Lab Report: Apache Web Server Installation & Maintenance

Name: Amit Kumar Sharma
Registration: 2020831009
Platform: Ubuntu on WSL (Windows Subsystem for Linux)

Objectives

1. Install, administer, and maintain an Apache web server.
2. Configure virtual hosts to host multiple websites on a single server.
3. Host dynamic websites using HTML and JavaScript.

Environment & Setup

- **OS:** Ubuntu 22.04 on Windows 10 via WSL
- **Web Server:** Apache2
- **Browser:** Firefox / Chrome (for testing)
- **Terminal Tools:** `curl` to demonstrate web server responses

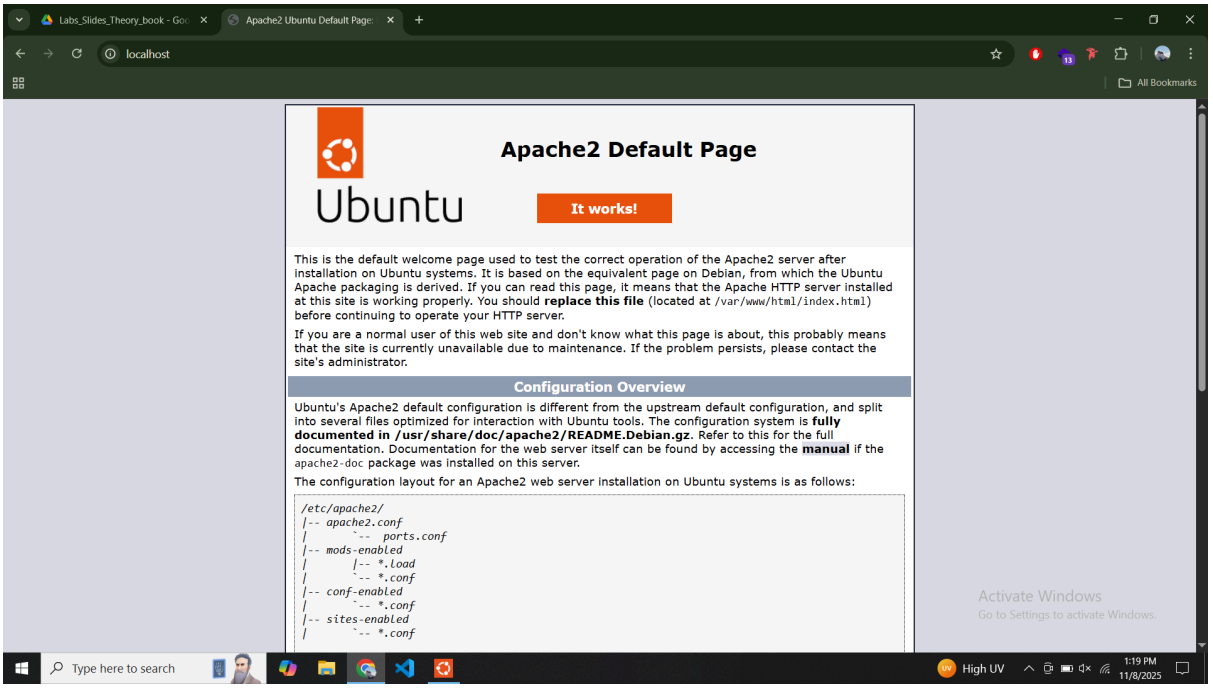
---

Task 1: Installing Apache Web Server

Step 1: Installing Apache

```
sudo apt update
sudo apt install apache2
```

Check



Apache status:

```
sudo systemctl status apache2
```

Observation: Apache service started successfully.

Step 2: Adjusting the Firewall

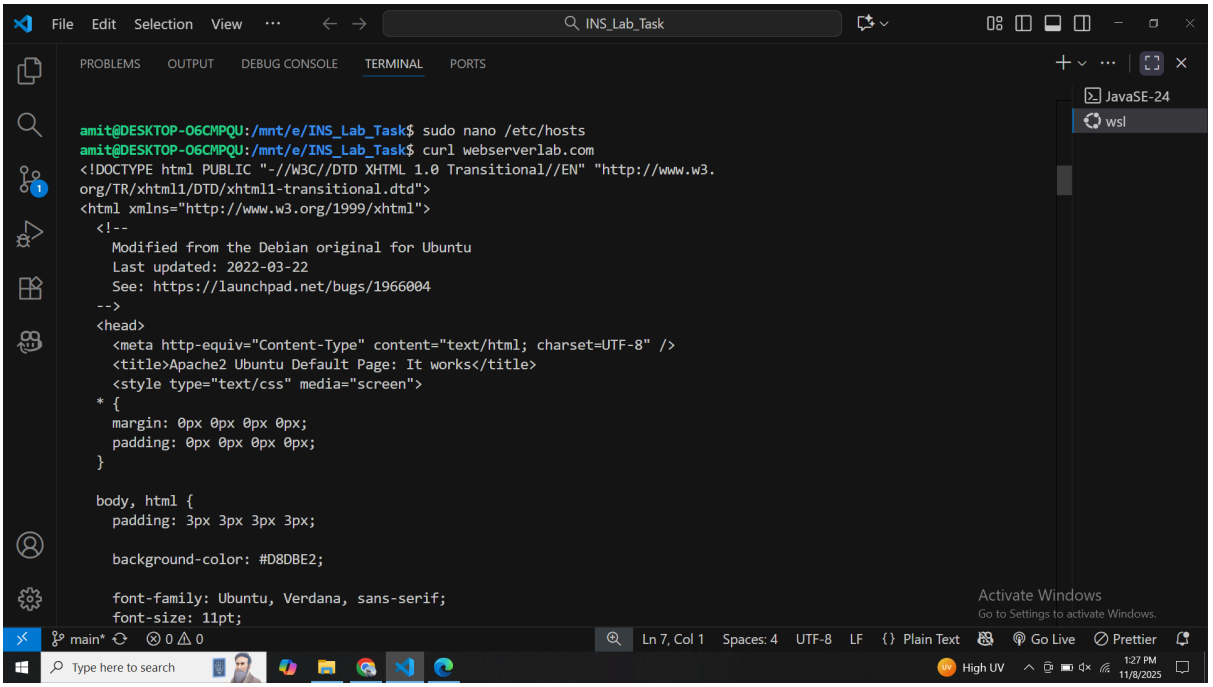
```
sudo ufw allow 'Apache'
sudo ufw status
```

Observation: HTTP traffic allowed.

Step 3: Checking the Web Server

Test via terminal:

```
curl http://webserverlab.com
```



Observation: Default Apache page displayed.

✅ **Checkpoint 1:** Apache installation verified.

Task 2: Setting Up Virtual Hosts

Step 1: Managing the Apache Process

```
sudo systemctl stop apache2
sudo systemctl start apache2
sudo systemctl restart apache2
sudo systemctl reload apache2
sudo systemctl enable apache2
sudo systemctl disable apache2
```

Observation: Commands to control Apache service demonstrated.

Step 2: Single Virtual Host – `example.com`

Create directory and assign permissions:

```
sudo mkdir -p /var/www/example.com/html
```

```
sudo chown -R $USER:$USER /var/www/example.com/html
sudo chmod -R 755 /var/www/example.com
```

Create `index.html`:

```
<html>
<head><title>Welcome to Example.com!</title></head>
<body><h1>Success! The example.com server block is
working!</h1></body>
</html>
```

Create virtual host configuration:

```
sudo nano /etc/apache2/sites-available/example.com.conf
```

Add:

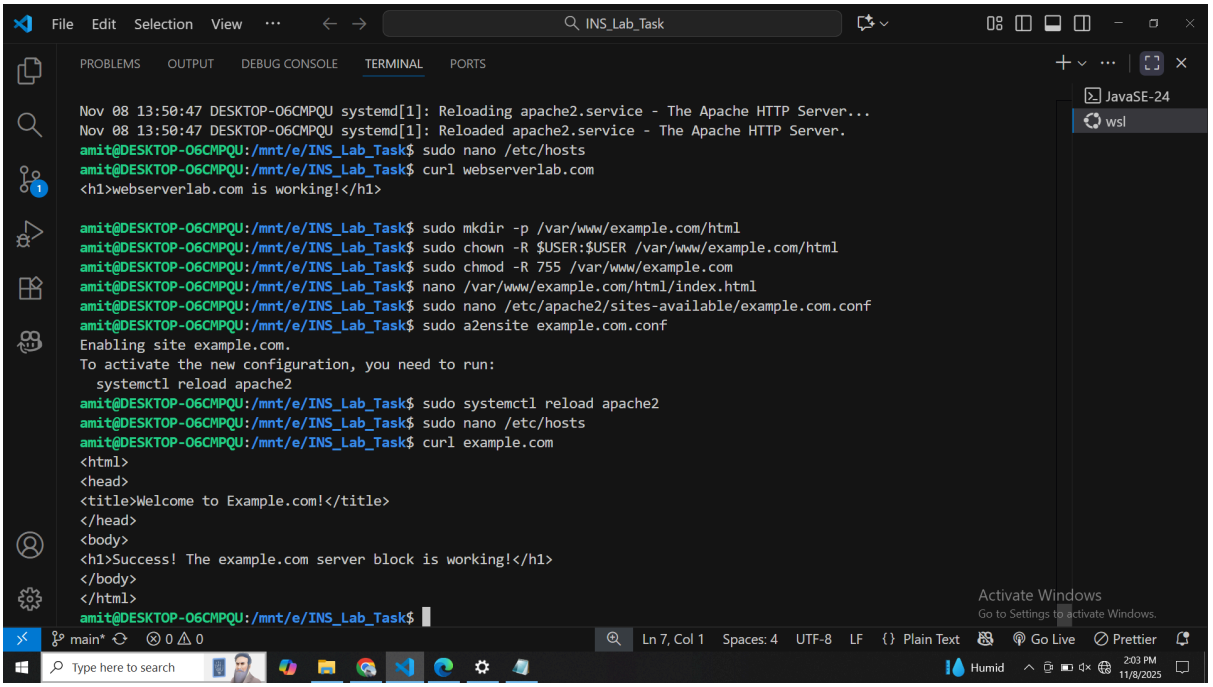
```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Enable site and disable default:

```
sudo a2ensite example.com.conf
sudo a2dissite 000-default.conf
sudo apache2ctl configtest
sudo systemctl restart apache2
```

Test:

```
curl http://example.com
```



✅ **Checkpoint 2:** Virtual host `example.com` is working.

Step 3: Observing Behavior with Multiple Hosts

After enabling `example.com` and restarting Apache:

```
sudo a2ensite example.com.conf
sudo systemctl restart apache2
```

Test:

```
curl http://webserverlab.com
curl http://127.0.0.1
```

Observation:

- `webserverlab.com` now points to `example.com` content because the default site was disabled.
- Requests to `127.0.0.1` also serve `example.com` content.

Reason: Apache serves the first matching virtual host. Without the default site, requests without a specific `ServerName` match the first enabled virtual host (`example.com`).

```
</html>
amit@DESKTOP-O6CMPQU:/mnt/e/INS_Lab_Task$ curl localhost
<html>
<head>
<title>Welcome to Example.com!</title>
</head>
<body>
<h1>Success! The example.com server block is working!!</h1>
</body>
</html>
```

```
amit@DESKTOP-O6CMPQU:/mnt/e/INS_Lab_Task$ curl webserverlab.com
<html>
<head>
<title>Welcome to Example.com!</title>
</head>
<body>
<h1>Success! The example.com server block is working!!</h1>
</body>
</html>
amit@DESKTOP-O6CMPQU:/mnt/e/INS_Lab_Task$
```

wsl
JavaSE-24
wsl

Whole Image 1076x254 27.07KB Go Live

12:48 AM
11/9/2025

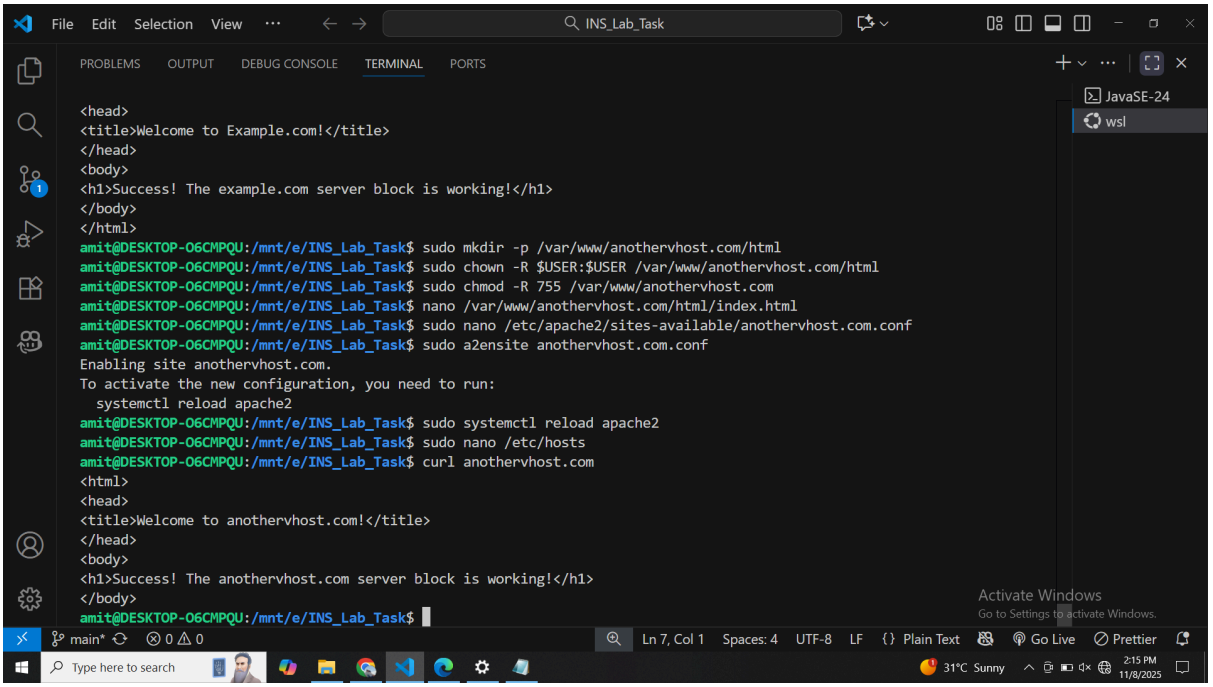
✅ **Checkpoint 3:** Virtual host behavior demonstrated and explained.

Step 4: Multiple Virtual Hosts – `anothervhost.com`

Repeat steps to create another virtual host with a different HTML file:

Test:

```
curl http://anothervhost.com
```



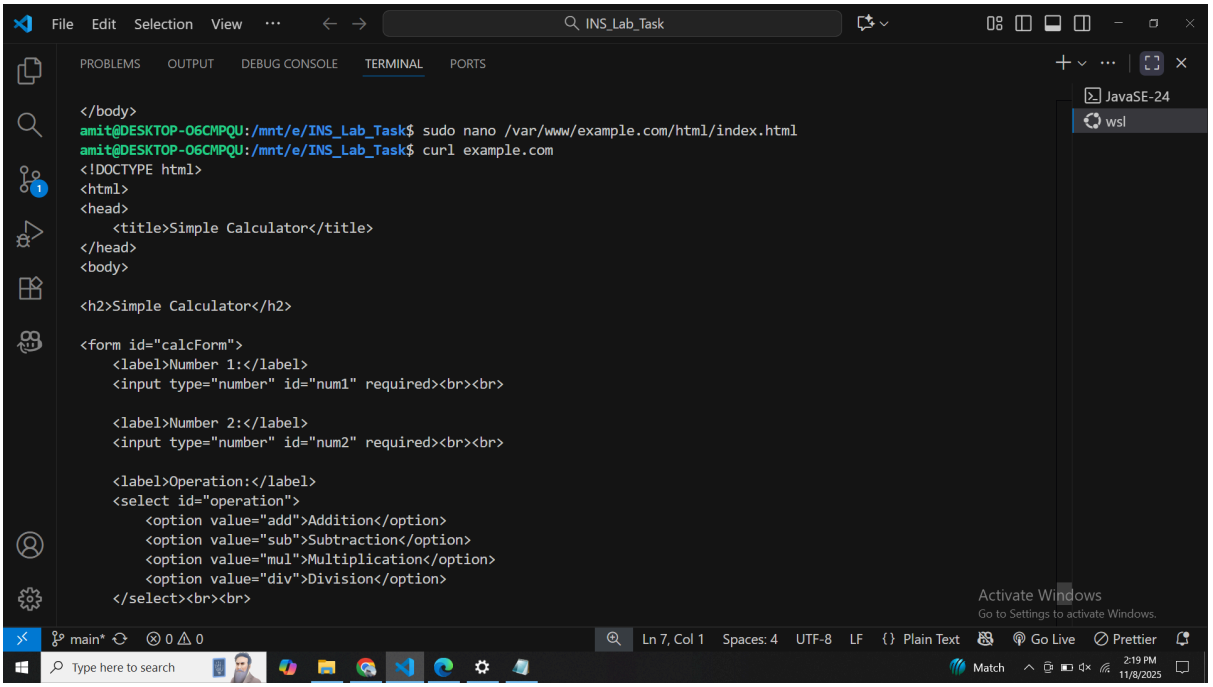
✅ **Checkpoint 4:** Multiple virtual hosts successfully deployed.

Task 3: Hosting Dynamic Websites Using HTML & JavaScript

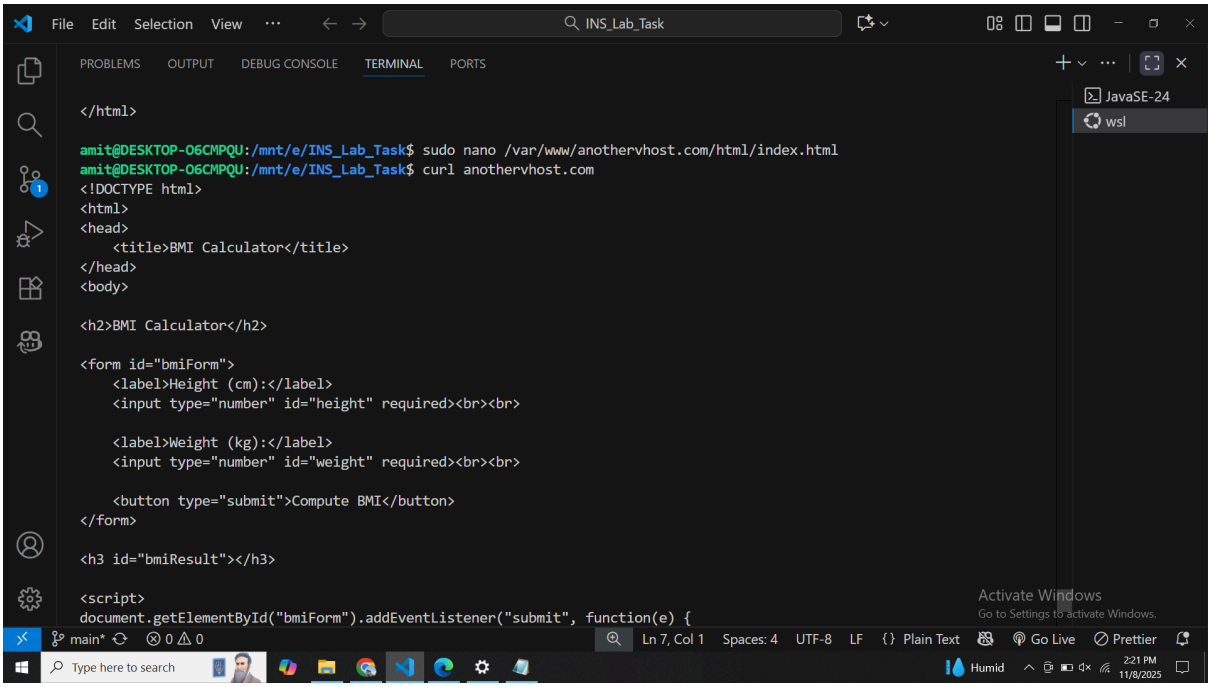
- Two dynamic websites were created using HTML forms and JavaScript.
- Example: `example.com` collects user input and displays it dynamically using JS.

Test via terminal:

`curl` http://example.com/form.html



curl http://anothervhost.com/form.html



✅ **Checkpoint 5:** Two dynamic websites deployed successfully.

Conclusion

1. Apache was successfully installed and verified.

2. Virtual hosts were configured, including multiple virtual hosts.
3. Dynamic websites using HTML & JavaScript were hosted.
4. Apache service management commands were demonstrated.

All 5 checkpoints completed successfully. ✅