NODEJS Solved Slips

Slip 1a. Create a Node.js file that will convert the output "Hello World!" into upper-case letters.

First install upper-case module

C:\Users\Your Name>npm upper-case

```
var http = require('http');
var uc = require('upper-case');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(uc.upperCase("Hello World!"));
  res.end();
}).listen(8080);
console.log('Server running at http://127.0.0.1:8080/');
```

Slip 1b. Create a nodejs file that demonstrates creating database student DB and student table student (roll_no, name, percentage) in mysql.

Step 1 : download and install https://dev.mysql.com/downloads/installer/

Step 2 : go to your folder and on terminal : npm install mysql2

Step 3 : create following files

Dbconn.js

```
var mysql = require('mysql2');
var con = mysql.createConnection({
host: "localhost",
user: "root",
password: "root"
});
con.connect(function(err) {
if (err) throw err;
console.log("Connected!");
});
```

Dbcreate.js

```
var mysql = require('mysql2');
```

```
var con = mysql.createConnection({
host: "localhost",
user: "root",
password: "root"
});
con.connect(function(err) {
if (err) throw err;
console.log("Connected!");
con.query("CREATE DATABASE mydb",
function (err, result) {
if (err) throw err;
console.log("Database created");
});
});
```

Createtable.js

```
var mysql = require('mysql2');
var con = mysql.createConnection({
host: "localhost",
user: "root",
password: "root",
database: "mydb"
});
con.connect(function(err) {
if (err) throw err;
console.log("Connected!");
var sql = "CREATE TABLE SYstudent (rollno INT AUTO_INCREMENT PRIMARY KEY,
studentname VARCHAR(255), percentage INT)";
con.query(sql, function (err, result) {
if (err) throw err;
console.log("Table created");
});
});
```

Insertrecord.js

```
var mysql = require('mysql2');
var con = mysql.createConnection({
host: "localhost",
user: "root",
password: "root",
database: "mydb"
```

```
});
con.connect(function(err) {
if (err) throw err;
console.log("Connected!");
var sql = "INSERT INTO SYstudent (studentname, percentage) VALUES ('akshay kumar', 80)";
con.query(sql, function (err, result) {
if (err) throw err;
console.log("1 record inserted");
});
});
```

Slip 2a) Create a Node.js Application that uses a user defined module to return the factorial of a given number.

fact.js

```
var fact={
  factorial: function(n)
{
    var f=1,i;
 for(i=1;i<=n;i++)

   {     f=f*i;
   }
  console.log('factorial of '+n+' is:'+f);}};
 module.exports=fact
```

app.js

```
var mymod=require('f:/swapna/nodejs/fact.js');
mymod.factorial(5);
```

Slip 2b. Create a nodejs application that contain the employee registration details and write a javascript function to validate DOB, joining date and salary

Same steps to be followed as given in step 1b

Slip 3a. Slip No.3 and Slip no. 30.Create a Node.js Application that uses user defined module circle.js which exports functions area() and circumference() and display details on console.

Circle.js

```
var circle={
  area: function(r)
{
    var pi=3.14,a;
     a=pi*r*r;

  console.log('area of circle is:'+a);
},
circumference: function(r)
{
    var pi=3.14,c;
  c=2*pi*r;
  console.log('circumference of circle is:'+c);
}
};
 module.exports=circle
```

Mycircle.js

```
var mymod=require('C:/swapna/nodejs/circle.js');
mymod.area(5);
mymod.circumference(5);
```

Slip 3b. Create a nodejs application to validate student registration form.

```
>npm install express-validator;
>npm install express;
>npm install ejs;
>npm install body-parser;
```

validatestudent.js

```
const { check, validationResult }
   = require('express-validator');
```

```
const bodyparser = require('body-parser')
const express = require("express")
const path = require('path')
const app = express()

var PORT = process.env.port || 3000

// View Engine Setup
app.set("views", path.join(__dirname))
app.set("view engine", "ejs")

// Body-parser middleware
app.use(bodyparser.urlencoded({ extended: false }))
app.use(bodyparser.json())

app.get("/", function (req, res) {
    res.render("SampleForm");
})

// check() is a middleware used to validate
// the incoming data as per the fields
app.post('/saveData', [
    check('email', 'Email length should be 10 to 30 characters').isEmail().isLength({ min: 10, max:
30 }),
    check('name', 'Name length should be 10 to 20 characters').isLength({ min: 10, max: 20 }),
    check('mobile', 'Mobile number should contains 10 digits').isLength({ min: 10, max: 10 }),
    check('password', 'Password length should be 8 to 10 characters').isLength({ min: 8, max: 10
})
], (req, res) => {

    // validationResult function checks whether
    // any occurs or not and return an object
    const errors = validationResult(req);

    // If some error occurs, then this
    // block of code will run
    if (!errors.isEmpty()) {
        res.json(errors)
    }

    // If no error occurs, then this
    // block of code will run
    else {
```

```
        res.send("Successfully validated")
    }
});

app.listen(PORT, function (error) {
    if (error) throw error
    console.log("http://127.0.0.1:3000 Server created Successfully on PORT ", PORT)
})
```

SampleForm.ejs

```
<!DOCTYPE html>
<html>
    <head>
        <title>Validation using Express-Validator</title>
    </head>
<body>
<h1>Student Registration Form</h1>

<form action="saveData" method="POST">
  <pre>
      Enter student Email    : <input type="text" name="email"> <br>
      Enter student Name     : <input type="text" name="name"> <br>
      Enter student Number   : <input type="number" name="mobile"> <br>
      Enter student Password : <input type="password" name="password"> <br>
      <input type="submit" value="Submit Form">
  </pre>
</form>

</body>
</html>
```

Slip no 4.

a) **Create a nodejs application that accepts first name and last name and concatenates them. Also display the date of birth entered.**

Step 1 : npm install findage

```
var http = require('http');
var firstname="swapna ";
var lastname="kolhatkar";
console.log(firstname.concat(lastname));
```

```
const getAge = require('findage') ;
getAge.fullAge
console.log(getAge.fullAge("04/04/2003"));
```

Slip 4b. Create teacher profile system using nodejs

Same steps to be followed as given in step 1b

Slip 5 a) Create a Node.js application that performs the following operations on buffer data

Concat
Compare
Copy

```
var buf1 = Buffer.from("Hello");
var buf2 = Buffer.from("World");
var buf3 = Buffer.from("SYBBA-CA");
var list = [buf1, buf2, buf3];
var newbuff = Buffer.concat(list);
console.log("The concatenated buffer:");
console.log(newbuff);

var buf4 = Buffer.from("s");
var buf5 = Buffer.from("s");
var buf6 = Buffer.from("t");
var combuf = Buffer.compare(buf4, buf5);
console.log("buf4 compared with buf5 is "+combuf);
var combuf = Buffer.compare(buf4, buf6);
console.log("buf4 compared with buf6 is "+combuf);

var buf7 = Buffer.from('This is new subject : ');
var buf8 = Buffer.from('SYBCA');
buf8.copy(buf7, 2);
console.log(buf7.toString());
```

Slip 5 b)

Create a node.js file that selects all records from customer table and deletes the specified record.

Same steps to be followed as given in step 1b

Slip no 6 a) : create a node.js file that opens the requested file and returns the content to the client. If anything goes wrong then throw the 404 error.

Step 1 : create a input.txt file in your directory

```
var fs = require("fs");
// Asynchronous - Opening File
console.log("Going to open file!");
fs.open('input.txt', 'r+', function(err, fd) {
if (err) {
return console.error("404" + err);
}
console.log("File opened successfully!");
});
```

Slip 6 b) : create a node.js file that inserts multiple records in the student table and displays the result object on the console.

Same steps to be followed as given in step 1b

Slip 7a) Using node js create a web page to read two file names from user and append contents of first file into second file.

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

console.log('File Reading from file1.txt .....');

fs.readFile('file1.txt', 'utf8', readingFile);

function readingFile(error, data) {
   if (error) {
      console.log(error);
   } else {
      console.log(data);
      fs.appendFile('file2.txt', data, 'utf8', writeFile);
   }
```

```
}

function writeFile(error) {
    if (error) {
        console.log(error)
    } else {
        console.log('Content of file1.txt has been pasted to file2.txt file');
    }
}
```

Slip 7b) create a nodejs file that selects all records from customer table and displays the result object on the console.

Same steps to be followed as given in step 1b

Slip 8 a) Using node.js create a web page to read two file names from user and combine in third file with all letters in upper case.

Serverpage.js

```
var express = require("express");
var app = express();
 app.get("/", function (request, response){
     response.sendFile(__dirname+"/serverhtml.html");
});

app.get("/addfiles", function (request, response){
    var file1 = request.query.file1;
    var file2 = request.query.file2;

    if (file1 != "") {
        response.send("Your files are read. Check the temp.txt for output");
    } else {
        response.send("Please provide us all file name");
    }
    const fs = require('fs');
    const uc = require('upper-case');
```

```
const file11 = uc.upperCase(fs.readFileSync(file1, 'utf-8'));

fs.writeFileSync('temp.txt', file11);
const file22 = uc.upperCase(fs.readFileSync(file2, 'utf-8'));
fs.appendFileSync('temp.txt',file22); //('temp.txt', file22);
});

app.listen(8081);
console.log("Something awesome to happen at http://localhost:8081");
```

Serverhtml.html

```
<html>
   <head>
      <title>Simple hello world</title>
   </head>
<body>
   <h1>Hello Node JS</h1>
   <form method='GET' action="/addfiles">
      <span>Enter file name 1 :</span>
      <input type="file" id="file1" name="file1" />
      <span>Enter file name 2 :</span>
      <input type="file" id="file2" name="file2" />
      <input type="submit" value="combine files" />
   </form>
</body>
</html>
```

Slip no 8 b) create node js application that contains student registration details and validate student first name and last name should not contain special characters or digits and age should be between 6 and 25.

Same as slip no 3 b) solution.

Slip No.9. Create a Node.js file that writes HTML form with an upload field.

upload.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"><br>');
  res.write('<input type="submit">');
  res.write('</form>');
  return res.end();
}).listen(8081);
```

Initiate upload.js file :

C:\Users\Your Name>node upload.js

Slip no 9 a) create a nodejs file that writes an html form with an upload field.

Slip no 9 b) using nodejs create a recipe book

Home.ejs

```
<%- include('navigation') %>

<h1><centre>Home</centre></h1>
<p> Welcome to Recipe Book. This page gives information of Indian recipes that can be cooked
by young and old alike. </p>
<ul>
   <li>Breakfast Recipe</li>
      <ol>upma</ol>
      <ol><a href = "http://google.com">pohe</a></ol>
      <ol>maggi</ol>
      <ol>bread</ol>
   <li>Lunch Recipe</li>
      <ol>thali</ol>
      <ol><a href = "http://google.com">poli bhaji</a></ol>
      <ol>dal rice</ol>
      <ol>sabji</ol>
   <li>Snacks Recipe</li>
      <ol>khichdi</ol>
      <ol><a href = "http://google.com">soup</a></ol>
```

```
        <ol>pasta</ol>
</ul>

<%- include('footer') %>
```

App.js

```
const express = require('express');
const app = express();
app.set('view engine', 'ejs');
app.get('/', (req, res) =>{
  res.render('home.ejs');
});
app.listen(3000);
```

navigation.js

```
<!DOCTYPE html>
<html lang="en">
<head>
      <meta charset="UTF-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content=
         "width=device-width, initial-scale=1.0">
      <title>Home</title>
</head>
<body>

<h1>Navbar</h1>
```

Footer.js

```
<h1>Footer</h1>
</body>
</html>
```

Slip No.14. Create a Node.js application to search particular word in file and display result on console.

search_word.js

```
var fs=require('fs');
fs.readFile('C:/Users/Public/node_prog/searchf.txt', function (err, data) {
  if (err) throw err;

  if(data.includes('dfgdf')){
   console.log(data.toString())
  }
else
{
  console.log('word not found');
}
});
```

Slip No.13 and Slip no.29. Create a Node.js application that uses user defined module to find area of rectangle and display details on console.

rect.js

```
var rect={
  area: function(l,b)
{
    var a;
     a=l*b;

  console.log('area of rectangle is:'+a);
}
};
 module.exports=rect
```

myrect.js

```
var mymod=require('C:/Users/Public/node_prog/rect.js');
mymod.area(5,4);
```

Initiate myrect.js file :

```
C:\Users\Your Name>node myrect.js
```