

ASSIGNMENT1

StudentName	RAMAKRISHNAN P
StudentRollNumber	95071912072
Team ID	PNT2022TMID53883

1.Create registration page in html with username, email and phone number and by using POST method display it in next html page.

Register.html

```
<html>

<head>

    <title>RegistrationPage</title>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

    <h1>RegistrationPage</h1>

    <form action="Register.php" method="POST">

        <label for="UserName">UserName:</label>

        <input type="text" name="UserName"><br/>

    <br/>

        <label for="Email">Email:</label>

        <input type="Email" name="Email"><br/>

    <br/>

        <label for="PhoneNumber">PhoneNumber:</label>

        <input type="PhoneNumber" name="PhoneNumber"><br/>

    <br/>

        <input type="submit" value="Register!">
```

```
</form>
```

```
</body>
```

```
</html>
```

Register.php

```
<?php
```

```
echo"Youhave submitted,UserName:".$_POST['UserName'].",Email:".
$_POST['Email'].",andPhoneNumber:".$_POST['PhoneNumber'];
```

```
?>
```

2.Developaflaskprogramwhichshouldcontainatleast5packagesusedfrompypi.org.

```
importos
```

```
importshutil
```

```
importpytest
```

```
from flaskimportrender_template,render_template_string,request
```

```
fromjinja2.exceptionsimportTemplateNotFound
```

```
from jinja2.sandboximportSecurityError
```

```
fromwerkzeug.testimportClient
```

```
from CTFd.configimportTestingConfig
```

```
from CTFd.utilsimportget_config,set_config
```

```
from tests.helpersimportcreate_ctfd,destroy_ctfd,gen_user,login_as_user
```

```
def test_themes_run_in_sandbox():
```

```
    app=create_ctfd()
```

```
    withapp.app_context():
```

```
        try:
```

```
            app.jinja_env.from_string(
```

```
                "{{().class.bases_____()[0].subclasses____()[40]('./test_utils.py').read()}}"
```

```
            ).render()
```

```

except SecurityError:
    pass

except Exception as e:
    raise e

destroy_ctfd(app)

def test_themes_cant_access_config_py_attributes():
    app = create_ctfd()
    with app.app_context():
        assert app.config["SECRET_KEY"] == "AAAAAAAAAAAAAAAAAAAAAA"
        assert(
            app.jinja_env.from_string("{% get_config('SECRET_KEY') %}").render()
        )
    ##...source file abbreviated to get to Flask examples...

    r = client.get("/challenges")
    assert r.status_code == 200
    assert "Challenges" in r.get_data(as_text=True)
    r = client.get("/scoreboard")
    assert r.status_code == 200
    assert "Scoreboard" in r.get_data(as_text=True)

destroy_ctfd(app)

def test_that_request_path_hijacking_works_properly():
    app = create_ctfd(setup=False, application_root="/ctf")
    assert app.request_class.name == "CTFdRequest"
    with app.app_context():
        with app.test_request_context("/challenges"):
            assert request.path == "/ctf/challenges"

destroy_ctfd(app)

```

```

app=create_ctfd()
assertapp.request_class.name=="CTFdRequest"
withapp.app_context():
    withapp.test_request_context("/challenges"):
        assertrequest.path=="/challenges"
from flaskimportFlask
test_app=Flask("test")
asserttest_app.request_class.name=="Request"
withtest_app.test_request_context("/challenges"):
    assertrequest.path=="/challenges"
destroy_ctfd(app)
deftest_theme_fallback_config():
    classThemeFallbackConfig(TestingConfig):
        THEME_FALLBACK=False
    app=create_ctfd(config=ThemeFallbackConfig)
    try:
        os.mkdir(os.path.join(app.root_path,"themes","foo_fallback"))
    exceptOSError:
        pass
    withapp.app_context():
        app.config["THEME_FALLBACK"]=False
        set_config("ctf_theme","foo_fallback")
        assertapp.config["THEME_FALLBACK"]==False
        withapp.test_client()asclient:
            try:
                r=client.get("/")

```