

Subject: WEB FRAMEWORK SLIP SOLUTION..

1.Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.

Slip1.js

```
function validateForm() {
    var fname = document.getElementById("fname").value;
    var lname = document.getElementById("lname").value;
    var age = document.getElementById("age").value;
    var reg = /^[a-zA-Z]+$/; //REGEX for only use of
    alphabets.
    //Without +$ it will match only the start position but
    with it will match the whole string/word

    //if Fields are empty
    if(fname.length == 0 || lname.length == 0 || age.length
    == 0){
        alert("All Fields are Mandatory");
        return false;
    }
    //test will check boolean values. if fname/lname does
    not contain alphabets
    else if(!reg.test(fname) || !reg.test(lname)){
        alert("Only Alphabets Allowed");
        return false;
    }
    else if(age < 18 || age > 50){
        alert("Enter Age between 18 & 50 Only")
        return false;
    }
    else{
        alert("Validation Successfull")
    }
}
```

```
        return true;
    }
}
```

Slip1.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Slip 1 Name and Age Validation</title>
        <script src="student.js"></script>
    </head>
    <body>
        <form>
            First Name: <input type="text"
id="fname"></input></br></br>
            Last Name: <input type="text"
id="lname"></input></br></br>
            Age: <input type="text"
id="age"></input></br></br>
            <button type="submit"
onclick="validateForm()">Submit</button>
        </form>
    </body>
</html>
```

Output

First Name:

Last Name:

Age:

This page says
Validation Successfull

Activate Windows
Go to Settings to activate Windows.



First Name:

Last Name:

Age:

This page says
Enter Age between 18 & 50 Only

Activate Windows
Go to Settings to activate Windows.



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

Slip2.js

```
function validateform(){
    var name = document.getElementById("name").value;
    var dob = document.getElementById("dob").value;
    var jdate = document.getElementById("jdate").value;
    var sal = document.getElementById("sal").value;
    var regName = /^[a-zA-Z0-9]+\s[a-zA-Z0-9]+$/; //REGEX
for Full Name
    var regdate = /[0-9]{1,2}\/[0-9]{1,2}\/[0-9]{4}$/; //
DD/MM/YYYY. '/' before '\' is used to treat '\' is as
symbol and not as escape character
    var regsal = /[0-9]/;

    //if Fields are empty
    if(name.length == 0 || dob.length == 0 || jdate.length
== "" || sal.length == 0){
        alert("All Fields are Mandatory");
        return false;
    }
    else if(!regName.test(name)){
        alert("Enter Name Format Correctly \"First Name
Last Name\"");
        return false;
    }
    else if(!regdate.test(dob)){
        alert("Registration Date Format DD/MM/YYYY");
        return false;
    }
    else if(!regdate.test(jdate)){
        alert("Date of Joining Format DD/MM/YYYY");
        return false;
    }
    else if(!regsal.test(sal)){
```

```

        alert("Enter Numerical Values only in Salary");
        return false;
    }
    else{
        alert("Validation Successfull")
        return true;
    }
}

```

Slip2.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>Slip 2 Employee Registration Details</title>
        <script src="Slip2.js"></script>
    </head>
    <body>
        <form>
            Full Name: <input type="text"
id="name"></input></br></br>
            DOB <input type="text"
id="dob"></input></br></br>
            Joining Date <input type="text"
id="jdate"></input></br></br>
            Salary <input type="text"
id="sal"></input></br></br>
            <button type="submit"
onclick="validateform()">Submit</button>
        </form>
    </body>
</html>

```

Output

Full Name:

DOB

Joining Date

Salary

This page says
Validation Successfull

Type here to search

Links

22°C Mostly clear

8:08 PM
12/28/2022

3.Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.

Slip3.js

```
function validateform() {  
    var email = document.getElementById("email").value;  
    var pass = document.getElementById("pass").value;  
    var regEmail = /^[a-zA-Z0-9\._]+@([a-z]+)(.[a-z]+)?$/;  
  
    //if Fields are empty  
    if(email.length == 0){  
        alert("All Fields are Mandatory");  
        return false;  
    }  
    else if(pass.length == 0){  
        alert("All Fields are Mandatory");  
        return false;  
    }  
    else if(!regEmail.test(email)){  
        alert("Enter Name Format Correctly \"First Name  
Last Name\"");  
    }  
}
```

```
        return false;
    }
    else{
        alert("Validation Successfull");
        return true;
    }
}
```

Slip3.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Slip 3 Email Validation</title>
        <script src="Slip3.js"></script>
    </head>
    <body>
        <form>
            Email: <input type="text"
id="email"></input></br></br>
            Password: <input type="password"
id="pass"></input></br></br>
            <button type="submit"
onclick="validateform()">Submit</button>
        </form>
    </body>
</html>
```

Output

Email:

Password:

This page says
Validation Successfull



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

Slip4.js

```
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);
```

HERE WE CAN DOWNLOAD THE MODULE UPPERCASE

Output

HELLO WORLD!!

Activate Windows
Go to Settings to activate Windows.



5.Using nodejs create a web page to read two file names from user and append contents of first file into second file.

Slip5.js

```
var fs = require('fs');
var path = require('path')
var data = fs.readFileSync("Sample1.txt", "utf8");
fs.appendFile("Sample2.txt", data, (err) =>{
    if(err) {
        console.log(err);
    }
    else{
        console.log("File Content after appending:
",fs.readFileSync("Sample2.txt", "utf8"));
    }
});
```

Sample1.txt

hello world

Sample2.txt

nice world

Ouput



6. Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error.

Slip6.js

```
var http=require('http');
var fs=require('fs');
var url=require('url');
http.createServer(function (request,response)
{
    var q=url.parse(request.url,true);
    var filename="."+q.pathname;
    fs.readFile(filename,function(error,data)
    {
        if(error)
        {
            response.writeHead(404,{ 'content-
type':'text/html' });
            return response.end("404 not found");
        }
        response.writeHead(200,{ 'content-type':'text/html' });
        response.write(data);
        response.end();
    });
}).listen(8000)
```

Sample.txt

WHAT IS GOOGLE?

Google is a popular internet search engine. It scans the Web to find Web pages that are relevant to the words you have typed in the search box.

Output

404 not found



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

Slip7.js

```
var http = require('http');
var formidable = require('formidable'); //module is used
for parsing form data, for handling incoming form data and
file uploads
```

```
http.createServer(function(req,res){
```

```

    var form = new formidable.IncomingForm(); //Creates a
new incoming form.
    form.parse(req,function(err,fields,files){
        if(req.url=='/fileupload'){ //if user request of
uploading file is successful then "File Uploaded
Successfully"
            console.log(files);
            res.write('File Uploaded');
            res.end();
        }
        else{
            res.writeHead(200,{'Content-
Type':'text/html'});
            //enctype multipart is used for dealing with
files, usually 'text' is the type for pass,name,email
            res.write('<form action = "fileupload" method =
"get" enctype = "multipart/form_data">');
            res.write('<input type = "file"
name="fileuploaded"><br><br>');
            res.write('<input type = "submit">');
            res.write('</form>');
            return res.end();
        }
    });
}).listen(8080);

```

Output

Choose File Narcissus-Fl...pressed.jpg

Submit



8.Create a Simple Web Server using node js.

Slip12.js

```
var http = require('http');
http.createServer(function(req,res){
    res.writeHead(200, {'Content-Type':'text/html'});
    res.end("hello world");
}).listen(8080);
```

Output

hello world



9. 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.

Slip15.js

```
var http = require('http');
var dt = require('./module');

http.createServer(function(req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('The date and time currentlty: ' +
dt.myDateTime());
    res.end();
}).listen(8080);
```

Module.js

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

Output

The date and time currently: Tue Dec 27 2022 15:13:38 GMT+0530 (India Standard Time)

Activate Windows
Go to Settings to activate Windows.



10. Write node js script to interact with the filesystem, and serve a web page from a file .

Slip14.js

```
var fs=require('fs')

var http=require('http')
object=http.createServer(function(request,response)
{
    fs.readFile("a.txt",function(error,data){
        if(error)
            console.log("error message is",error)
        else
            fs.appendFile("b.txt",data,function(error)
            {
                fs.readFile("b.txt",function(error,data){

                    var str=data.toString()
                    console.log(data.toString())
                    response.writeHead(200,{ 'content-
type': 'text/html'})
                    response.write(str)
                    response.end()

                });
            });
        });
    });
});
object.listen(8000);
```

output

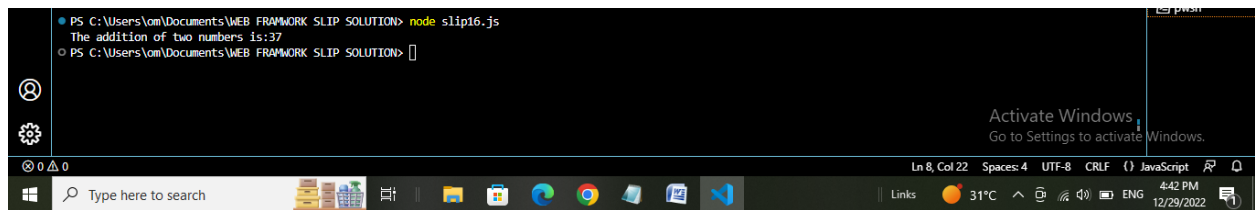


11. Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those events is detected.

Slip16.js

```
var events=require('events');
var em=new events.EventEmitter();
em.on('add',function(a,b)
{
    addition=a+b;
    console.log('The addition of two numbers
is:'+addition);
});
em.emit('add',12,25);
```

output



A screenshot of a Windows command prompt window. The title bar reads "cmd - PowerShell". The command prompt shows the following text:

```
PS C:\Users\om\Documents\WEB FRAMEWORK SLIP SOLUTION> node slip16.js
The addition of two numbers is:37
PS C:\Users\om\Documents\WEB FRAMEWORK SLIP SOLUTION>
```

The Windows taskbar is visible at the bottom, showing the search bar, task view button, and several application icons. The system tray on the right shows the date and time as 4:42 PM on 12/29/2022, along with various system icons.

12. 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.

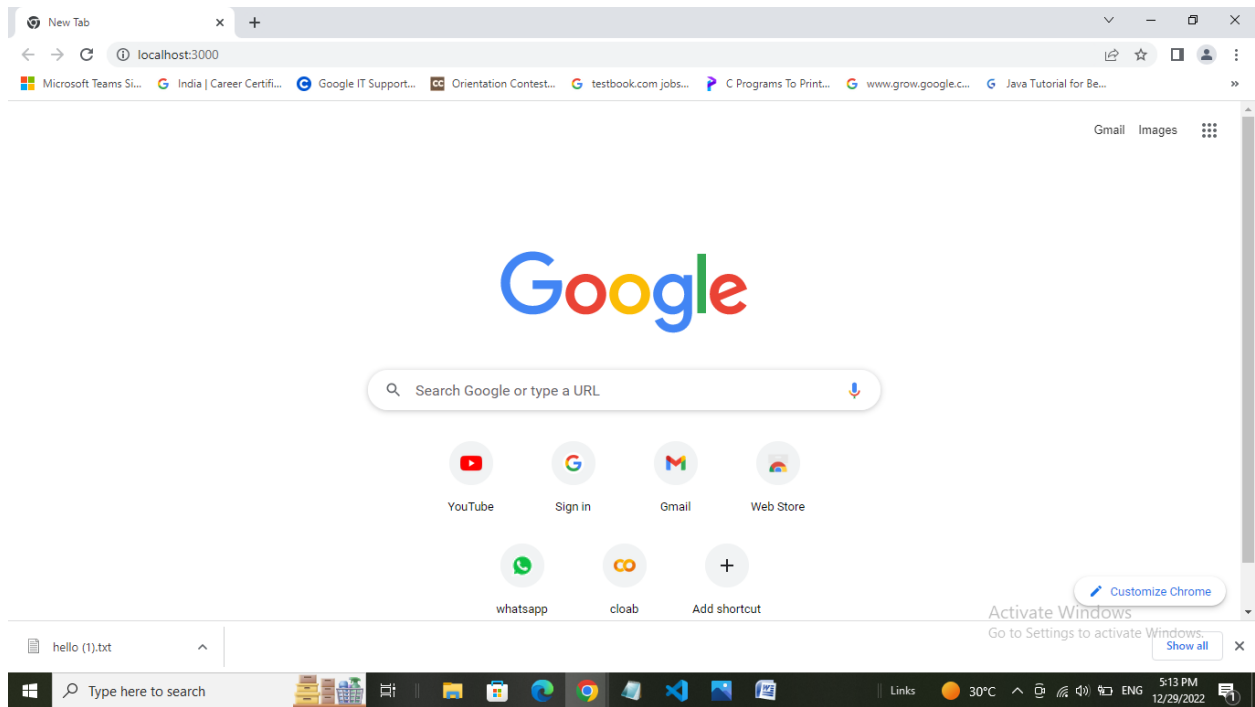
Slip17.js

```
var express = require('express');
var app = express();
var PORT = 3000;

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);
});
```

Output



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

Slip8.js

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

var con = mysql.createConnection({
  host: "localhost",
  port: 3306,
  user: "root",
  password: "sakshi@123"
});

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");
        });

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

```

14.Create a node.js file that Select all records from the "customers" table, and display the result object on console.

Slip9.js

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

```

var con = mysql.createConnection({
    host: "localhost",
    port: 3306,
    user: "root",
    password: "sakshi@123",
    database: "mydb"
});

```

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

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

Slip10.js

```
var mysql = require('mysql2');  
  
var con = mysql.createConnection({  
  host: "localhost",  
  port: 3306,  
  user: "root",  
  password: "sakshi@123",
```

```

    database: "mydb"
  });

  con.connect(function (err) {
    if (err) throw err;
    console.log("Connected!");
    var sql = "INSERT INTO student (name, address) VALUES ('Sakshi', 'Pune'), ('Arati', 'Akurdi'),('Ram','Nigdi')";
    con.query(sql, function (err, result) {
      if (err) throw err;
      console.log(result);
    });
    con.query("SELECT * FROM student", function (err, result, fields) {
      if (err) throw err;
      console.log(result);
    });
  });
});

```

16. Create a node.js file that Select all records from the "customers" table, and delete the specified record.

Slip11.js

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

```
var con = mysql.createConnection({  
  host: "localhost",  
  port: 3306,  
  user: "root",  
  password: "sakshi@123",  
  database: "mydb"  
});
```

```
con.connect(function (err) {  
  if (err) throw err;  
  console.log("Connected!");  
  con.query("SELECT * FROM customers", function (err, result, fields) {  
    if (err) throw err;  
    console.log(result);  
  });  
  con.query("delete FROM customers where name = 'sakshi'", function (err,  
result, fields) {  
    if (err) throw err;  
    console.log(result);  
  });  
});
```

17. Create your Django app in which after running the server, you should see on the browser, the text “Hello! I am learning Django”, which you defined in the index view.

Slip18

Greeting

views.py

```
from django.shortcuts import render
from django.http import HttpResponse
def index(request):
    return HttpResponse('<b>hello</b>')
```

urls.py

```
from django.urls import path
from . import views

urlpatterns=[
    path('',views.index,name='index'),
]
```

Webapp1

settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    "greeting",
]
```

urls.py

```
from django.contrib import admin
from django.urls import path
```



```
from greeting import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
]
```

Output

