

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

Change the default index page to `default.html` instead of `index.html` (use `DirectoryIndex`)

Steps:

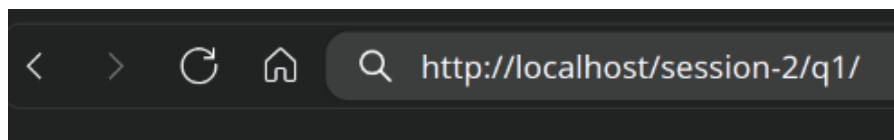
1. In apache config file, change the line `DirectoryIndex index.html` to `DirectoryIndex default.html`

```
1 <IfModule dir_module>
2     DirectoryIndex default.html
3 </IfModule>
```

2. Create a directory `/srv/http/session-2/q1/` and add a file `default.html` with the content below:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Default Page</title>
5 </head>
6 <body>
7     <h1>This is the default page.</h1>
8 </body>
9 </html>
```

2. Restart the apache server using `sudo systemctl restart apache2`
3. Open `localhost/session-2/q1/` in the browser and you should see the content of `default.html`



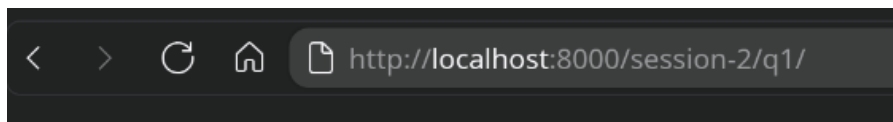
# This is the default page.

Figure 1: Default Page

### 1.2 Question 2

Change the default port to be 8000 instead of 80

1. Inside the apache config file, change the line `Listen 80` to `Listen 8000`
2. Restart the apache server using `sudo systemctl restart apache2`
3. Open `localhost:8000` in the browser and you should see the default apache page
4. Default page from [question 1](#) should be accessible at `localhost:8000/session-2/q1/`



**This is the default page.**

Figure 2: Default Page on Port 8000

### 1.3 Question 3

What is the importance of rewrite module?

Importance of the Rewrite Module:

#### 1. SEO-Friendly URLs:

- Converts complex, dynamic URLs (e.g., `example.com/index.php?id=123`) into clean, user-friendly URLs (e.g., `example.com/products/123`).
- Improves website readability and enhances search engine optimization (SEO).

#### 2. Redirection Rules:

- Redirects outdated or moved pages to their new locations without breaking links (e.g., redirecting `old-page.html` to `new-page.html`).
- Helps maintain traffic and rankings when migrating or restructuring a website.

#### 3. Custom Error Pages:

- Directs users to friendly error pages instead of default Apache error pages (e.g., `example.com/error404.html` for a 404 Not Found error).

#### 4. Access Control:

- Restricts or grants access to specific URLs based on conditions like IP addresses, user agents, or HTTP headers.

#### 5. Dynamic to Static URL Conversion:

- Rewrites dynamic URLs to appear static, which can improve speed and indexing in search engines.

#### 6. Proxying and Load Balancing:

## Bonus Question

- Used in combination with other modules to forward requests to other servers or backends, acting as a reverse proxy.

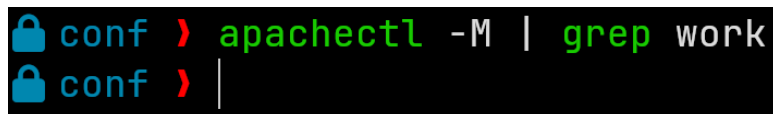
### 7. Localization and Multilingual Sites:

- Routes requests to language-specific resources based on the user's preferences or location.

## 1.4 Question 4

Configure apache to use Worker MPM

1. Check if the `worker` MPM is already enabled using `apachectl -M | grep worker`



```
conf > apachectl -M | grep work
worker_module (shared)
```

Figure 3: Find `worker` MPM Module in Enabled Modules

2. If you get nothing like in the image above, uncomment the line `LoadModule mpm_worker_module modules/mod_mpm_worker.so` in the apache config file, then disable the current MPM module.



```
conf > rg worker
httpd.conf
68:#LoadModule mpm_worker_module modules/mod_mpm_worker.so
```

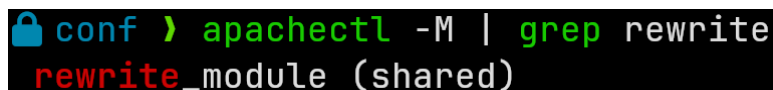
Figure 4: `mpm_worker_module` in apache config

3. Restart the apache server using `sudo systemctl restart apache2`

## 1.5 Question 5

Check the status of `rewrite_mod` and enable it if it's enabled.

1. Using the command `apachectl -M | grep rewrite`, we found that the `rewrite` module is enabled.
  - If it wasn't enabled, we would have to enable it by uncommenting the line `LoadModule rewrite_module modules/mod_rewrite.so` in the apache config file.



```
conf > apachectl -M | grep rewrite
rewrite_module (shared)
```

Figure 5: Find if `rewrite_module` is Enabled

## 1.6 Bonus Question

Write `RewriteCond` and `RewriteRules` for the following:

Before the following rules, we need to enable the `rewrite` engine by adding the line below:

```
1 | RewriteEngine On
```

## Bonus Question

---

Deny access to localhost/page?queryString if queryString contains the string forbidden.

```
1 RewriteCond %{QUERY_STRING} forbidden [NC]
2 RewriteRule ^ - [F]
```

Remove the Query String

```
1 RewriteCond "%{QUERY_STRING}" "var=(val)"
2 RewriteRule ^/page1 /page2? [R]
```

Rewrite URLs like localhost/page1?var=val to localhost/page2?var=val but do not rewrite if val isn't present.

```
1 RewriteCond %{QUERY_STRING} var=.+
2 RewriteRule ^/page1$ /page2?%{QUERY_STRING} [R]
```

Take a URL of the form localhost/path?var=val and transform it into localhost/path/var/val.

```
1 RewriteCond %{QUERY_STRING} (.)=(.*)
2 RewriteRule ^(.+) $1/%1/%2? [R]
```

Map localhost/example/one/two to localhost/something.cgi?arg=one&other=two

```
1 RewriteRule ^/example/(.+)/(.+)$ /something.cgi?arg=$1&other=$2 [R]
```