

Graded Assignment on Networking and Servers

Course 3: Networking and Servers | DevOps B4

Question 1:

Deploy a website on localhost using either apache2 or Nginx. Create a DNS name for this website as 'awesomeweb'. You can use any web template you want or can write your own simple HTML code. Write detailed documentation with the steps involved.

Prerequisites:

- A computer with a Linux operating system (e.g., Ubuntu).
- Basic understanding of terminal commands.

Steps:

1. Open your terminal and update package lists. If packages can be upgraded, then upgrade it.

sudo apt update

Sudo apt upgrade -y

```
kinnar@KINNAR-PC:~$ sudo apt update
[info] password for kinnar:
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.3 kB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [29.6 kB]
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [119 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease [119 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1206 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [189 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1362 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [225 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1831 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-updates/main Translation-en [250 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1859 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [250 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [850 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [209 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1097 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [239 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [62.1 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.4 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]
Fetched 9745 kB in 9s (1116 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
18 packages can be upgraded. Run 'apt list --upgradable' to see them.
kinnar@KINNAR-PC:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  ubuntu-pro-client
The following packages will be upgraded:
  alsa-ucm-conf apache2 apache2-bin apache2-data apache2-utils binutils binutils-common binutils-x86-64-linux-gnu docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  dpkg less libbrotli1 libctf-nobfd0 libctf9 libcups2 libexpat1 libfdup-2.5-0 libfdup-common libss1 libtiff5 libxml2 libxslt2 openjdk-17-jdk openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless
  openssl python3-cryptography python3-update-manager tcpdump tntdata ubuntu-advantage-tools ubuntu-pro-client-libs update-manager-core
38 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
19 standard IT security updates
Need to get 223 MB of archives.
After this operation, 818 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.11.0-1-ubuntu.22.04-jammy [29.3 MB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 dpkg amd64 1.21.1ubuntu2.3 [1239 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libfdup-2.5-0 amd64 2.5.174dfg-ubuntu8.22.04.1 [183 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libss1 amd64 3.0-2-ubuntu1.15 [1906 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:25.0-4-1-ubuntu.22.04-jammy [13.7 MB]
```

2. Install Nginx:

```
sudo apt install nginx
```

```
kinnar@DESKTOP-TK100QA:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

3. Create a Test Website:

Create a directory for your website files (e.g., /var/www/html/awesomeweb).

We'll use a simple HTML file for demonstration.

```
sudo mkdir -p /var/www/html/awesomeweb
```

4. Create an index.html file inside the directory with your preferred content.

HTML Code Block:

```
<!DOCTYPE html>
<html>
<head>
  <title>Awesome Website</title>
</head>
<body>
  <h1>Welcome to my awesome website!</h1>
</body>
</html>
```

Note: You can replace this code with any HTML content you want for your website.

5. Configure Nginx Server Block:

Create a new server block configuration file:

```
sudo nano /etc/nginx/sites-available/awesomeweb.conf
```

Nginx Server Block

```
server {
    listen 80;
    server_name localhost.awesomeweb.com;
    location / {
        root /var/www/html/awesomeweb;
        index index.html index.htm;
    }
    error_log /var/log/nginx/awesomeweb_error.log;
    access_log /var/log/nginx/awesomeweb_access.log;
}
```

Save and close the file.

6. Enable the Server Block:

Create a symbolic link from the available configuration file to the enabled sites directory:

```
sudo ln -s /etc/nginx/sites-available/awesomeweb.conf /etc/nginx/sites-enabled/
```

7. Test Nginx Configuration:

Check the Nginx configuration for syntax errors:

```
sudo nginx -t
```

If there are no errors, proceed to the next step. Otherwise, fix the errors in your configuration file and re-run the test command.

8. Restart Nginx:

```
sudo systemctl restart nginx
```

9. Configure Local DNS (hosts file):

Open the hosts file for editing:

```
sudo nano /etc/hosts
```

Add a new line at the bottom with the following format:

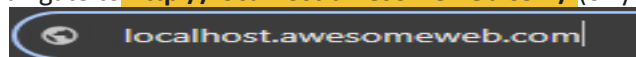
```
127.0.0.1 localhost.awesomeweb.com
```

Note: Replace awesomeweb with your desired DNS name for the website.

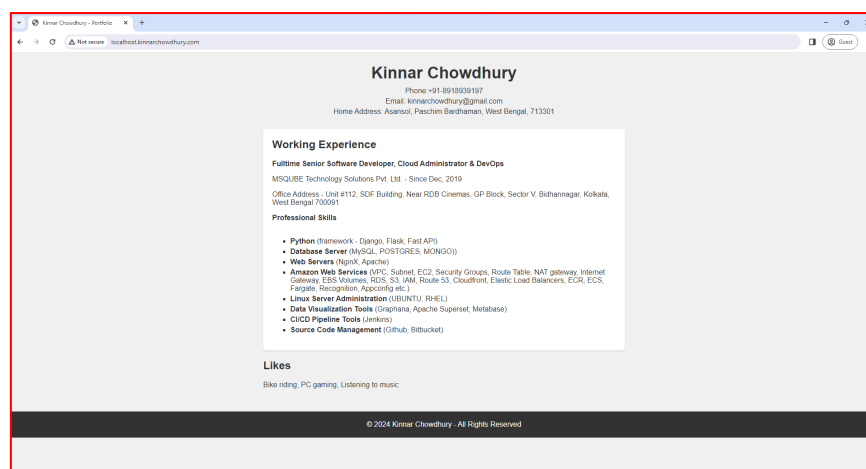
Save and close the file.

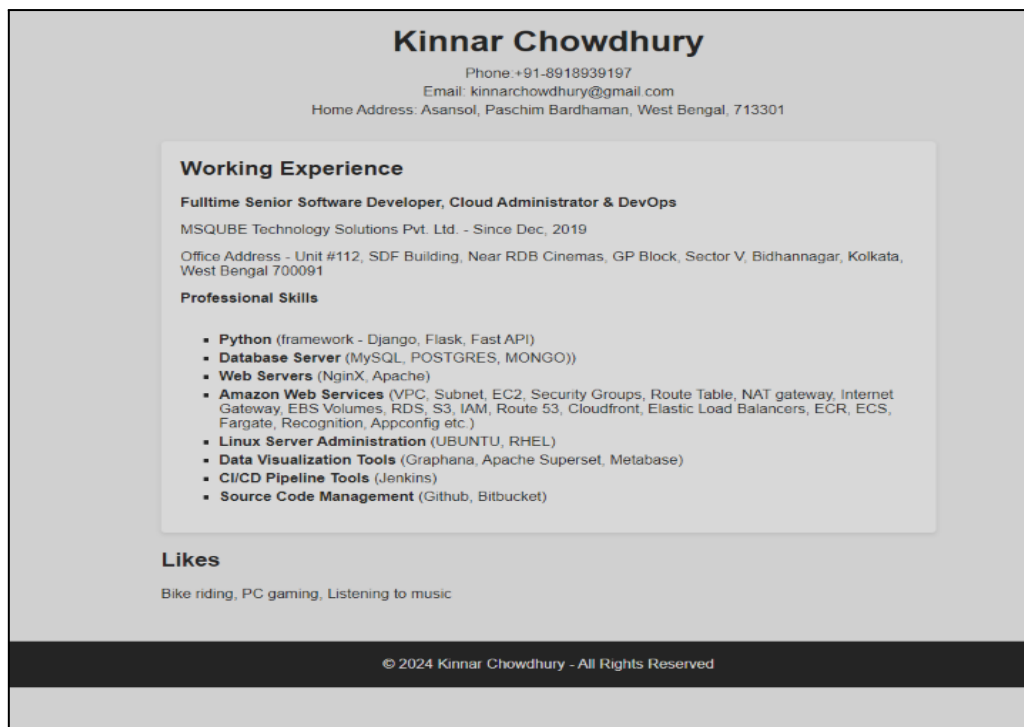
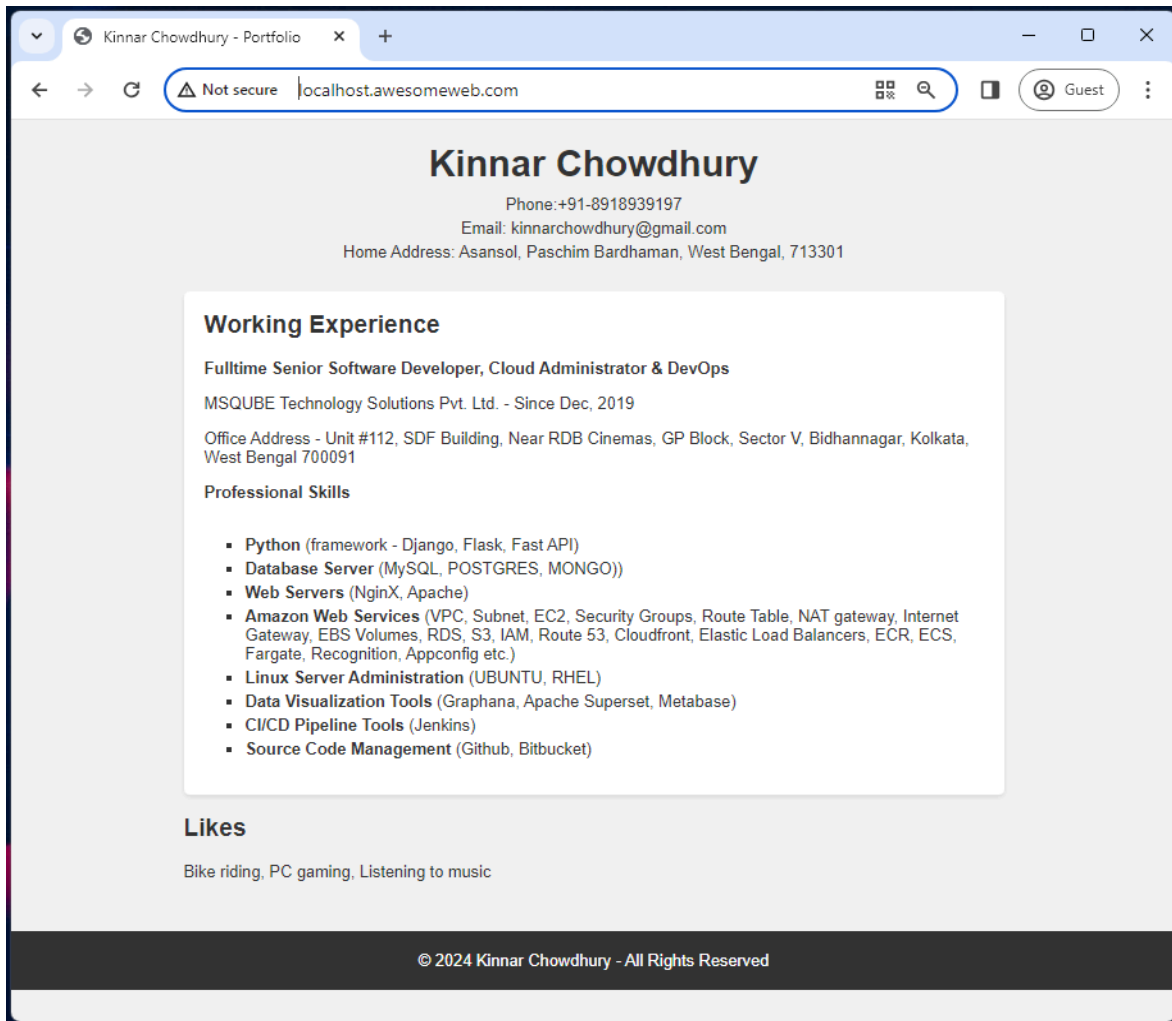
10. Testing Your Website:

Open a web browser and navigate to <http://localhost.awesomeweb.com/> (or your chosen DNS name) in the address bar.



You should see your website content below if everything is set up correctly.





THANK YOU

===== End of File =====