Team Members	
Mohamed Ahmed Emary	Fathy Elbadrawy
Andrew Essam	Mohamed Ali El-Ghannam
Amr Ahmed Khallaf	Abdullah Mahany

1 Assignment Questions

1.1 Question 1

> Install Apache HTTP server

Installation on Debian-based systems: `sudo apt install apache2`

Installation on Arch Linux: `sudo pacman -S apache --needed`

```
~ ❱ sudo pacman -S apache --needed
resolving dependencies...
looking for conflicting packages...

Package (3)     New Version  Net Change

extra/apr       1.7.5-3        1.16 MiB
extra/apr-util  1.6.3-2        0.67 MiB
extra/apache    2.4.62-1       6.39 MiB

Total Installed Size:  8.21 MiB

:: Proceed with installation? [Y/n]
(3/3) checking keys in keyring                     [----------------------------------------] 100%
(3/3) checking package integrity                   [----------------------------------------] 100%
(3/3) loading package files                        [----------------------------------------] 100%
(3/3) checking for file conflicts                  [----------------------------------------] 100%
(3/3) checking available disk space                [----------------------------------------] 100%
:: Processing package changes...
(1/3) installing apr                               [----------------------------------------] 100%
(2/3) installing apr-util                          [----------------------------------------] 100%
Optional dependencies for apr-util
    gdbm: enable gdbm support [installed]
    libldap: enable ldap support [installed]
    unixodbc: enable odbc support [installed]
    mariadb-libs: enable mysql/mariadb support
    postgresql-libs: enable postgres support [installed]
    db: enable berkley db support
    sqlite: enable sqlite support [installed]
    nss: enable nss crypto support [installed]
    openssl: enable openssl crypto support [installed]
(3/3) installing apache                            [----------------------------------------] 100%
```

Figure 1: Arch Linux `apache` Installation

Then we can start Apache server service via `systemd`: `sudo systemctl start httpd.service`

1.2 Question 2

> Create two simple html pages named `page1.html`, `page2.html` then use the suitable directive to automatically redirect from `localhost/page1.html` to `localhost/page2.html`.

Steps:

1. Create two HTML pages `page1.html` and `page2.html` in `/srv/http/session-1/q2/` directory.

```
~ ❱ z q2
/srv/http/session-1/q2
q2 ❱ l
Permissions Size User  Date Modified Name
drwxr-xr-x     - emary 22 Jan 06:14  .
drwxr-xr-x     - emary 22 Jan 07:39  ..
.rw-r--r--    76 emary 22 Jan 06:14  .htaccess
.rw-r--r--   126 emary 21 Jan 22:24  page1.html
.rw-r--r--   121 emary 21 Jan 22:12  page2.html
```

Figure 2: `/srv/http/session-1/q2/` directory content

2. Then add the code below to `page2.html`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page 2</title>
  </head>
  <body>
    <h1>Welcome to Page 2</h1>
  </body>
</html>
```

3. Edit apache global config in `/srv/http/session-1/q2/` Directory block, change `AllowOverride None` to `AllowOverride All`:

```
<Directory "/srv/http/session-1/q2">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

4. Create a `.htaccess` file in `/srv/http/session-1/q2/` and add:

```
Redirect /session-1/q2/page1.html /session-1/q2/page2.html
```

5. Restart the apache server with `sudo systemctl restart httpd`.
6. Now when we visit `localhost/session-1/q2/page1.html` we get redirected to `localhost/session-1/q2/page2.html`, and the browser shows the content of `page2.html`.

Welcome to Page 2

Figure 3: `page1.html` redirects to `page2.html`

1.3 Question 3

> Ask for username and password when accessing a directory

Steps:

1. Inside `/srv/http/session-1/q3/` directory, we run the command below to create a `.htpasswd` file:

```
htpasswd -c ./.htpasswd team4
```

```
q3 ❱ htpasswd -c ./.htpasswd team4
New password:
Re-type new password:
Adding password for user team4
q3 ❱ cat .htpasswd
```

	File: .htpasswd
1	team4:$apr1$u0zlhUM6$bxe20bsP2XvcA9RrCKefi.

Figure 4: `htpasswd` command

2. Then inside apache global config `/etc/httpd/conf/httpd.conf`, we add the following block:

```
<Directory "/srv/http/session-1/q3">
    AuthType Basic
    AuthName "Enter your username and password"
    AuthUserFile /srv/http/session-1/q3/.htpasswd
    Require valid-user
</Directory>
```

This block enables basic authentication for the directory `/srv/http/session-1/q3/` and uses the `.htpasswd` file for authentication.

3. Restart the apache server with `sudo systemctl restart httpd`.
4. Now when we visit `localhost/session-1/q3/` we get a prompt to enter a username and password.

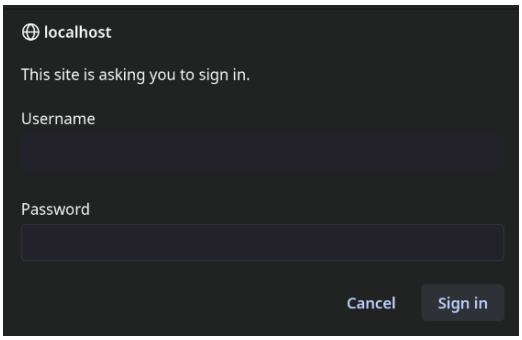

Figure 5: Authentication prompt

1.4 Question 4

> Apply authentication before downloading PDF files

Steps:

1. Create a directory named `q4` in `/srv/http/session-1/` and add a pdf file to it.

2. Inside apache global config `/etc/httpd/conf/httpd.conf`, add the following block:

```
<Directory "/srv/http/session-1/q4">
    AllowOverride All
</Directory>
```

3. Inside `/srv/http/session-1/q3/` directory, we run the command below to create a `.htpasswd` file:

```
htpasswd -c ./.htpasswd team4
```

4. In the same directory, create a `.htaccess` file and add the following content:

```
AuthType Basic
AuthName "Enter your username and password to access pdf"
AuthUserFile /srv/http/session-1/q4/.htpasswd
Require valid-user
```

```
q4 ❱ htpasswd -c ./.htpasswd team4
New password:
Re-type new password:
Adding password for user team4
q4 ❱
q4 ❱ cat .htaccess .htpasswd
       File: .htaccess
   1   AuthType Basic
   2   AuthName "Enter your username and password to access pdf"
   3   AuthUserFile /srv/http/session-1/q4/.htpasswd
   4   Require valid-user

       File: .htpasswd
   1   team4:$apr1$YhdvQnbZ$DPHwXZOqyIQlO4AZZx/mC0
```

Figure 6: `.htaccess` and `.htpasswd`

5. Restart the apache server with `sudo systemctl restart httpd`.
6. Now when we visit `localhost/session-1/q4/` we get a prompt to enter a username and password.

1.5 Question 5

> Create a directory then allow access to one of your classmates only

Suppose my classmate's IP address is `165.32.65.74`

Steps:

1. Create two files `test1.html` and `test2.html` in `/srv/http/session-1/q5/` directory.
2. Inside apache global config `/etc/httpd/conf/httpd.conf`, add the following block:

```
<directory "/srv/http/session-1/q5/">
    deny from all
    allow from 165.32.65.74
```

```
    order deny,allow
</directory>
```

3. Restart the apache server with `sudo systemctl restart httpd`.
4. Now when we visit `localhost/session-1/q5/` we get a `403 Forbidden` error.

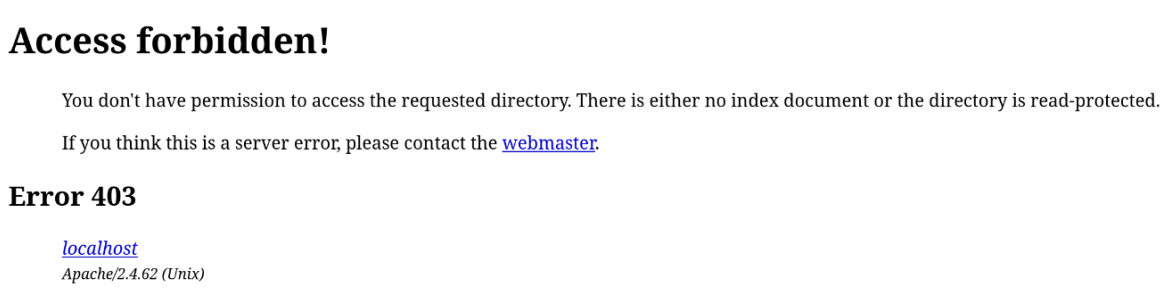

Figure 7: Access Forbidden Error

1.6 Question 6

> Disable listing the directory content (hint use `Indexes`)

Apache serves the path `/srv/http/` by default on Arch Linux, so I created a directory for session 1 assignment with the name `session-1` and added a sub directory `q6` in it for question 6.

In apache global config `/etc/httpd/conf/httpd.conf`, I created a `<Directory>` block with this content:

```
<Directory "/srv/http/session-1/q6">
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

After adding that block, I restarted the apache server with `sudo systemctl restart httpd` and now when I visit `http://localhost/session-1/q6` I get a `403 Forbidden` error.

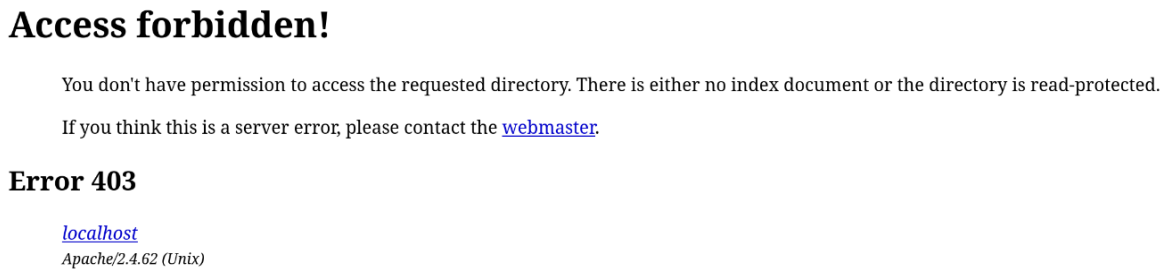

Figure 8: Access Forbidden Error