

NAME :- AKSHADA RAMCHANDRA DAUNDKAR

ROLL NO.:- SG009

WEB **FRAMEWORKS** **PRACTICAL SLIP** **SOLUTION**

Solutions:

slip1)

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.

```
<html>
  <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>STUDENT REGISTRATION</title>
    <script>
      function student()
      {
        var fname=document.getElementById("s1").value;
        var patt=/[A-Za-z]+$/;
        if(!fname.match(patt))
        {
          alert("first name should contain alphabets only");
        }
      }
    </script>
  </head>
  <body>
    <input type="text" id="s1"/>
  </body>
</html>
```

```

return false;
}
else
{
alert("first name entered");
}
var lname=document.getElementById("s2").value;
var patt2=/[A-Za-z]+$/
if(!lname.match(patt2))
{
alert("last name should contain alphabets only");
return false;
}
else
{
alert("last name entered");
}
let age=parseInt(document.getElementById("age").value);
if(age<18||age>50)
{
document.write("invalid age");
}
}
</script>
</head>
<body>
<form action="">
Student First name:<input type="text" id="s1"><br>
Student Last name:<input type="text" id="s2"><br>
Student Age:<input type="number" id="age"><br>
<button onclick="student()">Submit</button>
</form>
</body>
</html>

```

slip2)

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

```

<!DOCTYPE html>
<html>
<head>

```

```
<script>
function formValidation()
{
var first=document.registration.first;

var last=document.registration.last;

var birth=document.registration.birth;

var jdate=document.registration.jdate;

var salary =document.registration.salary;

if(allb(birth))

{

if(allnumerics(jdate))

{

if(allnumeric(salary))

{

}}}}

function allb(birth)

{

var birth_len = birth.value.length;

if (birth_len == 0)

{

alert("birth date should not be empty");

birth.focus();

return false;
```

```
}

alert("birth of date submitted");

return true;

}

function allnumerics(jdate)

{

var jdate_len = jdate.value.length;

if (jdate_len == 0)

{

alert("date of joining should not be empty");

birthday.focus();

return false;

}

alert("date of joining submitted");

return true;

}

function allnumeric(salary)

{

var sal = /[0-9]+$/;

if(salary.value.match(sal))

{

alert("salary submitted");

return true;

}
```

```
}

else

{

alert('salry is not submitted');

salary.focus();

return false;

}

}

</script>

</head>

<body>

<h1>Employee Registration Form</h1>

<form name='registration' onSubmit="return formValidation();">

<ul>

<li><label for="first">First Name:</label></li>

<li><input type="text" name="first" size="50" /></li>

<li><label for="last">Last Name:</label></li>

<li><input type="text" name="last" size="50" /></li>

<li><label for="birth">Birth of date:</label></li>

<li><input type="date" id="birth" name="birth"></li>

<li><label for="jdate">Date of joining:</label></li>
```

```
<li><input type="date" id="" name="jdate"></li>
<li><label for="salary">salary:</label></li>

<li><input type="number" id="salary" name="salary"></li>

<li><input type="submit" name="submit" value="Submit" /></li>

</ul>

</form>

</body>

</html>
```

Slip3)

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

```
<html>

<head>

<title>

Email validation

</title>

<script>

function validate()

{

var name=document.getElementById("t1").value;
```

```

var patt=/[A-Z a-z]{3,}[0-9]*[@]{1}[A-Z a-z]{3,}[.]{1}[A-Z a-
z.]{2,6}$/;

document.write(name.match(patt))

if(name.match(patt))
{
    alert("Email Address is Valid")

}
else{
    alert("Invalid Email Address")

}

}

</script>

</head>

<form>

<h1>EMAIL VALIDATION</h1>

Enter Email Address:<input type="text" id="t1"><br>

<button onClick="validate()">click</button>

</form>

</html>

```

Slip4)

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

```
var str="hello world"; var http=require('http');

var uc=require('upper-case');

http.createServer(function(req,res)

{

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

    //use uppercase module

res.write(uc.toUpperCase(str));

res.end();

}).listen(4000);
```

Slip5)

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

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

fs.readFile("b.txt",function(err,data)

{

    if(err)

        console.log("Error message is",err)

    else

        fs.appendFile("first.txt",data,function(err)
```



```

{
    if(err) throw err

    console.log(fs.readFileSync("first.txt","utf-8"))

    console.log("Content is updated")

});
})

```

Slip6)

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

```

:
var http=require('http');

var fs=require('fs');

var url=require('url');

http.createServer(function(request,response)
{
    var q=url.parse(request.url,true);//(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(3080);
```

Slip7)

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

```
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)
    {
      console.log(files)

      res.write('fileupload');

      res.end();

    });
  }
});
```

```

    }else
    {

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

        res.write('<form action="fileupload" method="get enctype="
multipart/form-data">');

        res.write('<input type="file" name="fileupload">');

        res.write('<input type="submit">');

        res.write('</forms>');

        return res.end();

    }

    console.log("file upload sucessfully");

}).listen(4080);

```

Slip12,Slip25)

Create a Simple Web Server using node js.

```

var http=require('http')

http.createServer(function(request,response){

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

    response.end("Hello World");

}).listen(4000);

```

Slip14)

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

```
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("first.txt",data,function(error)

            {

                fs.readFile("first.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(2080);
```

Slip15)

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.

Node.js:-

```
var http=require('http')
var dt=require('./module.js')
http.createServer(function(request,response)
{
    response.writeHead(200,{'content-type':'text/HTML'});
    response.write("The date and time are currently is"+dt.myDatetime())
    response.end();
}).listen(8080);
```

Module.js

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

Slip16)

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.

```
const EventEmitter = require('events');  
var events=require('events')  
var em=new EventEmitter()  
em.on('power',function(a,b)  
{  
    power=a;  
    for(var i=1;i<=b;i++)  
        power=power*a  
    console.log("Power is:",power)  
  
});  
em.emit('power',2,3)
```

Django Application

PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> python --
version

Python 3.11.1

PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip --
version

```
pip 22.3.1 from
C:\Users\Akshida\AppData\Local\Programs\Python\Python311\Lib\site-
packages\pip (python 3.11)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip install
django
Requirement already satisfied: django in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (4.1.4)
Requirement already satisfied: asgiref<4,>=3.5.2 in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (3.6.0)
Requirement already satisfied: sqlparse>=0.2.2 in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (0.4.3)
Requirement already satisfied: tzdata in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (2022.7)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> clear
* History restored
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> python --
version
Python 3.11.1
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip --
version
pip 22.3.1 from
C:\Users\Akshida\AppData\Local\Programs\Python\Python311\Lib\site-
packages\pip (python 3.11)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip install
django
Requirement already satisfied: django in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (4.1.4)
Requirement already satisfied: asgiref<4,>=3.5.2 in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (3.6.0)
```

Requirement already satisfied: sqlparse>=0.2.2 in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (0.4.3)
Requirement already satisfied: tzdata in
c:\users\akshida\appdata\local\programs\python\python311\lib\site-
packages (from django) (2022.7)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> django-
admin --version
4.1.4
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> mkdir pro

Directory: C:\Users\Akshida\Desktop\slip\webframework\New folder (2)

Mode	LastWriteTime	Length	Name
d-----	28-12-2022 15:10		pro

PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> cd pro
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)\pro> django-
admin startproject webapplication
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)\pro> dir

Directory: C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro

Mode	LastWriteTime	Length	Name
d-----	28-12-2022 15:10		webapplication


```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)\pro> cd
webapplication          cd : Cannot find path
'C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication' because it does not exist.
```

```
At line:1 char:1
```

```
+ cd webapplication
```

```
+ ~~~~~
```

```
+ CategoryInfo          : ObjectNotFound:
(C:\Users\Akshid...o\webapplication:String) [Set-Location], ItemNotFoundException
```

```
+ FullyQualifiedErrorId :
PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)\pro> cd
webapplication
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication> dir
```

```
Directory: C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication
```

Mode	LastWriteTime	Length	Name
d----	28-12-2022 15:10		webapplication
-a----	28-12-2022 15:10	692	manage.py

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication> cd webapplication
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication\webapplication> dir
```

```
Directory: C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication\webapplication
```

Mode	LastWriteTime	Length	Name
-a----	28-12-2022 15:10	421	asgi.py
-a----	28-12-2022 15:10	3368	settings.py
-a----	28-12-2022 15:10	777	urls.py
-a----	28-12-2022 15:10	421	wsgi.py
-a----	28-12-2022 15:10	0	__init__.py

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication\webapplication> python manage.py runserver
C:\Users\Akshida\AppData\Local\Programs\Python\Python311\python.exe:
can't open file 'C:\\Users\\Akshida\\Desktop\\slip\\webframework\\New
folder (2)\\pro\\webapplication\\webapplication\\manage.py': [Errno 2] No
such file or directory
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication\webapplication> cd..
```

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder
(2)\pro\webapplication> python manage.py runserver
```

```
Watching for file changes with StatReloader
Performing system checks...
```

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

December 28, 2022 - 15:12:55

Django version 4.1.4, using settings 'webapplication.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[28/Dec/2022 15:13:20] "GET / HTTP/1.1" 200 10681

[28/Dec/2022 15:13:20] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423

[28/Dec/2022 15:13:20] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692

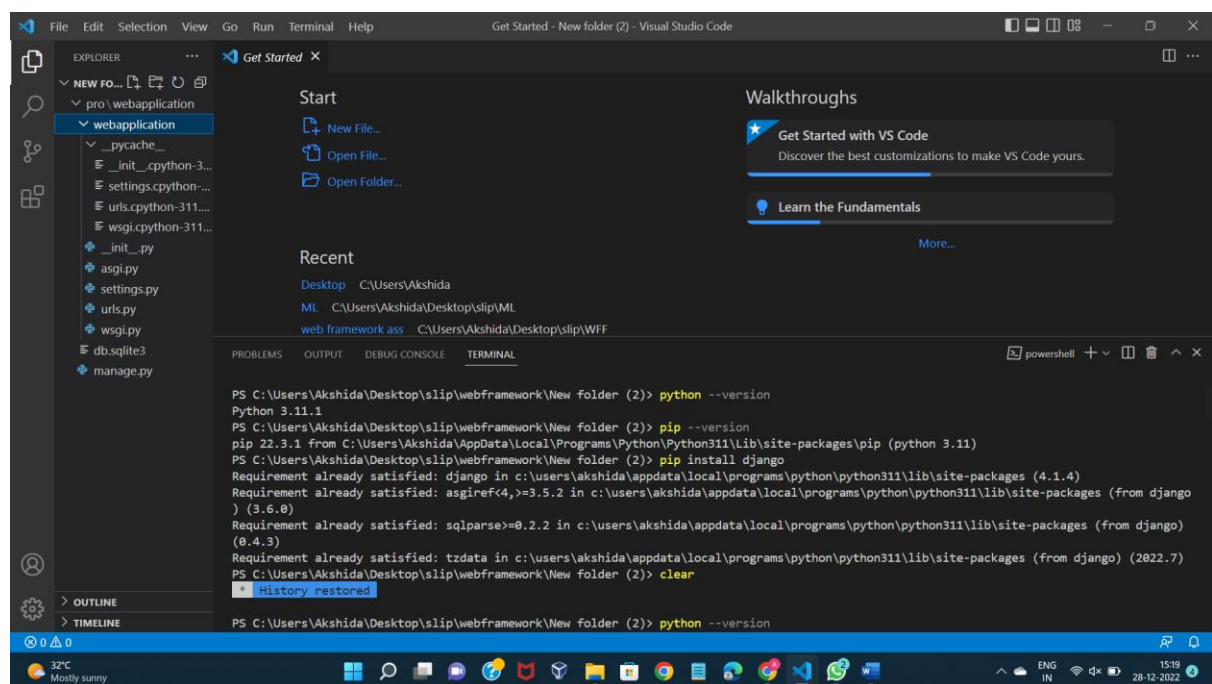
[28/Dec/2022 15:13:20] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876

[28/Dec/2022 15:13:20] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184

Not Found: /favicon.ico

[28/Dec/2022 15:13:21] "GET /favicon.ico HTTP/1.1" 404 2118

PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)\pro\webapplication>



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> python --version
Python 3.11.1
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip --version
pip 22.3.1 from C:\Users\Akshida\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> pip install django
Requirement already satisfied: django in c:\users\akshida\appdata\local\programs\python\python311\lib\site-packages (4.1.4)
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\akshida\appdata\local\programs\python\python311\lib\site-packages (from django) (3.6.0)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\akshida\appdata\local\programs\python\python311\lib\site-packages (from django) (0.4.3)
Requirement already satisfied: tzdata in c:\users\akshida\appdata\local\programs\python\python311\lib\site-packages (from django) (2022.7)
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> clear
History restored
PS C:\Users\Akshida\Desktop\slip\webframework\New folder (2)> python --version
```

The interface also shows the Explorer panel with a file tree for 'webapplication' and the Walkthroughs panel on the right.

