Activity - 02 : Resource Creation Assignment (RCA)

Name: A Kamil Khan
Roll No: P191311

Q1. Explain how to create CGI bin directory (Write or type the steps and send as image (if you written)or pdf(if you typed))

Steps:

1. Apache Server is running in my mac machine in this folder /private/etc/apache2
2. I create a folder called /Users/kkhan/MSC/WebTechnology/CGI under my home directory.
3. edit the file httpd.conf under apache folder. Make the following entries.

3a. Add Entry for Script Alias Match.

```
       <IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server
scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to
the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAliasMatch ^/cgi-bin/((?!(?i:webobjects)).*$) "/Users/kkhan/
MSC/WebTechnology/CGI/$1"

</IfModule>
```

3b. Give the required permission for this directory.

```
#
# "/Library/WebServer/CGI-Executables" should be changed to whatever
your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/Users/kkhan/MSC/WebTechnology/CGI">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

3c. Enable the handling capacity for cgi scripts and python scripts by giving the extension properly.

```
<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings
from
    # filename extension to MIME-type.
    #
    TypesConfig /private/etc/apache2/mime.types

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
    #
    # If the AddEncoding directives above are commented-out, then you
    # probably should define those extensions to indicate media types:
    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to
"handlers":
    # actions unrelated to filetype. These can be either built into the
server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    AddHandler cgi-script .cgi .py

    # For type maps (negotiated resources):
    #AddHandler type-map var
    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
```

4. create the folder /Users/kkhan/MSC/WebTechnology/CGI and create the scripts.

```
kamils-air:CGI kkhan$ pwd
/Users/kkhan/MSC/WebTechnology/CGI
kamils-air:CGI kkhan$ ls -lh
total 56
-rwxr-xr-x  1 kkhan  staff    91B Apr 24 20:21 first.cgi
-rw-r--r--  1 kkhan  staff   936B Apr 24 20:54 hello.class
-rw-r--r--  1 kkhan  staff   493B Apr 24 20:54 hello.java
-rwxr-xr-x  1 kkhan  staff   812B Apr 25 12:49 hello.py
-rw-r--r--  1 kkhan  staff   315B Apr 24 20:20 hello.sh
-rwxr-xr-x  1 kkhan  staff   812B Apr 25 13:02 helloworld.py
-rwxr-xr-x  1 kkhan  staff   591B Apr 24 20:22 save_file.py
```

Q2. Implement the hello world program in CGI fashion using python (do it , take snap of it (with illustration of each and every step and send as pdf file)

Program:

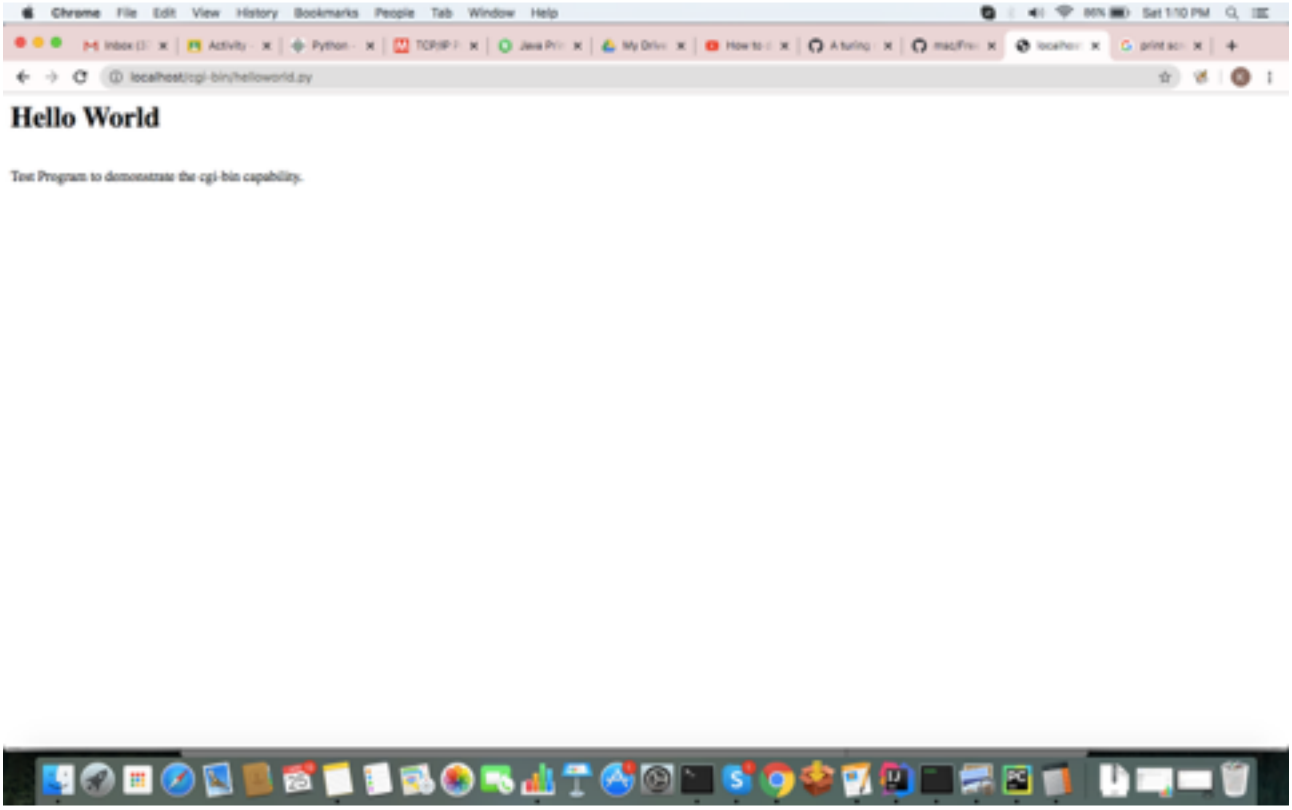
```
#!/usr/bin/python
# Importing the 'cgi' module
import cgi

print("Content-type: html\r\n\r\n")
print("<html><h1> Hello World </h1> <br>Test Program to demonstrate the cgi-bin capability.")
print("<body>")
print("</body></html>")
```

Steps:

1. create the file helloworld.py and edit with any python editor
2. Enter the following line indicating where the python interpreter is installed.
    #!/usr/bin/python - In My machine python is installed in /usr/bin. Different users may need to change this line if python is installed in different   .
3. import cgi module to access environment variables like content length etc.   import cgi
4. First line should indicate the content type. print("Content-type: html\r\n\r\n")

5. print the required html to the console output so that it will be redirected to the browser.

Screen Shot:


Chrome File Edit View History Bookmarks People Tab Window Help
localhost/cgi-bin/helloworld.py

Hello World

Test Program to demonstrate the cgi-bin capability.