**A SS I GN M ENT1**

|  |  |
| --- | --- |
| StudentName | SUNDARAM V.A |
| StudentRollNumber | 950712099 |
| Team ID | PNT2022TMID53883 |

**1.Createregistrationpageinhtmlwithusername,emailandphonenumberandbyusingPOSTmethod displayitinnexthtmlpage.**

**Register.html**

<html>

<head>

<title>RegistrationPage</title>

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

</head>

<body>

<h1>RegistrationPage</h1>

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

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

<inputtype="text"name="UserName"><br/>

<br/>

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

<inputtype="Email"name="Email"><br/>

<br/>

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

<inputtype="PhoneNumber"name="PhoneNumber"><br/>

<br/>

<inputtype="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 fromflaskimportrender\_template,render\_template\_string,request fromjinja2.exceptionsimportTemplateNotFound from jinja2.sandboximportSecurityError fromwerkzeug.testimportClient from CTFd.configimportTestingConfig from CTFd.utilsimportget\_config,set\_config fromtests.helpersimportcreate\_ctfd,destroy\_ctfd,gen\_user,login\_as\_user deftest\_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()

exceptSecurityError: pass exceptExceptionase:

raisee destroy\_ctfd(app) def test\_themes\_cant\_access\_configpy\_attributes():

app=create\_ctfd() withapp.app\_context():

assertapp.config["SECRET\_KEY"]=="AAAAAAAAAAAAAAAAAAAA"

assert( app.jinja\_env.from\_string("{{get\_config('SECRET\_KEY')}}").render()

##...sourcefileabbreviatedtogettoFlaskexamples...

r=client.get("/challenges") assertr.status\_code ==200 assert"Challenges"inr.get\_data(as\_text=True) r=client.get("/scoreboard") assertr.status\_code==200 assert"Scoreboard"inr.get\_data(as\_text=True) destroy\_ctfd(app) deftest\_that\_request\_path\_hijacking\_works\_properly(): app=create\_ctfd(setup=False,application\_root="/ctf") assertapp.request\_class.nam e =="CTFdRequest" withapp.app\_context():

withapp.test\_request\_context("/challenges"):

assertrequest.path=="/ctf/challenges" destroy\_ctfd(app)

app=create\_ctfd() assertapp.request\_class.nam e =="CTFdRequest" withapp.app\_context():

withapp.test\_request\_context("/challenges"):

assertrequest.path=="/challenges" from flaskimportFlask test\_app=Flask("test") asserttest\_app.request\_class.nam e =="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("/")