1>
<p>Node.js is an open source server environment</p>
</body>
</html>
```

node.js file
```javascript
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('slip14.html', function(err, data) {
      res.writeHead(200, {'Content-Type': 'text/html'});
      res.write(data);
      return res.end();
  });
}).listen(8080);
```

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. [20 Marks]

modules.js
```
exports.mydatetime = function () {
  return Date();
};
```

slip10.js

```
var http = require('http');
var dt = require('./modules');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("The date and time is currently: " +
dt.mydatetime());
  res.end();
}).listen(8082);
```

------------------------- Slip 25 ------------------

Q.1) Create an angular JS Application that shows the location of the current web page.
[10 Marks]

```html
< html>
<html ng-app="locationApp">
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js"></script>
</head>
<body ng-controller="locationCtrl">
    <h2>Current Location:</h2>
    <p>{{ currentLocation }}</p>
    <script>
    var app = angular.module('locationApp', []);
    app.controller('locationCtrl', function ($scope, $location) {
        // Get and display the current location
        $scope.currentLocation = $location.absUrl();
    });
    </script>
</body>
</html>
```

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 callback function when one of those events is detected. [20 Marks]

```
var events=require('events');
var eventEmitter=new events.EventEmitter();

var connectHandler=function connected(s){
console.log('its',s);
}
eventEmitter.on('data_received',function(name){
console.log(name,"understood event driven");
});
eventEmitter.emit('data_received',"NOde");

eventEmitter.on('connection',connectHandler);
eventEmitter.emit('connection',"simplesolution");
 console.log("program ended");
```

Cancel Slips

----------------------- Slip 4 ------------------
Q.1) Using node js create an eLearning System.
[10 Marks]
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).
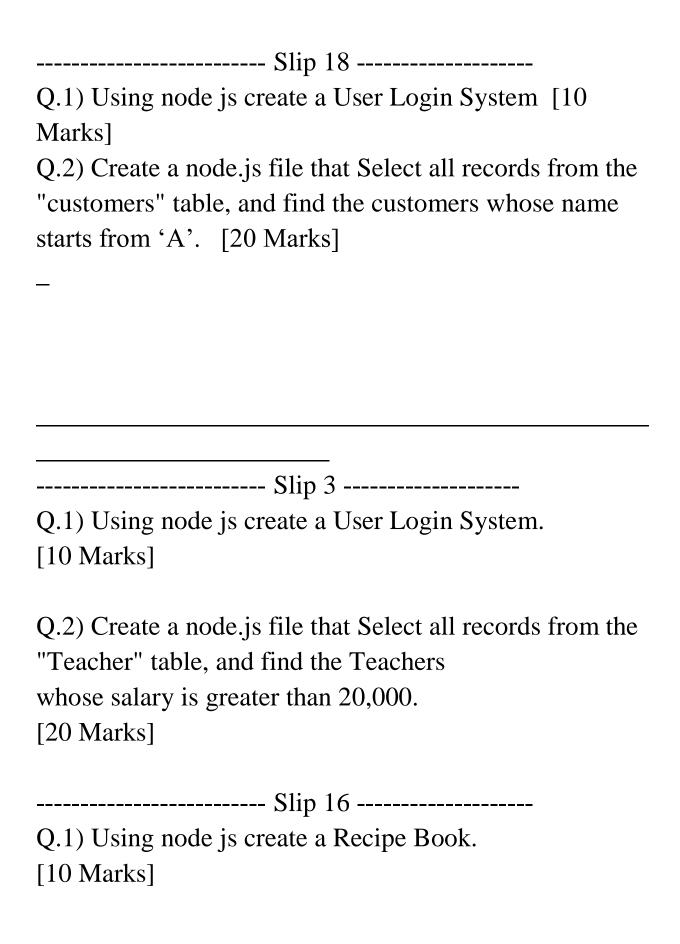[20 Marks]

----------------------- Slip 24 ------------------
Q.1) Using node js create an eLearning System
[10 Marks]
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]

----------------------- Slip 22 ------------------

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

[10 Marks]

Q.2) Create a js file named main.js for event-driven application. There should be a main loop thatlistens for events, and then triggers a callback function when one of those events is detected. [20 Marks]

------------------------ Slip 18 -------------------

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

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]

_

------------------------ Slip 3 -------------------

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

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.
[20 Marks]

------------------------- 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]

-------------------------- Slip 5 -------------------

Q.1) Create a Node.js file that writes an HTML form, with an upload field.
[10 Marks]

slip5_uploadform.js
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">
');

```javascript
  res.write('<input type="submit">');
  res.write('</form>');
  return res.end();
}).listen(8080);
```

Slip5_uploadedfile.js

```javascript
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);
```

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]

```html
<html>
<html ng-app="validationApp">
<head>
    <title>AngularJS Validation</title>
```

```html
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/
angular.min.js"></script>
</head>
<body>

<div ng-controller="validationController">

    <label for="username">Username:</label>
    <input type="text" id="username" ng-
model="user.username" ng-
change="validateUsername()">
    <br>

    <label for="password">Password:</label>
    <input type="password" id="password" ng-
model="user.password" ng-
change="validatePassword()">
    <br>

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

</div>
```

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

    app.controller('validationController', function
($scope) {
    $scope.user = {
        username: '',
        password: ''
    };

    $scope.validateUsername = function () {
        if (!$scope.user.username) {
        alert('Enter username');
        } else if ($scope.user.username.length < 3) {
        alert('Username is too short');
        } else {
        alert('Valid username');
        }
    };

    $scope.validatePassword = function () {
        if ($scope.user.password.length < 8) {
        alert('Password should be minimum 8
characters');
```

```
        }
    };

    $scope.submitForm = function () {
        // Additional logic to submit the form if needed
        // You can perform any further actions here.
    };
    });
</script>

</body>
</html>
```

-------------------------- 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');
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(8081);
```

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]