

Includes fix for:

- /usr/bin/python: No module named http when running run.sh
- hello_get.py shows the whole python code after submitting a song
- OperationError after running python createTable.py or readTable.py

Create html and cgi-bin directory

Ubuntu 18.04 LTS

```
mva456@DESKTOP-OE7HLB3:~/cs522$ mkdir html
mva456@DESKTOP-OE7HLB3:~/cs522$ cd html
mva456@DESKTOP-OE7HLB3:~/cs522/html$ mkdir cgi-bin
mva456@DESKTOP-OE7HLB3:~/cs522/html$ cd cgi-bin
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$
```

Go to cgi-bin directory

Copy and paste code from GitHub then save

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code /

ch20 / [advcgi.py](#) / <> Jump to ▾

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ gedit advcgi.py
```

The screenshot shows a Windows desktop environment. A browser window is open to the GitHub page for 'advcgi.py' from the 'Core-Python-Programming-2nd-Edition-Examples-and-Source-Code' repository. Below the browser, a code editor window titled 'advcgi.py' is open, showing the Python code for the CGI application. A red arrow points from the browser window towards the code editor window, indicating the transition from viewing the code online to editing it locally.

```
#!/usr/bin/env python
from cgi import FieldStorage
from os import environ
from cStringIO import StringIO
from urllib import quote, unquote
from string import capwords, strip, split, join
class AdvCGI:
    header = 'Content-Type: text/html\n\n'
    url = '/py/advcgi.py'
    formhtml = '''<HTML><HEAD><TITLE>
Advanced CGI Demo</TITLE></HEAD>
<BODY><H2>Advanced CGI Demo Form</H2>
<FORM METHOD=post ACTION=%s ENCTYPE="multipart/form-data">
<H3>My Cookie Settings:</H3>
<L1> <CODE><B>CPPUser = %s</B></CODE>
<H3>Enter cookie value<BR>
<INPUT NAME=cookie value=%s> (<I>optional</I>)</H3>
<H3>Enter your name<BR>
<INPUT NAME=person VALUE=%s> (<I>required</I>)</H3>
<H3>What languages can you program in?
(<I>at least one required</I>)</H3>
%s
<H3>Enter file to upload</H3>
<INPUT TYPE=file NAME=upfile VALUE=%s SIZE=45>
<P><INPUT TYPE=submit>
</FORM></BODY></HTML>'''
    langSet = ('Python', 'PERL', 'Java', 'C++', 'PHP',
              'C', 'JavaScript')
    langItem = \
        '<INPUT TYPE=checkbox NAME=lang VALUE=%s> %s\n'
    def getCPPCookies(self):
        if environ.has_key('HTTP_COOKIE'):
            for eachCookie in map(strip,
                                  split(environ['HTTP_COOKIE'], ',')):
                if len(eachCookie) > 6 and \
                   eachCookie[:3] == 'CPP':
                    tag = eachCookie[3:7]
                    try:
                        self.cookies[tag] = \
                            eval(unquote(eachCookie[8:]))
                    except (NameError, SyntaxError):
                        self.cookies[tag] = \
                            eval(unquote(eachCookie[8:])))
    def getCPPCookies(self):
        # reads cookies from client
        if environ.has_key('HTTP_COOKIE'):
            for eachCookie in map(strip,
                                  split(environ['HTTP_COOKIE'], ',')):
                if len(eachCookie) > 6 and \
                   eachCookie[:3] == 'CPP':
                    tag = eachCookie[3:7]
                    try:
                        self.cookies[tag] = \
                            eval(unquote(eachCookie[8:]))
                    except (NameError, SyntaxError):
                        self.cookies[tag] = \
                            eval(unquote(eachCookie[8:])))
```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code /

ch20 / [friends1.py](#) / <> Jump to ▾

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ gedit friends1.py
```

```

#!/usr/bin/env python
import cgi

reshtml = '''Content-Type: text/html\n
<HTML><HEAD><TITLE>
Friends CGI Demo (dynamic screen)
</TITLE></HEAD>
<BODY><H3>Friends list for: <I>%s</I></H3>
Your name is: <B>%s</B><br>
You have <B>%s</B> friends.
</BODY></HTML>'''

form = cgi.FieldStorage()
who = form['person'].value
howmany = form['howmany'].value
print reshtml % (who, who, howmany)

```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code / ch20 / friends2.py / < Jump to ▾

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ gedit friends2.py
```

```

#!/usr/bin/env python
import cgi
header = 'Content-Type: text/html\n\n'

formhtml = '''<HTML><HEAD><TITLE>
Friends CGI Demo</TITLE></HEAD>
<BODY><H3>Friends list for: <I>NEW USER</I></H3>
<FORM ACTION="/cgi-bin/friends2.py">
<B>Enter your Name:</B>
<INPUT TYPE=hidden NAME=action VALUE=edit>
<INPUT TYPE=text NAME=person VALUE="NEW USER" SIZE=15>
<P><B>How many friends do you have?</B>
<S>
<P><INPUT TYPE=submit></FORM></BODY></HTML>'''

radio = '<INPUT TYPE=radio NAME=howmany VALUE="%s" %s> %s\n'

def showForm():
    friends = ''
    for i in [0, 10, 25, 50, 100]:
        checked = ''
        if i == 0:
            checked = 'CHECKED'
        friends = friends + radio % (str(i), checked, str(i))

    print header + formhtml % (friends)

reshtml = '''<HTML><HEAD><TITLE>
Friends CGI Demo</TITLE></HEAD>
<BODY><H3>Friends list for: <I>%s</I></H3>
Your name is: <B>%s</B><br>
You have <B>%s</B> friends.
</BODY></HTML>'''

def doResults(who, howmany):
    print header + reshtml % (who, who, howmany)

def process():
    form = cgi.FieldStorage()
    if form.has_key('person'):
        who = form['person'].value
    else:
        who = 'NEW USER'

    if form.has_key('howmany'):

```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code / ch20 / friends3.py / < Jump to ▾

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ gedit friends3.py
```

```

for i in [8, 10, 25, 50, 100]:
    checked = ''
    if str(i) == hommany:
        checked = 'CHECKED'
    friends = friends + fradio % \
        (str(i), checked, str(i))
    print header + formhtml % (who, url, who, friends)

resthtml = '''<HTML><HEAD><TITLE>
Friends CGI Demo</TITLE></HEAD>
<BODY><H3>Friends list for: <I>%s</I></H3>
Your name is: <B>%s</B>
You have <B>%s</B> friends.
<P><A HREF="#">Here</A> to edit your data again.
</BODY></HTML>'''

def doResults(who, hommany):
    newurl = url + '?action=edit&person=%s&hommany=%s' % \
        (quote_plus(who), hommany)
    print header + resthtml % (who, who, hommany, newurl)

def process():
    error = ''
    form = cgi.FieldStorage()
    if form.has_key('person'):
        who = capwords(form['person'].value)
    else:
        who = 'NEW USER'

    if form.has_key('hommany'):
        hommany = form['hommany'].value
    else:
        if form.has_key('action') and \
            form['action'].value == 'edit':
            error = 'Please select number of friends.'
        else:
            hommany = 0

    if not error:
        if form.has_key('action') and \
            form['action'].value != 'redit':
            doResults(who, hommany)
        else:
            showForm(who, hommany)
    else:
        showError(error)

formhtml = '''<HTML><HEAD><TITLE>
Friends CGI Demo</TITLE></HEAD>
<BODY><H3>Friends list for: <I>%s</I></H3>
<FORM ACTION="#">
<B>Your Name:</B>
<INPUT TYPE=hidden NAME=action VALUE=edit>
<INPUT TYPE=text NAME=person VALUE="%s" SIZE=15>
</FORM>
</BODY></HTML>'''

def showError(error_str):
    print header + erthtml % (error_str)

def showForm(who, hommany):
    print header + formhtml % (hommany)

def main():
    error = ''
    if form.has_key('action') and \
        form['action'].value == 'redit':
        doResults(who, hommany)
    else:
        showForm(who, hommany)

if __name__ == '__main__':
    main()

```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code /
ch20 / **myhttpd.py** / <> Jump to ▾

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin\$ gedit myhttpd.py

```

#!/usr/bin/env python

from os import curdir, sep
from BaseHTTPServer import \
    BaseHTTPRequestHandler, HTTPServer
class MyHandler(BaseHTTPRequestHandler):

    def do_GET(self):
        try:
            f = open(curdir + sep + self.path)
            self.send_response(200)
            self.end_headers()
            self.wfile.write(f.read())
            f.close()
        except IOError:
            self.send_error(404,
                           'File Not Found: %s' % self.path)

    def main():
        try:
            server = HTTPServer(('', 80), MyHandler)
            print 'Welcome to the machine...'
            print 'Press ^C once or twice to quit'
            server.serve_forever()
        except KeyboardInterrupt:
            print '^C received, shutting down server'
            server.socket.close()

if __name__ == '__main__':
    main()

```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code /
ch20 / **uniCGI.py** / <> Jump to ▾

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin\$ gedit uniCGI.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
CODEC = 'UTF-8'
UNICODE_HELLO = u'Hello!\n\u00A1Hola!\n\u4F60\u597D!\n\u3053\u3093\u306B\u3061\u306F!\n...'
print 'Content-Type: text/html; charset=%s\r' % CODEC
print '\r'
print '<HTML><HEAD><TITLE>Unicode CGI Demo</TITLE></HEAD>'
print '<BODY>'
print UNICODE_HELLO.encode(CODEC)
print '</BODY></HTML>'

16 lines (14 sloc) 326 Bytes
```

Core-Python-Programming-2nd-Edition-Examples-and-Source-Code / ch20 / urlopenAuth.py / <> Jump to ▾

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin\$ gedit urlopenAuth.py

```
#!/usr/bin/env python
# encoding: utf-8
import urllib2

LOGIN = 'wesley'
PASSWD = "you'llNeverGuess"
URL = 'http://localhost'

def handler_version(url):
    from urlparse import urlparse as up
    hdrl = urllib2.HTTPBasicAuthHandler()
    hdrl.add_password('Archives', up(url)[1], LOGIN, PASSWD)
    opener = urllib2.build_opener(hdrl)
    urllib2.install_opener(opener)
    return url

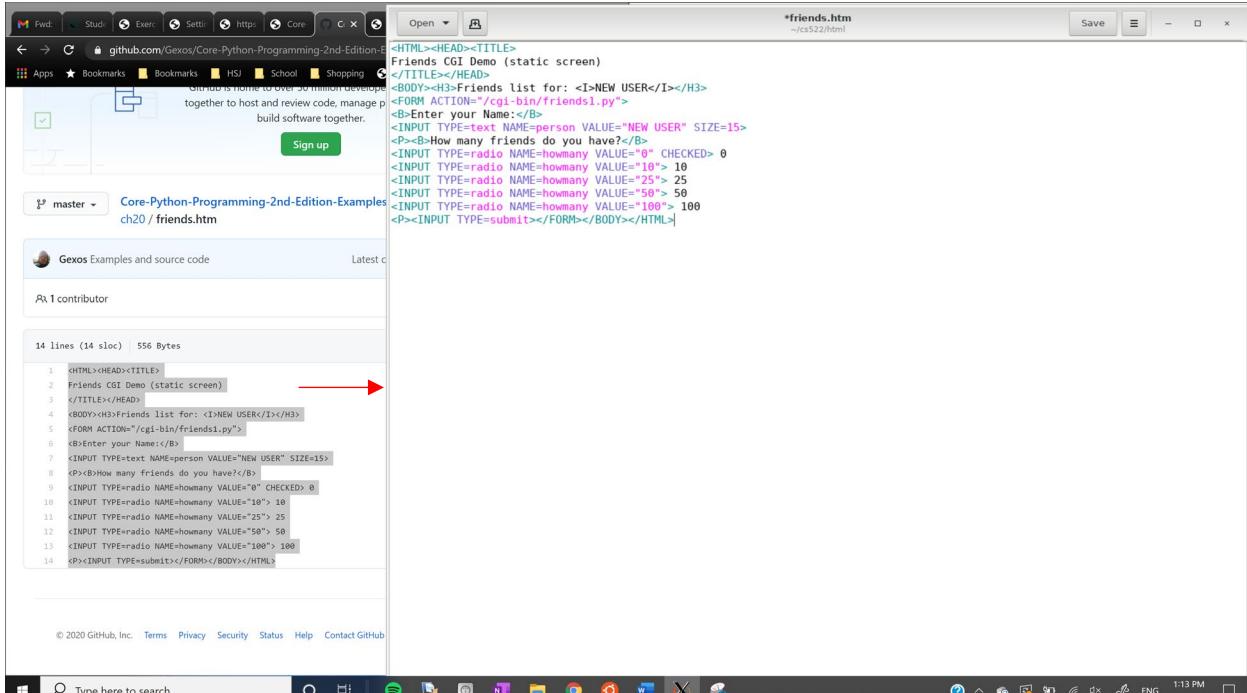
def request_version(url):
    from base64 import encodestring
    req = urllib2.Request(url)
    b64str = encodestring('%s:%s' % (LOGIN, PASSWD))[:-1]
    req.add_header("Authorization", "Basic %s" % b64str)
    return req

for funcType in ('handler', 'request'):
    print "*** Using %s: %s" % (funcType.upper(), funcType)
    url = eval("%s_version")(URL)
    f = urllib2.urlopen(url)
    print f.read()
    f.close()

for funcType in ('handler', 'request'):
    print "*** Using %s: %s" % (funcType.upper(), funcType)
    url = eval("%s_version")(URL)
    f = urllib2.urlopen(url)
    print f.readline()
```

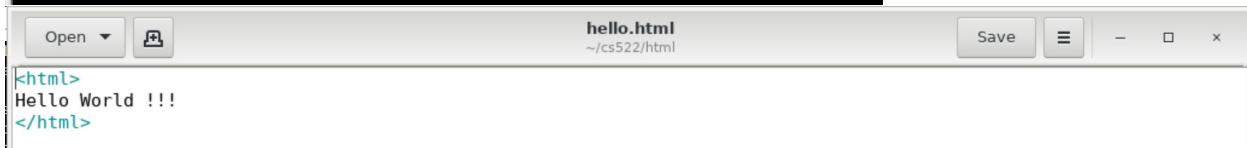
Core-Python-Programming-2nd-Edition-Examples-and-Source-Code /
 ch20 / friends.htm

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ cd ..
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit friends.htm
```



```
<HTML><HEAD><TITLE>
Friends CGI Demo (static screen)
</TITLE></HEAD>
<BODY><H3>Friends list for: <?>NEW USER</?></H3>
<FORM ACTION="/cgi-bin/friends1.py">
<INPUT TYPE="text" NAME="person" VALUE="NEW USER" SIZE=15>
<P>Enter your Name:</P>
<INPUT TYPE="radio" NAME="howmany" VALUE="0" CHECKED> 0
<INPUT TYPE="radio" NAME="howmany" VALUE="10"> 10
<INPUT TYPE="radio" NAME="howmany" VALUE="25"> 25
<INPUT TYPE="radio" NAME="howmany" VALUE="50"> 50
<INPUT TYPE="radio" NAME="howmany" VALUE="100"> 100
<P><INPUT TYPE="submit"></FORM></BODY></HTML>
```

```
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit hello.html
```



```
<html>
Hello World !!!
</html>
```

```
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit simple_form.html
```

```
<!DOCTYPE html>
<html>
<!--
If you select Song 1 and press submit, the client will
make the following request to the server
    http://&tdserver_doman_name//prog.py?song=song1
-->
<head>
<title>Online Jukebox Form</title>
</head>
<body>
<a id="link"/>Select a song:</a>
<form action="/cgi-bin/hello_get.py" method="get">
<input type="radio" id="song1" name="song" value="song1">Song 1<br />
<input type="radio" id="song2" name="song" value="song2">Song 2<br />
```

```

<input type="radio" id="song3" name="song" value="song3">Song 3<br />
<br />
<input type="submit" value="Submit" id="button1"></input>
<input type="reset">
</form>
</body>
</html>

```

```
mva456@DESKTOP-0E7HLB3:~/cs522/html$ cd cgi-bin
mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin$ gedit hello_get.py
```

```

#!/usr/bin/python
# Import modules for CGI handling
import cgi, cgitb
import pymysql, requests, json

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
song = form.getvalue('song')
captchaResponse = form.getvalue('g-recaptcha-response')

def is_human(captcha_response):
    """ Validating recaptcha response from google server
        Returns True captcha test passed for submitted form else returns
    False.
    """
    secret = "6LdiFeQZAAAAANFnkZYrr7aHh0V0dKmylwPmAEl0"
    payload = {'response':captcha_response, 'secret':secret}
    response =
    requests.post("https://www.google.com/recaptcha/api/siteverify",
                  payload)
    response_text = json.loads(response.text)
    return response_text['success']
if True: #is_human(captchaResponse):
    print "Content-type:text/html\r\n\r\n"
    print "<html>"
    #print "<head>"
    #print "<title>Online Jukebox</title>"
    #print "</head>"
    print "<body>"
    print "<h2>Selected song is %s</h2>" % (song)
    print "</body>"
    print "</html>"
    db = pymysql.connect(host="localhost", user="root",
passwd="12345",db="mydb" )

```

```

# Prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = "INSERT INTO songList(song) VALUES (%s)"
try:
    # Execute the SQL command
    cursor.execute(sql,(song))
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
else:
    print "Content-type:text/html\r\n\r\n"
    print "<html>"
    print "<head>"
    print "<title>Online Jukebox</title>"
    print "</head>"
    print "<body>"
    print "<h2>Failed captcha. Please retry.</h2>"
    print "</body>"
    print "</html>"
```

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ chmod a+x hello_get.py
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ cd ..
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit run.sh
```

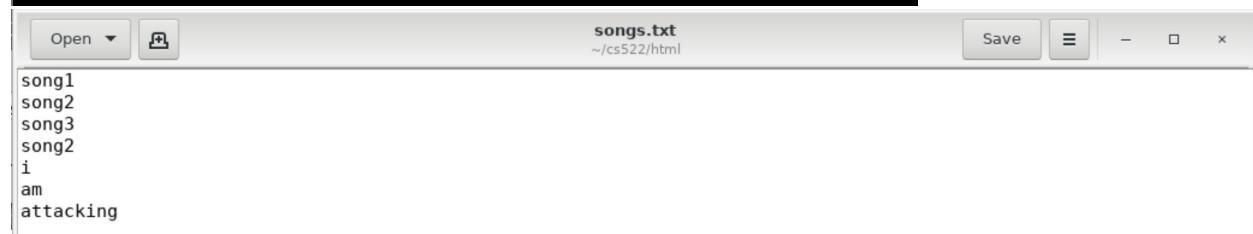
To serve all IP addresses



```
run.sh
~/cs522/html
Open Save - x
python3 -m http.server --cgi 8000
# python3 -m http.server --bind localhost --cgi 8000
# python -m http.server --cgi 7000
# python -m http.server --cgi
```

DOS attack's random words that will be used

```
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit songs.txt
```



```
songs.txt
~/cs522/html
Open Save - x
song1
song2
song3
song2
i
am
attacking
```

DOS script used to automate attack

```
mva456@DESKTOP-OE7HLB3:~/cs522/html$ gedit dos.sh
```

```
dos.sh
~/cs522/html
#!/bin/bash
random()
{
    start_time=$1
    end_time=$2
    DIFF=$((($end_time-$start_time+1)))
    echo $((($RANDOM%$DIFF)+$start_time))
}
start_time=1
end_time=25
RANDOM=$$
cat songs.txt | while read song
do
    sleep_time=`random $start_time $end_time`
    sleep $sleep_time
    wget "http://localhost:8000/cgi-bin/hello_get.py?song=${song}"
done
```

Create database table to store data

- Check for updates then install pip, pymysql and requests:
 - sudo apt-get update -y
 - sudo apt-get install pymysql
 - sudo apt-get install -y python-requests
 - sudo apt-get install python-mysqldb
 - sudo apt-get install mysql-server
 - sudo apt-get install python3-pip
 - pip install mysql-connector-python

Find out pymysql path

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ pip3 show pymysql
Name: PyMySQL
Version: 0.10.1
Summary: Pure Python MySQL Driver
Home-page: https://github.com/PyMySQL/PyMySQL/
Author: yutaka.matsubara
Author-email: yutaka.matsubara@gmail.com
License: "MIT"
Location: /home/mva456/.local/lib/python3.6/site-packages
Requires:
```

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ gedit createTable.py
```

createTable.py

```

import sys
sys.path.append('/home/mva456/.local/lib/python3.6/site-packages')
import pymysql
db = pymysql.connect(host="localhost", user="root", passwd="12345", db="mydb" )

# Prepare a cursor object using cursor() method
cursor = db.cursor()

# Prepare SQL query to Create a songList table into the database.
sql ="CREATE TABLE songList(song varchar(32))"

try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()

# disconnect from server
db.close()

```

mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin\$ gedit readTable.py

readTable.py

```

#!/usr/bin/python
import sys
sys.path.append('/home/mva456/.local/lib/python3.6/site-packages')
import pymysql

# Open database connection
db = pymysql.connect(host="localhost", user="root", passwd="12345", db="mydb" )

# Prepare a cursor object using cursor() method
cursor = db.cursor()

# Prepare SQL query to INSERT a record into the database.
sql = "SELECT * FROM songList;"
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Fetch all the rows in a list of lists.
    results = cursor.fetchall()
    for row in results:
        song = row[0]
        # Now print fetched result
        print ("song = %s" % (song))
except:
    print ("Error: unable to fetch data")

# disconnect from server
db.close()

```

Add the import sys path to hello_get.py

mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin\$ gedit hello_get.py

```

# Import modules for CGI handling
import cgi, cgitb
import sys
sys.path.append('/home/mva456/.local/lib/python3.6/site-packages/')
import pymysql, requests, json

```

Create recaptcha

Google reCAPTCHA

M

← Register a new site

Label (i)

Online Jukebox

14 / 50

reCAPTCHA type (i)

reCAPTCHA v3 Verify requests with a score

reCAPTCHA v2 Verify requests with a challenge

"I'm not a robot" Checkbox Validate requests with the "I'm not a robot" checkbox

Invisible reCAPTCHA badge Validate requests in the background

reCAPTCHA Android Validate requests in your android app

Domains (i)

+ localhost

Owners

mavamontaos@gmail.com (You)

+• Enter email addresses h

Accept the reCAPTCHA Terms of Service Privacy - Terms

Adding reCAPTCHA to your site

'Online Jukebox' has been registered.

Use this site key in the HTML code your site serves to users. See client side integration

key COPY SITE KEY

6LdiFeQZAAAAANg2xtlyeCzRDxqxNlje3wSSYGe8

Use this secret key for communication between your site and reCAPTCHA. See server side integration

key COPY SECRET KEY

6LdiFeQZAAAANFnkZYrr7aHhOV0dKmylwPmAElo

[GO TO SETTINGS](#)

[GO TO ANALYTICS](#)

Add recaptcha in hello_get.py

```
*hello_get.py
~/cs522/html/cgi-bin

#!/usr/bin/python
# Import modules for CGI handling
import cgi, cgitb
import sys
sys.path.append('/home/mva456/.local/lib/python3.6/site-packages')
import pymysql, requests, json

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
song = form.getvalue('song')
captchaResponse = form.getvalue('g-recaptcha-response')

def is_human(captcha_response):
    """ Validating recaptcha response from google server
    Returns True captcha test passed for submitted form else returns False.
    """
    secret = "6LdiFeQZAAAAANFnkZYrr7aHh0V0dKmylwPmAEl0"
    payload = {'response':captcha_response, 'secret':secret}
    response = requests.post("https://www.google.com/recaptcha/api/siteverify",
    payload)
    response_text = json.loads(response.text)
    return response_text['success']

if True: #is_human(captchaResponse):
    print "Content-type:text/html\r\n\r\n"
    print "<html>"
    #print "<head>"
    #print "<title>Online Jukebox</title>"
    #print "</head>"
    print "<body>"
    print "<h2>Selected song is %s</h2>" % (song)
    print "</body>"
    print "</html>"
    db = pymysql.connect(host="localhost", user="root", passwd="12345",db="mydb" )
    # Prepare a cursor object using cursor() method
    cursor = db.cursor()
    # Prepare SQL query to INSERT a record into the database.
    sql = "INSERT INTO songList(song) VALUES (%s)"
    try:
        # Execute the SQL command
        cursor.execute(sql,(song))
        # Commit your changes in the database
        db.commit()
    except:
        # Rollback in case there is any error
        db.rollback()
    # disconnect from server
```

The screenshot shows a dual-pane interface. On the left, a web browser window displays the Google reCAPTCHA setup page for 'Online Jukebox'. It shows the site key ('6LdiFeQZAAAAANG2xtlyeCzRDxqxNIje3wSSYGe8') and secret key ('6LdiFeQZAAAAANFkZYr7ahh0V0dkaylwPmAElo'). On the right, a code editor window titled 'hello_get.py' contains Python code for handling reCAPTCHA responses. A red arrow points from the site key in the browser to the 'secret' variable in the Python code.

```
#!/usr/bin/python
# Import modules for CGI handling
import cgi, cgitb
import pymysql, requests, json
# Create instance of FieldStorage
form = cgi.FieldStorage()
# Get data from fields
song = form.getvalue('song')
captchaResponse = form.getvalue('g-recaptcha-response')
isHuman(captchaResponse)
"""
    """ Validating reCAPTCHA response from google server
    Returns True captcha test passed for submitted form else returns False.
"""

secret = "6LdiFeQZAAAAANFkZYr7ahh0V0dkaylwPmAElo"
payload = {'response':captchaResponse, 'secret':secret}
response = requests.post("https://www.google.com/recaptcha/api/siteverify",
payload)
response_text = json.loads(response.text)
return response_text['success']
if True:#is_human(captchaResponse):
    print("Content-type:text/html\r\n\r\n")
    print "<html>"
    #print "<head>"
    #print "<title>Online Jukebox</title>"
    #print "</head>"
    print "<body>"
    print "<h2>Selected song is %s</h2>" % (song)
    print "<p>Selected song is %s</p>" % (song)
    print "</body>"
    print "</html>"
db = pymysql.connect("localhost","root","12345","mydb" )
# Prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to INSERT a record into the database.
sql = "INSERT INTO songList(songName) VALUES (%s)"
try:
    # Execute the SQL command
    cursor.execute(sql,(song,))
    # Commit your changes in the database
    db.commit()
except:
    print(cursor.statement)
```

Add reCAPTCHA in simple_form.html

The screenshot shows a code editor with 'simple_form.html' open. A red box highlights the JavaScript code for reCAPTCHA and the corresponding HTML input field. Another red box highlights the 'onclick' event handler for the submit button.

```
<!DOCTYPE html>
<html>
<!--
If you select Song 1 and press submit, the client will
make the following request to the server
    http://&tdserver_doman_name//prog.py?song=song1
-->
<head>
<title>Online Jukebox Form</title>
</head>
<body>
<script src="https://www.google.com/recaptcha/api.js"></script>
<script>
function checkRecaptcha() {
    var response = grecaptcha.getResponse();
    if(response.length == 0) {
        //reCaptcha not verified
        alert("fail");
    }
    else {
        //reCaptcha verified
        alert("pass");
    }
}</script>

<a id="link">Select a song:</a>
<form action="/cgi-bin/hello_get.py" method="get">
<input type="radio" id="song1" name="song" value="song1">Song 1<br />
<input type="radio" id="song2" name="song" value="song2">Song 2<br />
<input type="radio" id="song3" name="song" value="song3">Song 3<br />
<div class="g-recaptcha" data-sitekey="6LdiFeQZAAAAANG2xtlyeCzRDxqxNIje3wSSYGe8" ></div>
<br />
<input type="submit" value="Submit" id="button1" onclick="checkRecaptcha();"></input>
<input type="reset">
</form>
</body>
</html>
```

```

<!DOCTYPE html>
<html>
<head>
<title>Online Jukebox Form</title>
<script type="text/javascript">
function displayResult(song)
{
    //document.getElementById(song).href = "http://www.npu.edu";
    document.getElementById("song1").checked = false;
    document.getElementById("song2").checked = false;
    document.getElementById("song3").checked = false;
    document.getElementById(song).checked = true;
    alert(song);
}
</script>
</head>
<body>
    <a id="link">Select a song:</a>
    <form action="/cgi-bin/hello_get.py" method=get>
        <input type="radio" id="song1" name=song onclick="displayResult(this.value)" value="song1">Song
    1<br />
        <input type="radio" id="song2" name=song onclick="displayResult(this.value)" value="song2">Song
    2<br />
        <input type="radio" id="song3" name=song onclick="displayResult(this.value)" value="song3">Song
    3<br />
        <div class="g-recaptcha" data-sitekey="6LdiFeQZAAAAANg2xtlyeCzRDxqxNIje3wSSYGe8" style="border: 1px solid #ccc; width: 250px; height: 40px; margin-top: 10px;">

```

Prepare database

* Should fix OperationError after running python createTable.py or readTable.py

```

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE mydb;
Query OK, 1 row affected (0.00 sec)

mysql> USE mydb;
Database changed
mysql> SHOW TABLES;
Empty set (0.01 sec)

mysql> exit
Bye

```

```

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin$ python createTable.py

```

```

mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin$ mysql -u root -p

```

```

mysql> USE mydb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| songList        |
+-----+
1 row in set (0.00 sec)

mysql> exit

```

Run website (Do not close this window)

```

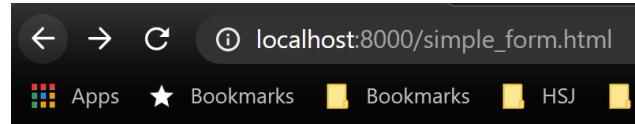
mva456@DESKTOP-OE7HLB3:~/cs522/html$ chmod 755 run.sh
mva456@DESKTOP-OE7HLB3:~/cs522/html$ ./run.sh
[...]

```

Fix for error /usr/bin/python: No module named http when running run.sh:

- In run.sh, change python to python3
- Save and exit
- chmod 755 run.sh
- ./run.sh

Open your browser and select a song

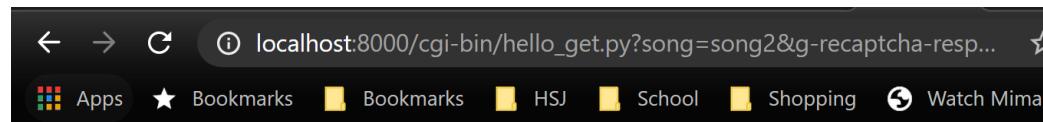


Select a song:

- Song 1
 Song 2
 Song 3

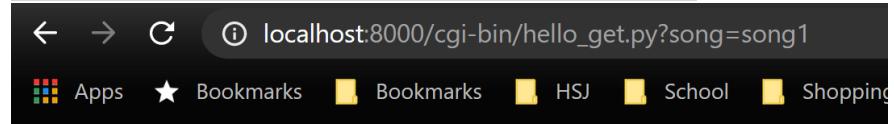
I'm not a robot

reCAPTCHA
[Privacy](#) - [Terms](#)



Selected song is song2

Fix for error (refer to image below) after submitting song:



```
#!/usr/bin/python

# Import modules for CGI handling
import cgi, cgitb

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
song1 = form.getvalue('song1')
song2 = form.getvalue('song2')
song3 = form.getvalue('song3')

print "Content-type:text/html\r\n\r\n"
print "<html>"
print "<head>"
print "<title>Online Jukebox</title>"
print "</head>"
print "<body>"
print "<h2>Selected song is %s %s %s</h2>" % (song1, song2, song3)
print "</body>"
print "</html>"



- Edit run.sh
- Add --cgi



---



```
python3 -m http.server --cgi 8000
python3 -m http.server --bind localhost --cgi 8000
python -m http.server --cgi 7000
python -m http.server --cgi
```



- Edit hello_get.py
- Change form values to song

```

```
#!/usr/bin/python

# Import modules for CGI handling
import cgi, cgitb

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
song = form.getvalue('song')
#song2 = form.getvalue('song2')
#song3 = form.getvalue('song3')

print "Content-type:text/html\r\n\r\n"
print "<html>"
print "<head>"
print "<title>Online Jukebox</title>"
print "</head>"
print "<body>"
print "<h2>Selected song is %s</h2>" % (song)
print "</body>"
print "</html>"
```

Open a new Ubuntu window

```
mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ python readTable.py
song = song2
```

Try DOS attack by typing ./dos.sh under html directory

```

Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'hello_get.py?song=song3.1'

hello_get.py?song=song3.1      [ <=> ]       63  --.-KB/s  in 0s

2020-11-17 15:28:53 (3.51 MB/s) - 'hello_get.py?song=song3.1' saved [63]

--2020-11-17 15:29:01-- http://localhost:8000/cgi-bin/hello_get.py?song=song2
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'hello_get.py?song=song2.4'

hello_get.py?song=song2.4      [ <=> ]       63  --.-KB/s  in 0s

2020-11-17 15:29:02 (2.61 MB/s) - 'hello_get.py?song=song2.4' saved [63]

--2020-11-17 15:29:25-- http://localhost:8000/cgi-bin/hello_get.py?song=i
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'hello_get.py?song=i.1'

hello_get.py?song=i.1      [ <=> ]       59  --.-KB/s  in 0s

2020-11-17 15:29:25 (2.50 MB/s) - 'hello_get.py?song=i.1' saved [59]

--2020-11-17 15:29:47-- http://localhost:8000/cgi-bin/hello_get.py?song=am
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'hello_get.py?song=am.1'

hello_get.py?song=am.1      [ <=> ]       60  --.-KB/s  in 0s

2020-11-17 15:29:47 (3.41 MB/s) - 'hello_get.py?song=am.1' saved [60]

--2020-11-17 15:30:09-- http://localhost:8000/cgi-bin/hello_get.py?song=attacking
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'hello_get.py?song=attacking.1'

hello_get.py?song=attacking      [ <=> ]       67  --.-KB/s  in 0s

2020-11-17 15:30:10 (3.45 MB/s) - 'hello_get.py?song=attacking.1' saved [67]

```

Result of DOS attack

```

mva456@DESKTOP-OE7HLB3:~/cs522/html/cgi-bin$ python readTable.py
song = song1
song = song1
song = song2
song = song3
song = song2
song = i
song = am
song = attacking

```

Enable recaptcha function to avoid DOS attack

```
mva456@DESKTOP-0E7HLB3:~/cs522/html/cgi-bin$ gedit hello_get.py
```

```
def is_human(captcha_response):
    """ Validating recaptcha response from google server
    Returns True captcha test passed for submitted form else returns False.
    """
    secret = "6LdiFeQZAAAAANFnkZYrr7aHh0V0dKmylwPmAElo"
    payload = {'response':captcha_response, 'secret':secret}
    response = requests.post("https://www.google.com/recaptcha/api/siteverify",
    payload)
    response_text = json.loads(response.text)
    return response_text['success']

if is_human(captchaResponse):
    print "Content-type:text/html\r\n\r\n"
    print "<html>"
    #print "<head>"
```

Submitting a song without answering recaptcha or doing DOS attack should not add more songs under readTable.py

