NAME: PREETI BARKUL

ROLL NO: SG001

SUBJECT: WEB FRAMEWORK

SLIP 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.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Slip 1 Name and Age Validation</title>
        <script src="Slip1.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>
            <input type="text" id = "output1"></input>
        </form>
    </body>
</html>
```

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]+$/;
```

```
    if(fname.length == 0 || lname.length == 0 || age.length
== 0){
        alert("All Fields are Mandatory");
        return false;
    }
    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{
        document.getElementById('output1').value = age + "
" + fname + " " + lname;
        return true;
    }
}
```

SLIP 2

CREATE AN HTML FORM THAT CONTAIN THE EMPLOYEE REGISTRATION DETAILS AND WRITE A JAVASCRIPT TO VALIDATE DOB, JOINING DATE, AND SALARY.

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>
```

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]+$/;
    var regdate = /^[0-9]{1,2}\/[0-9]{1,2}\/[0-9]{4}$/;
    var regsal = /^[0-9]$/;

    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;
    }
}
```

SLIP 3

CREATE AN HTML FORM FOR LOGIN AND WRITE A JAVASCRIPT TO VALIDATE EMAIL ID USING REGULAR EXPRESSION.

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>
```

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(email.length == 0){
        alert("All Fields are Mandatory");
        return false;
```

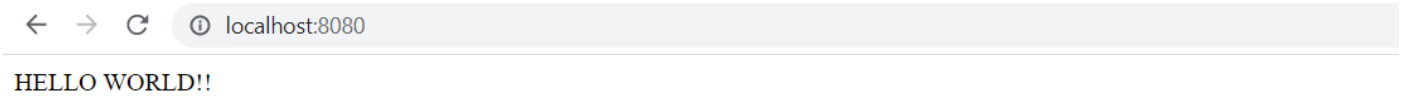
```
    }
    else if(pass.length == 0){
        alert("All Fields are Mandatory");
        return false;
    }
    else if(!regEmail.test(email)){
        alert("Enter Name Format Correctly First Name Last
Name");
        return false;
    }
    else{
        alert("Validation Successfull");
        return true;
    }
}
```

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);
```

Output

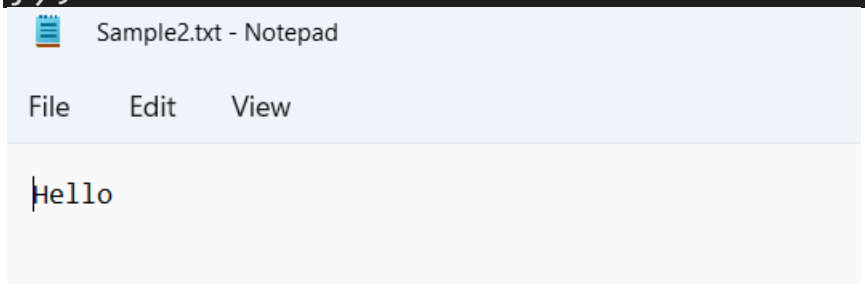


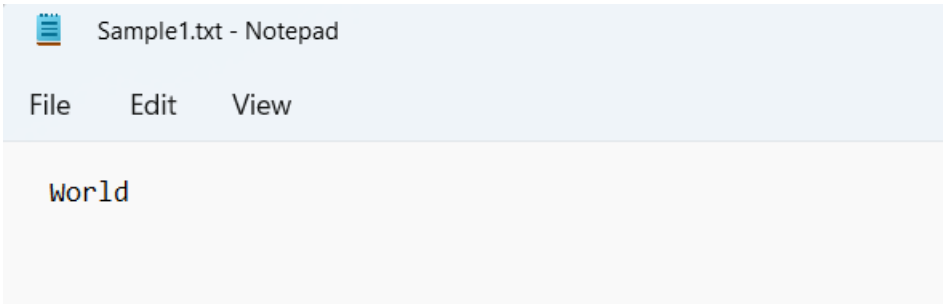
SLIP 5

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');
var path = require('path')

var data = fs.readFileSync("Sample1.txt","utf8");
```

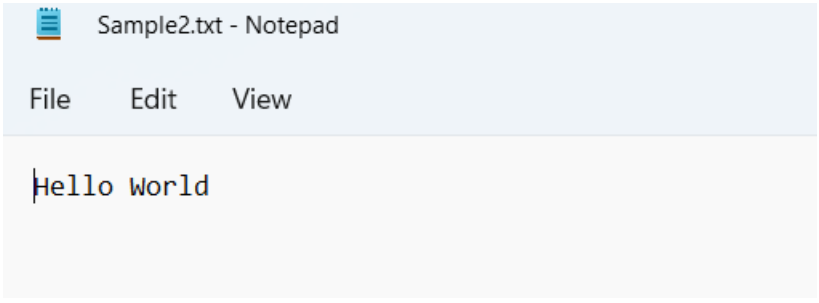
```
fs.appendFile("Sample2.txt", data,(err) =>{
    if(err) {
        console.log(err);
    }
    else{
        console.log("File Content after appending:
",fs.readFileSync("Sample2.txt","utf8"));
    }
});
```





Output

File Output



SLIP 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

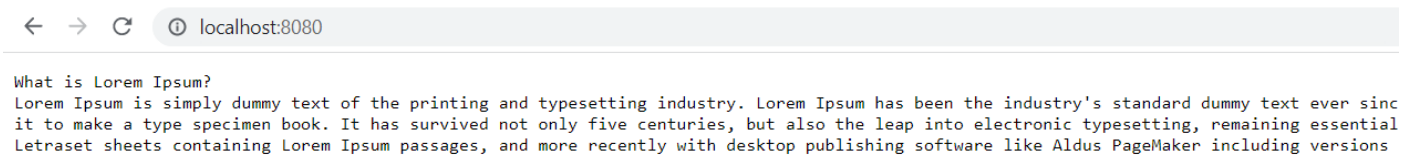
```
var http = require('http');
var fs = require('fs');
```

```
var server = http.createServer(function(req,res){
    fs.open('Sample.txt','r',function(err,fd){
        if(err){
            res.writeHead(404, {'Content-Type':
'text/html'});
            return res.end("404 File Not Found");
        }
        else{
            console.log("File Open Successfully");
            fs.readFile('Sample.txt',function(err,data){
                if(!err){
                    console.log('File Read Successfully');
                    res.end(data);
                    fs.close(fd);
                }
                else{
                    console.log('Read File is not
possible');
                }
            });
        }
    });
}).listen(8080);
```

Output

Case 1: File Not Present in Directory

Case 2: File Present in Directory



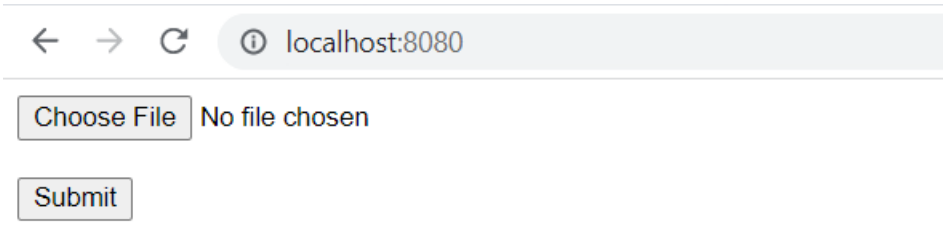
SLIP 7

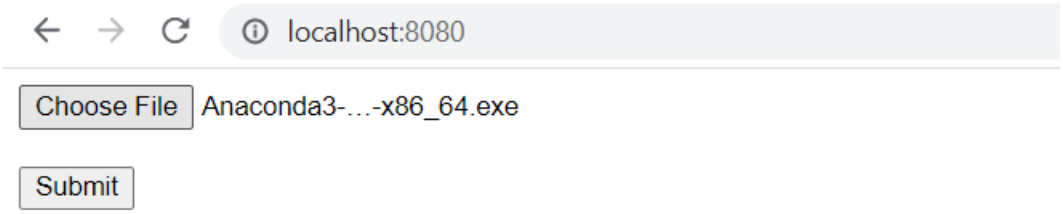
CREATE A NODE.JS FILE THAT WRITES AN HTML FORM, WITH AN UPLOAD FIELD

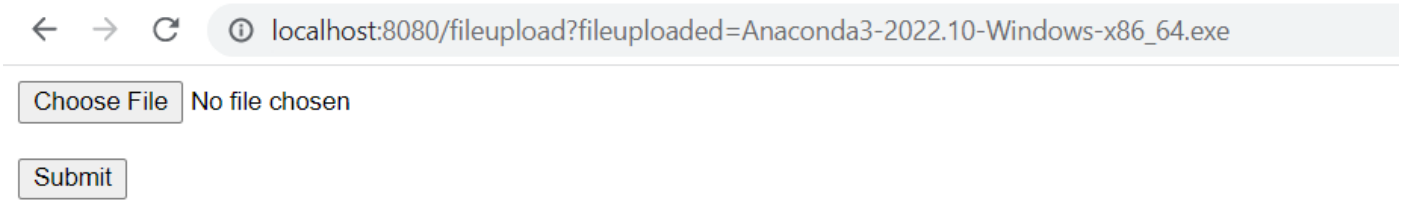
Slip7.js

```
var http = require('http');
var formidable = require('formidable');

http.createServer(function(req,res){
    var form = new formidable.IncomingForm();
    form.parse(req,function(err,fields,files){
        if(req.url=='/fileupload'){
            console.log(files);
            res.write('File Uploaded');
            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="fileuploaded"><br><br>');
            res.write('<input type = "submit">');
            res.write('</form>');
            return res.end();
        }
    });
}).listen(8080);
```

Output







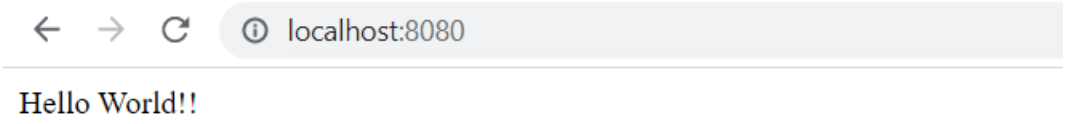
SLIP 12

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.write("Hello World!!");
    res.end();
}).listen(8080);
```

Output



SLIP 13

USING NODE JS CREATE A USER LOGIN SYSTEM

index.html

```
<html>
  <head>
  <title>Welcome Home</title>
  </head>
```

```
  <body>
  <a href="/login">Please Login Here!</a>
  </body>
</html>
```

login.html

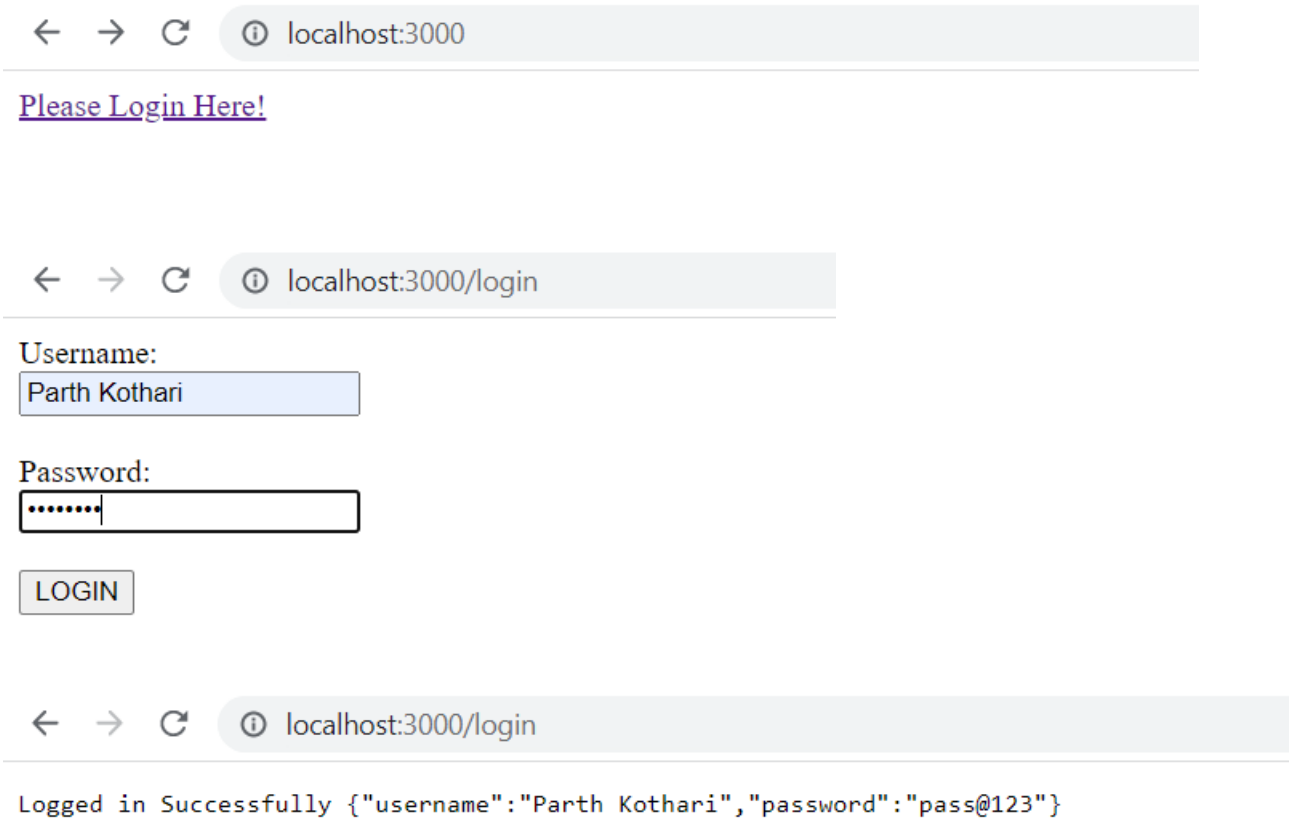
```
<html>
    <head>
        <title>LOGIN FORM</title>
    </head>
    <body>
        <form action="http://localhost:3000/login"
method="post">
            Username: <br> <input type="text"
name="usrname" required> <br><br>
            Password: <br> <input type="password"
name="pass" required> <br><br>
            <input type="submit" value="LOGIN">
        </form>
    </body>
</html>
```

login.js

```
var express = require('express');
var bodyParser = require('body-parser');
var app = express();

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

app.get('/',function(req,res){
    res.sendFile(__dirname + '/index.html');
});

app.get('/login',function(req,res){
    res.sendFile(__dirname + '/login.html');
});

app.post('/login',function(req,res){
```

```
    person = {
        username:req.body.usrname,
        password:req.body.pass
    };
    res.end("Logged in Successfully " +
JSON.stringify(person));
});

var server = app.listen(3000,function(){
    console.log("Server is Running");
});
```

Output



SLIP 16

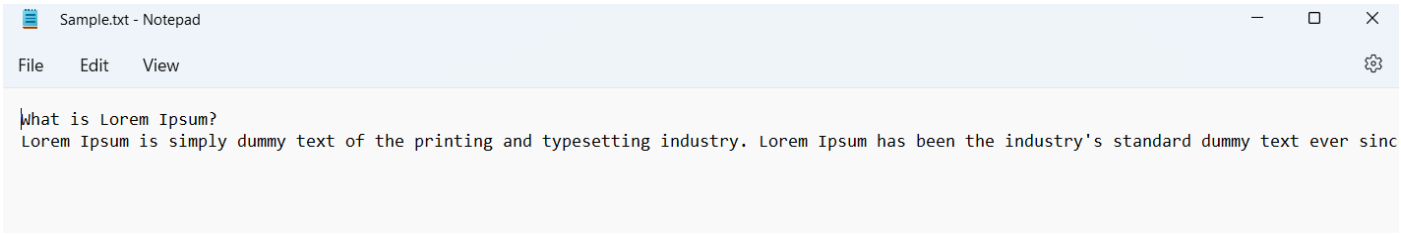
WRITE NODE JS SCRIPT TO INTERACT WITH THE FILESYSTEM, AND SERVE A WEB PAGE FROM A FILE

Slip16.js

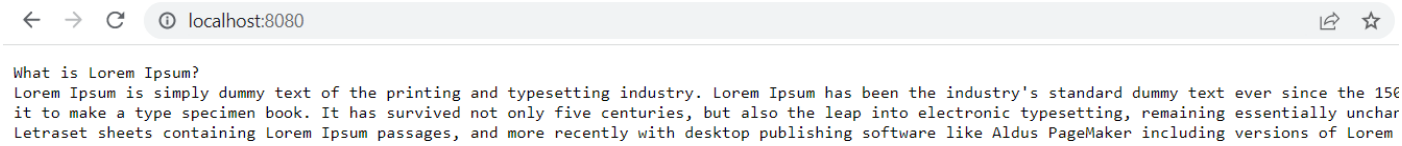
```
var http = require('http');
```

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

var server = http.createServer(function(req,res){
    //fs.open( filename, , mode[r = read & r+ = readwrite],
callback )
    fs.open('Sample.txt','r',function(err,fd){
        fs.readFile('Sample.txt',function(err,data){
            if(!err){
                console.log('File Read Successfully');
                res.end(data);
                fs.close(fd);
            }
            else{
                console.log('Read File is not possible');
            }
        });
    });
}).listen(8080);
```

Sample.txt



Output



Slip 17

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.

Slip17.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

SLIP 18

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.

```
VAR EVENTS = REQUIRE('events');

var em = new events.EventEmitter();

em.on('add', function(a,b){
    addition = a + b;
    console.log('The Addition of 2 Numbers is: '+addition);
});

em.emit('add',12,25);
```

Output

```
PS C:\Users\PARTH\Desktop\S.Y. Msc\WF\Practical\Prac\Slip18> node Slip18.js
Debugger attached.
The Addition of 2 Numbers is: 37
Waiting for the debugger to disconnect...
```

SLIP 20

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

greeting\views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def index(request):
    return HttpResponse('<i> Hello! I am learning
Django</i>')

# Create your views here.
```

greeting\urls.py

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

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

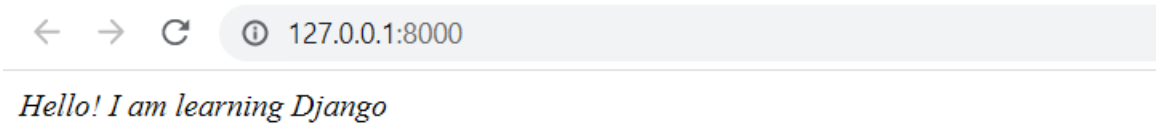
webapp1\urls.py

```
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'),
]
```

webapp1\settings.py

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

Output



Hello! I am learning Django