# CS218

# MOBILE COMPUTING

## ASSIGNMENT 1
## DOCUMENTATION


Group Members:
Shanesh Dewan (S11219250)
Vishant Chand (S11219214)

# Contents

# Student Contribution

| Student Name | Student ID | Percentage Contribution | Student Signature |
|---|---|---|---|
| Vishant Chand | S11219214 | 100% | VChand |
| Shanesh Dewan | S11219250 | 100% | SDewan |

# Part 1: Virtual Network & Web Server Configuration

## Introduction

In this assignment we will be designing and configuring a mobile network using packet tracer to simulate connectivity from a home network to a foreign network with roaming. The key requirement of this section is to  configure DHCP services which will allow IP addresses to be assigned to devices dynamically (Fortinet, n.d.) , ping tests must be done to ensure there is connectivity between devices, and a working HTML web page configured on the  web server which will allow students to login using their details and see fun facts.

## Network topology design

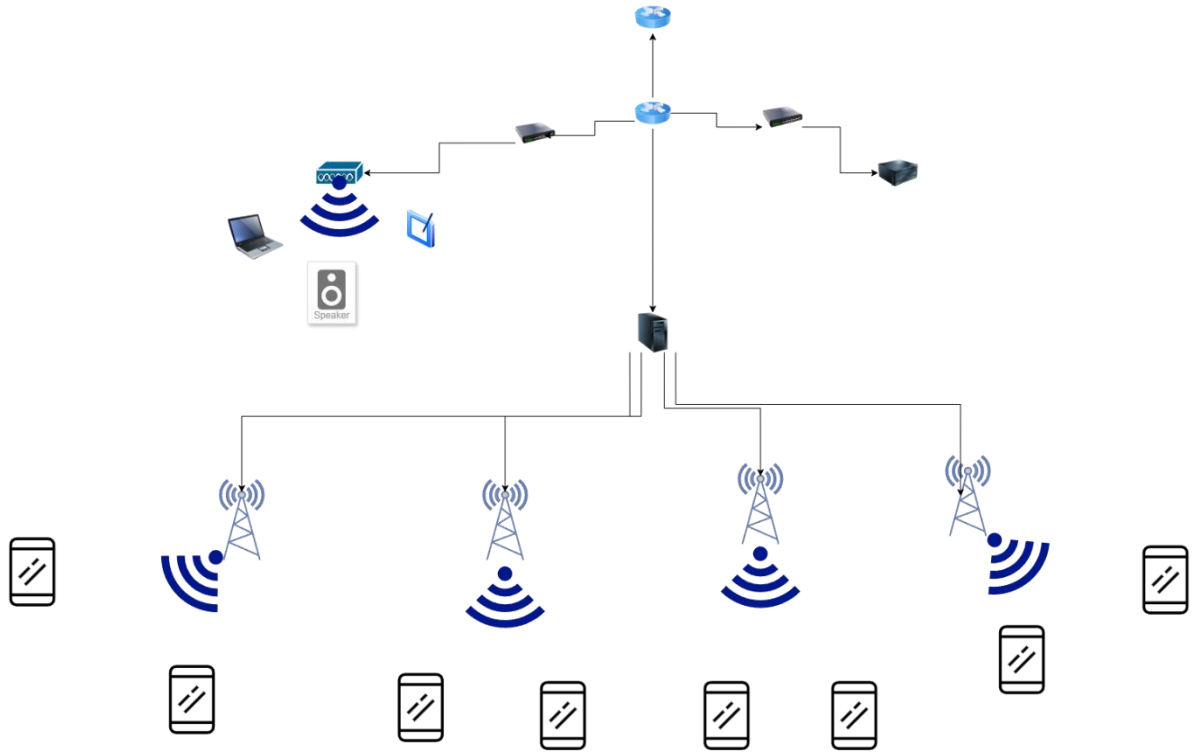



*Figure 1: The diagram above represents the mobile network topology and was designed using 'app.diagrams.net*

## Physical topology

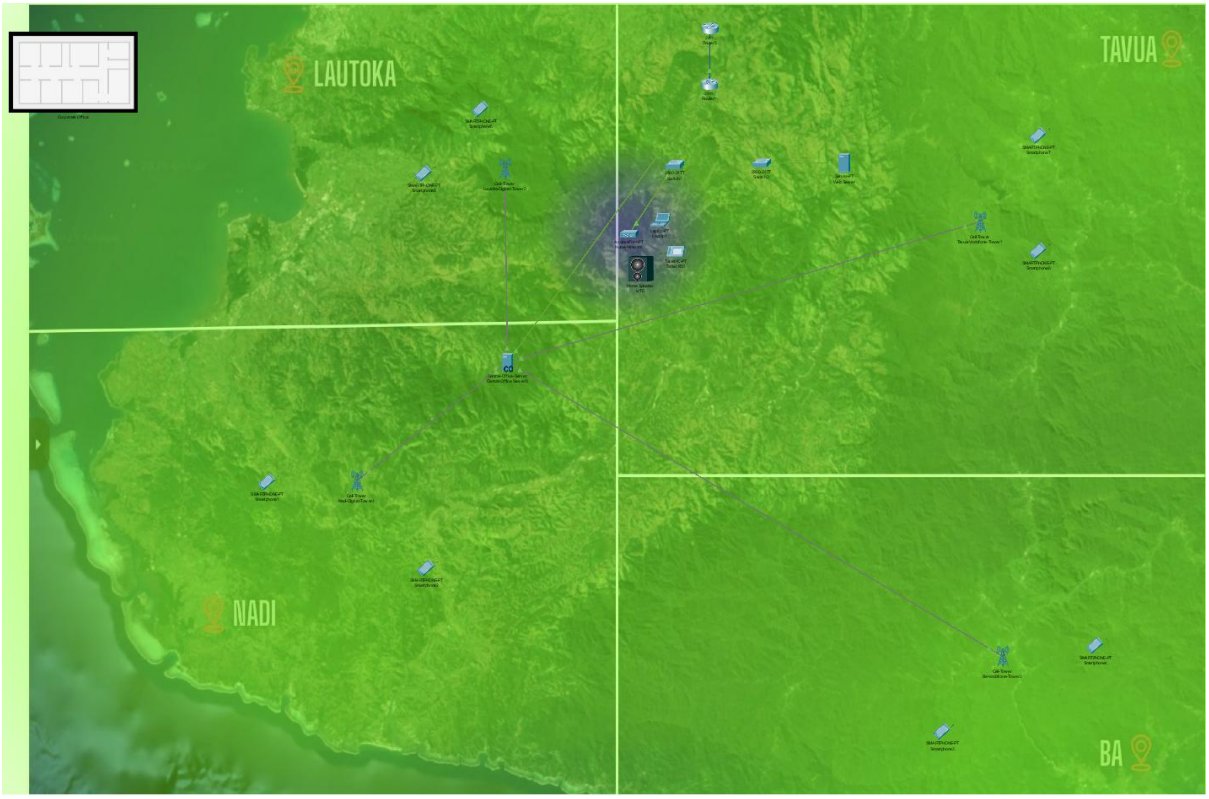



*Figure 2: The physical topology represents four major cities (Nadi, Lautoka, Tavua and Ba)*

## Description

Devices used in the topology:

- 1 x Cloud (Internet)
- 1 x Router (2911)
- 2 x Switch (2960)
- 1 x MEC Server (Central Office server)
- 1 x Web server (hosts index.html file for student login)
- 1 x Wireless Access Point (connecting IoT and client devices)
- 3 x Client devices:
    - -1 x Laptop
    - -1 x Home Speaker (IoT)
    - -1 x Tablet
- 4 x Cell Towers (Base Stations, connecting cell phones via 3G/4G connectivity)

# Logical Topology

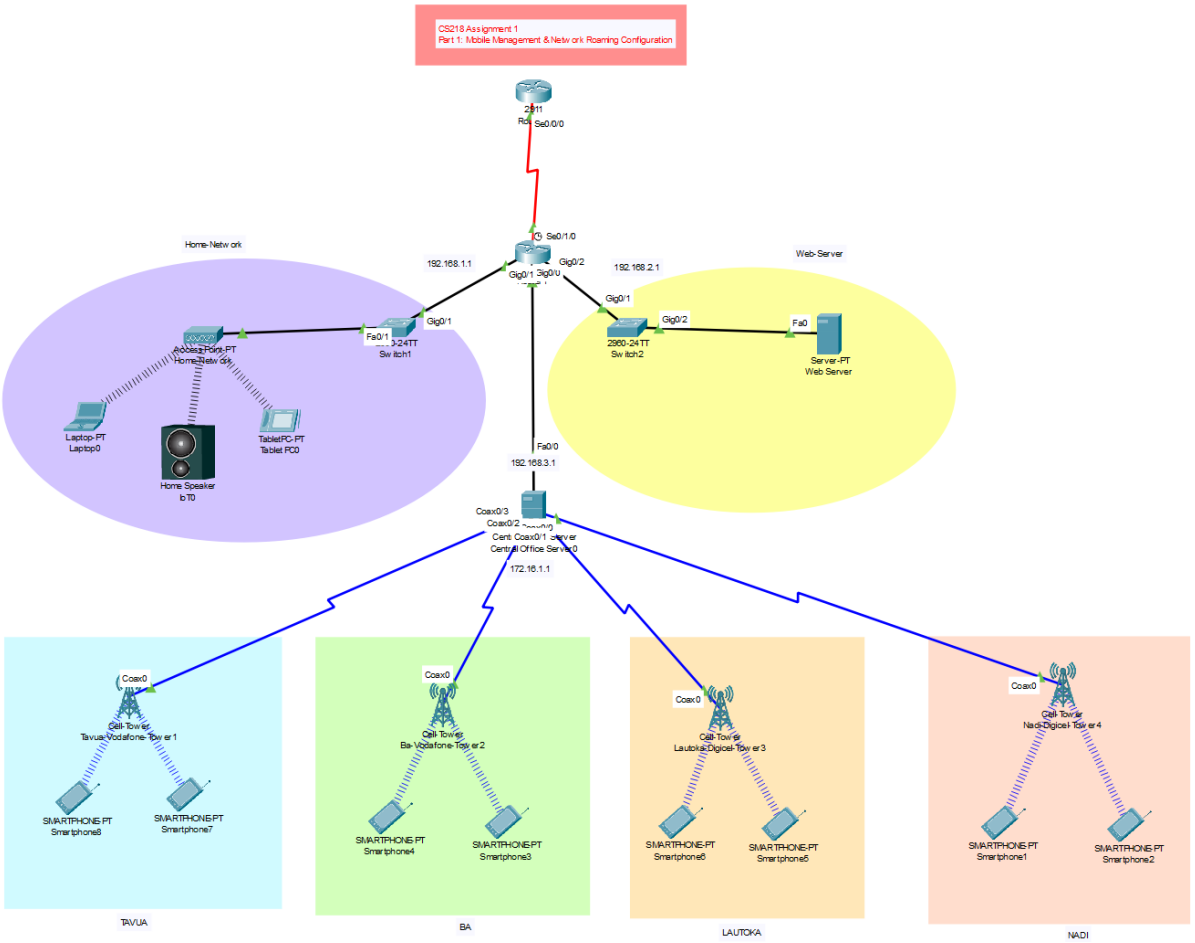



*Figure 3: Final mobile network topology in logical view*

# IP Addressing & Configuration

- Router1(Main-Router) configuration

```
Main-Router#sh running-config | section interface
interface GigabitEthernet0/0
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
interface GigabitEthernet0/2
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
interface FastEthernet0/0/0
 switchport mode access
interface FastEthernet0/0/1
 switchport mode access
interface FastEthernet0/0/2
 switchport mode access
interface FastEthernet0/0/3
 switchport mode access
interface Serial0/1/0
 ip address 200.0.0.2 255.255.255.252
 clock rate 2000000
interface Serial0/1/1
 no ip address
 clock rate 2000000
interface Vlan1
 no ip address
 shutdown
```

*Figure 4: The command used in the screenshot shows configurations on the interfaces of the router*

```
Main-Router#sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.3.1     YES NVRAM  up                    up
GigabitEthernet0/1     192.168.1.1     YES NVRAM  up                    up
GigabitEthernet0/2     192.168.2.1     YES NVRAM  up                    up
FastEthernet0/0/0      unassigned      YES unset  up                    down
FastEthernet0/0/1      unassigned      YES unset  up                    down
FastEthernet0/0/2      unassigned      YES unset  up                    down
FastEthernet0/0/3      unassigned      YES unset  up                    down
Serial0/1/0            200.0.0.2       YES manual up                    up
Serial0/1/1            unassigned      YES unset  down                  down
Vlan1                  unassigned      YES unset  administratively down down
```

*Figure 5: The command 'show ip interface brief' shows all interfaces, Ip addresses and port status*

```
ip route 172.16.1.0 255.255.255.0 192.168.3.2
ip route 0.0.0.0 0.0.0.0 200.0.0.1
```

*Figure 6: Assign static IP addresses to allow connectivity between devices on different interfaces*

```
Main-Router#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 200.0.0.1 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 1 subnets
S       172.16.1.0/24 [1/0] via 192.168.3.2
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.1/32 is directly connected, GigabitEthernet0/1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/2
L       192.168.2.1/32 is directly connected, GigabitEthernet0/2
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/0
L       192.168.3.1/32 is directly connected, GigabitEthernet0/0
     200.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       200.0.0.0/30 is directly connected, Serial0/1/0
L       200.0.0.2/32 is directly connected, Serial0/1/0
S*   0.0.0.0/0 [1/0] via 200.0.0.1
```

*Figure 7: Verify configured routes by using the command 'show ip route'*

- Router2(ISP-Router) configuration

```
ISP-Router#sh running-config | section interface
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
interface GigabitEthernet0/2
 no ip address
 duplex auto
 speed auto
 shutdown
interface Serial0/0/0
 ip address 200.0.0.1 255.255.255.252
interface Serial0/0/1
 no ip address
 clock rate 2000000
 shutdown
interface Vlan1
 no ip address
 shutdown
```

*Figure 8: Configure interfaces connected to the Main-Router*

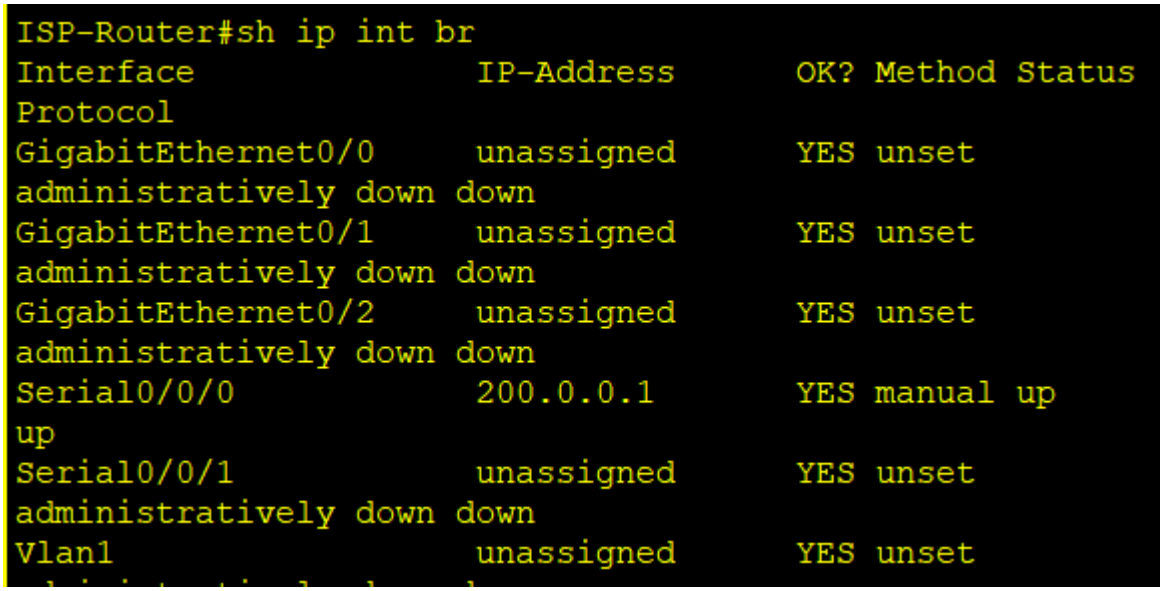


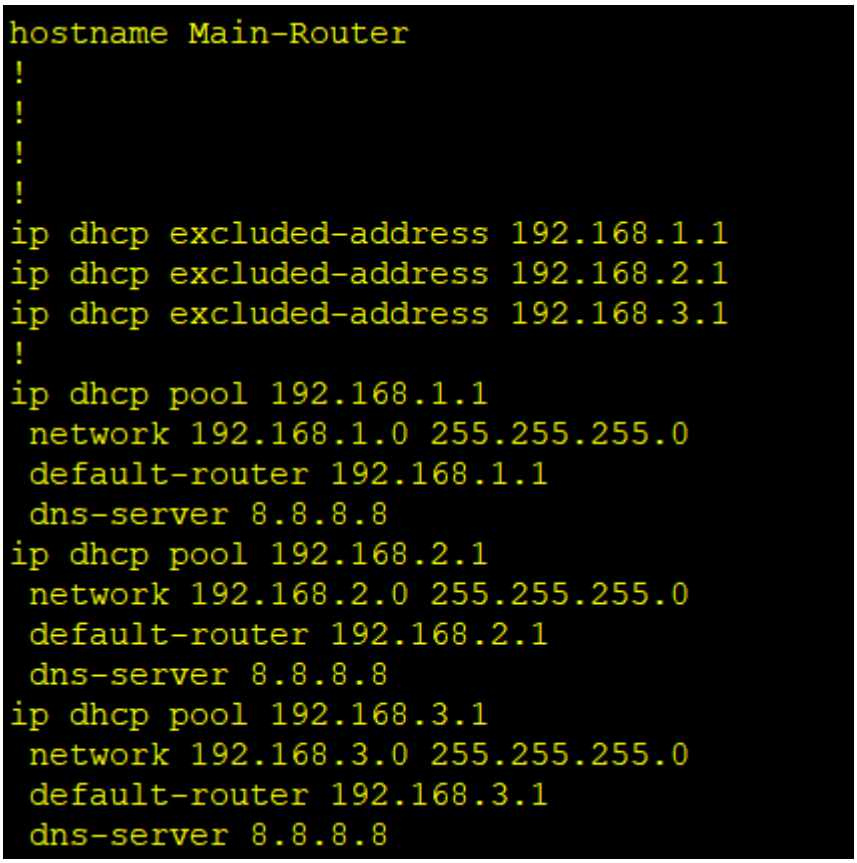

*Figure 9: Verify interface configuration*

- Table of IP assignments

| Device | Subnet Mask | Interface | IP | Gateway |
|---|---|---|---|---|
| Router1(Main-Router) | 255.255.255.0 | S0/0/0 | 10.1.1.1 | - |
| Router2(ISP-Router) | 255.255.255.0 | G0/0 | 192.168.3.1 | - |
| | 255.255.255.0 | G0/1 | 192.168.1.1 | - |
| | 255.255.255.0 | G0/2 | 192.168.2.1 | - |
| | 255.255.255.0 | S0/1/0 | 10.1.1.2 | - |
| Central Office Server(Cellular) | 255.255.255.0 | C0/0 | DHCP | 172.16.1.1 |
| | | C0/1 | DHCP | |
| | | C0/2 | DHCP | |
| | | C0/3 | DHCP | |
| Central Office Server(backbone) | 255.255.255.0 | F0/0 | DHCP | 192.168.3.1 |
| Server(Web server) | 255.255.255.0 | G0/2 | DHCP | 192.168.2.1 |
| Switch 1 | 255.255.255.0 | F0/1 | DHCP | 192.168.1.1 |
| Switch 2 | 255.255.255.0 | - | DHCP | 192.168.2.1 |
| Wireless AP | - | wireless | - | 192.168.1.1 |
| Cell Tower | 255.255.255.0 | wireless | DHCP | 172.16.1.1 |
| Smart Phone | 255.255.255.0 | - | DHCP | 172.16.1.1 |
| Laptop | 255.255.255.0 | - | DHCP | 192.168.1.1 |
| Home Speaker | 255.255.255.0 | - | DHCP | 192.168.1.1 |
| Tablet | 255.255.255.0 | - | DHCP | 192.168.1.1 |

# DHCP Server Setup

- Screenshot of DHCP configuration on the Router1(Main-Router)



```
hostname Main-Router
!
!
!
!
ip dhcp excluded-address 192.168.1.1
ip dhcp excluded-address 192.168.2.1
ip dhcp excluded-address 192.168.3.1
!
ip dhcp pool 192.168.1.1
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
 dns-server 8.8.8.8
ip dhcp pool 192.168.2.1
 network 192.168.2.0 255.255.255.0
 default-router 192.168.2.1
 dns-server 8.8.8.8
ip dhcp pool 192.168.3.1
 network 192.168.3.0 255.255.255.0
 default-router 192.168.3.1
 dns-server 8.8.8.8
```

- Table for DHCP configurations on Main-Router:

| DCHP Pool Name | Network Address | Subnet Mask | Default Gateway | Excluded IP Addresses |
|---|---|---|---|---|
| 192.168.1.1 | 192.168.1.0 | 255.255.255.0 | 192.168.1.1 | 192.168.1.1 |
| 192.168.2.1 | 192.168.2.0 | 255.255.255.0 | 192.168.2.1 | 192.168.2.1 |
| 192.168.3.1 | 192.168.3.0 | 255.255.255.0 | 192.168.3.1 | 192.168.3.1 |

The excluded IP addresses are the Gateway IP address of the network

# Wireless & Cellular Configuration

- **Access Point Configuration**

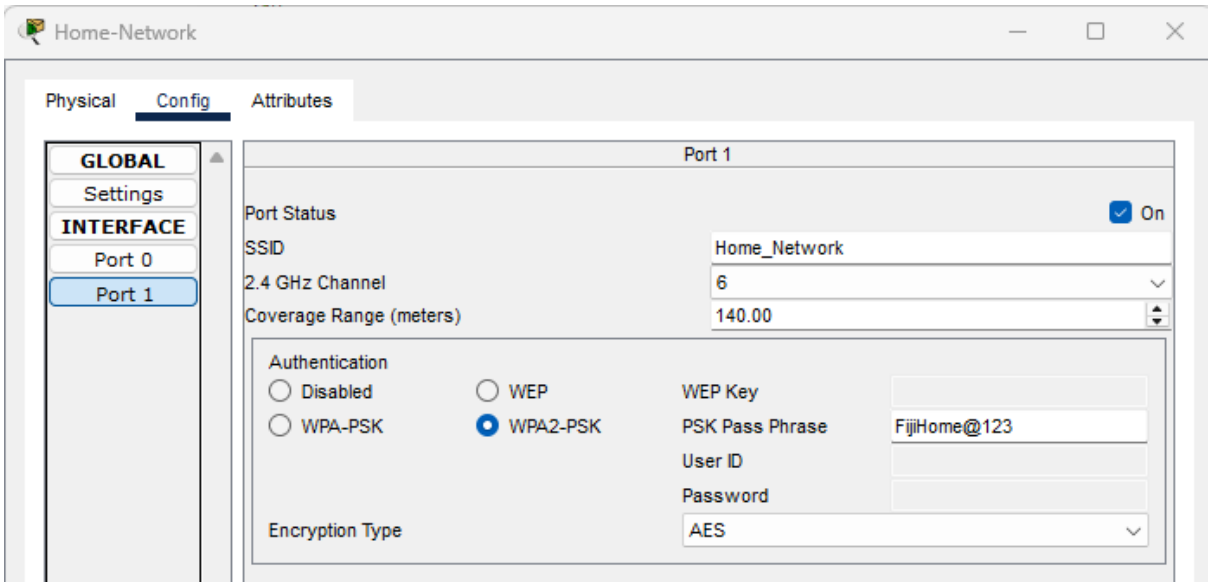



*Figure 10: Screenshot shows configurations on the Access Point*

- Table for wireless Access Point configurations:

| | |
|---|---|
| **SSID (Service Set Identifier)** | Home_Network |
| **Authentication** | WPA2-PSK |
| **Passphrase** | FijiHome@123 |
| **Subnet** | 192.168.1.0/24 |
| **Default Gateway** | 192.168.1.1 |
| **Connected Devices** | Laptop-PT, Tablet-PC, Home Speaker |

- **Cellular Setup**

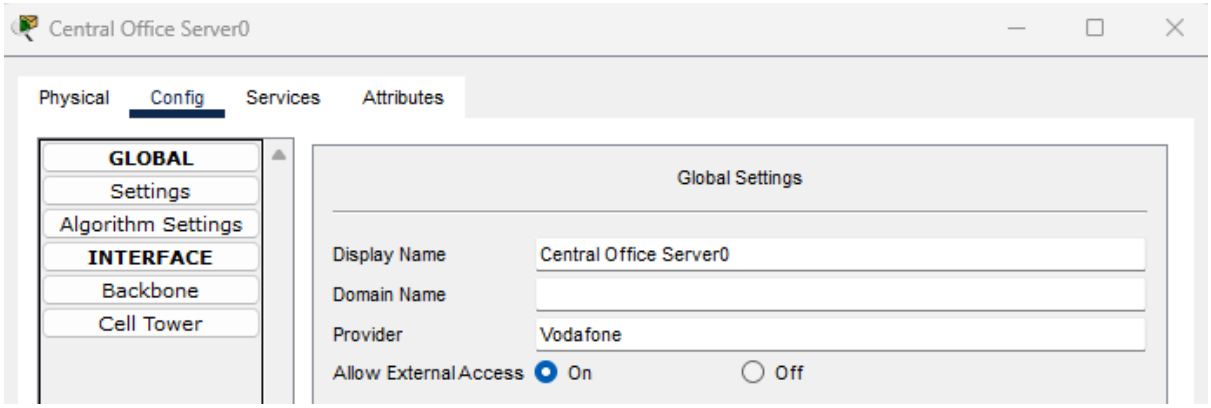



*Figure 11: Screenshot shows the display name of the Central Office Server and the Provider 'Vodafone'*

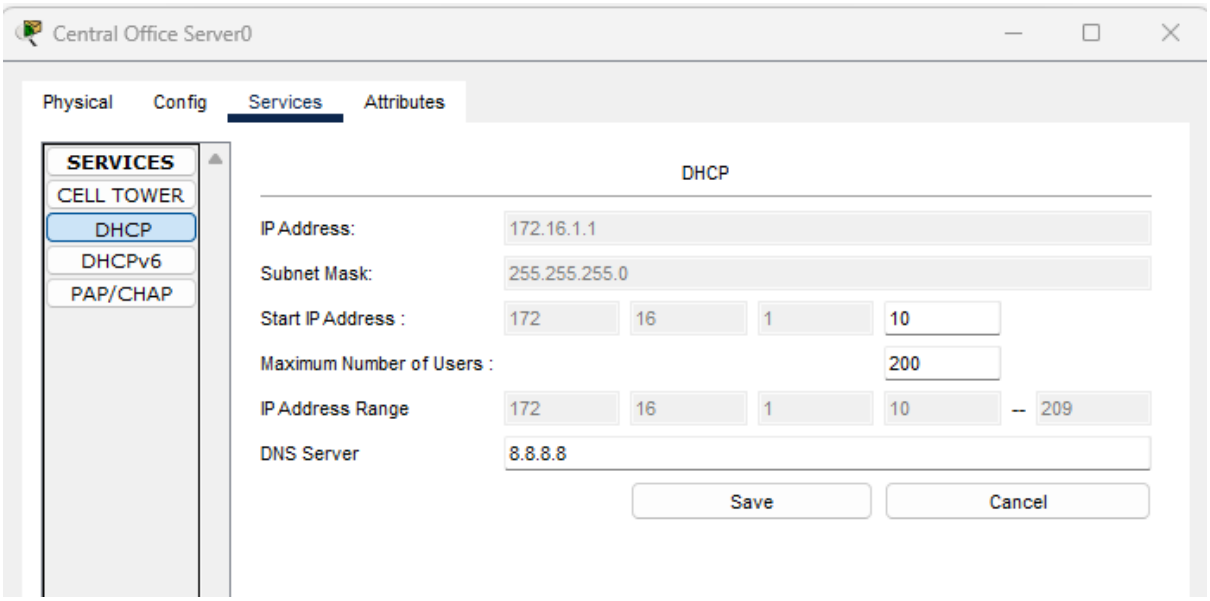



*Figure 12:DHCP configurations on the Central Office Server*

- Table for cellular configurations:

| Device | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|
| **Central Office Server** | 172.16.1.1 | 255.255.255.0 | - |
| **Tavua-Vodafone-Tower1** | DHCP | DHCP | 172.16.1.1 |
| **Ba-Vodafone-Tower2** | DHCP | DHCP | 172.16.1.1 |
| **Lautoka-Digicel-Tower3** | DHCP | DHCP | 172.16.1.1 |
| **Nadi-Digicel-Tower4** | DHCP | DHCP | 172.16.1.1 |
| **Smartphones** | DHCP | DHCP | 172.16.1.1 |

# Connectivity Testing

- Table showing Ping results:

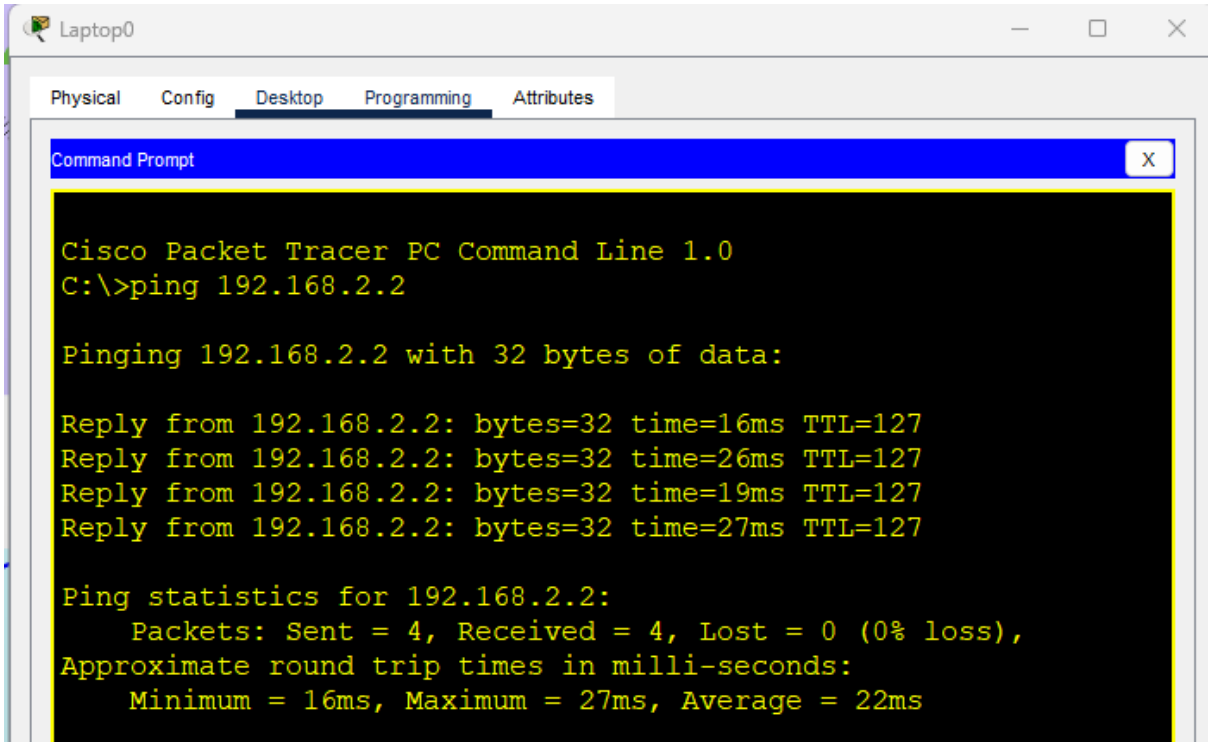



*Figure 13: Test 1 – Laptop to Web-Server*

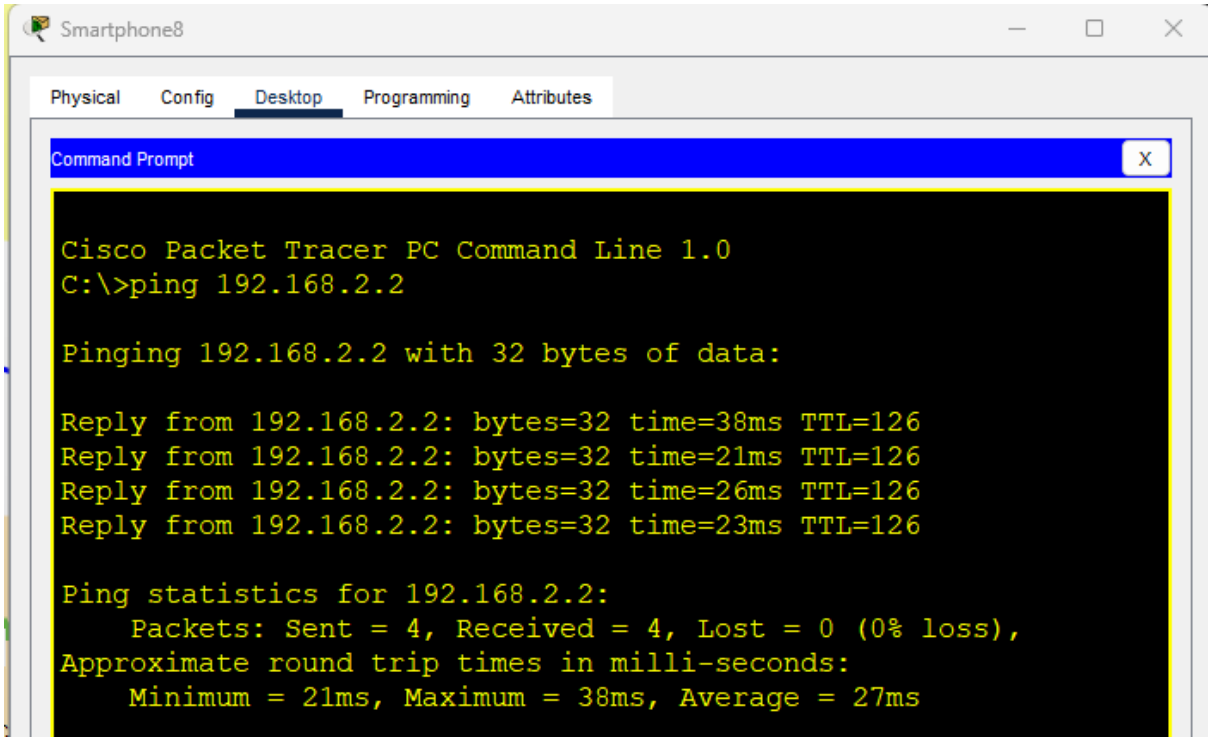



*Figure 14:Test 2 – Smartphone to Web-Server*

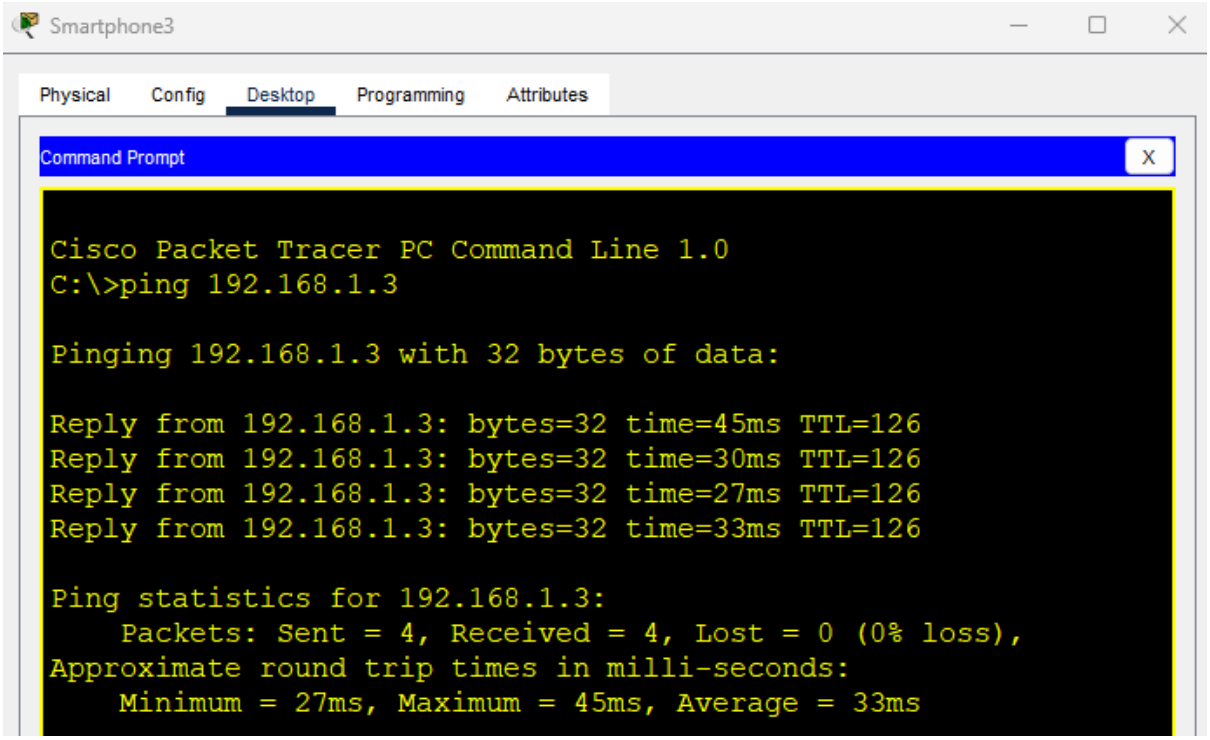



*Figure 15: Test 3 – Smartphone to TabletPC*

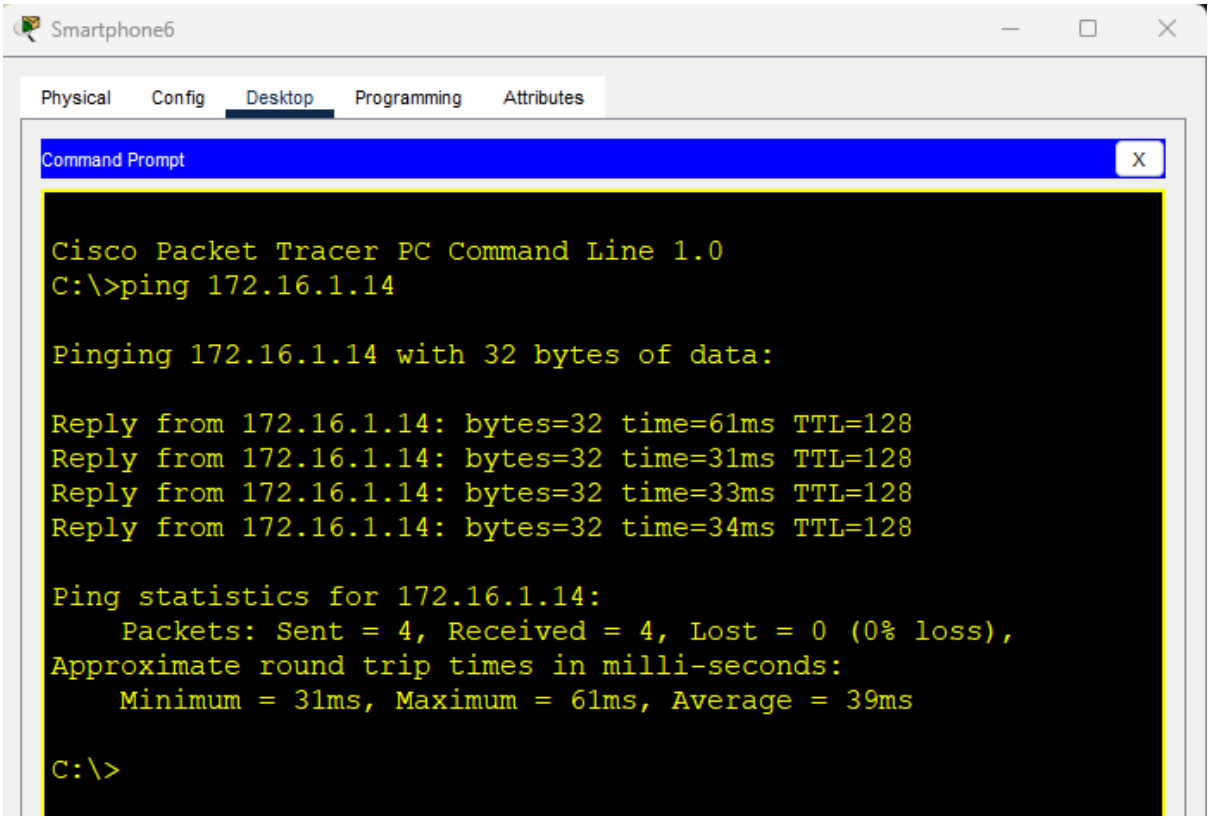



*Figure 16: Test 4 – Smartphone to another Smartphone (Different Service Providers*

# Web Server & HTML Page

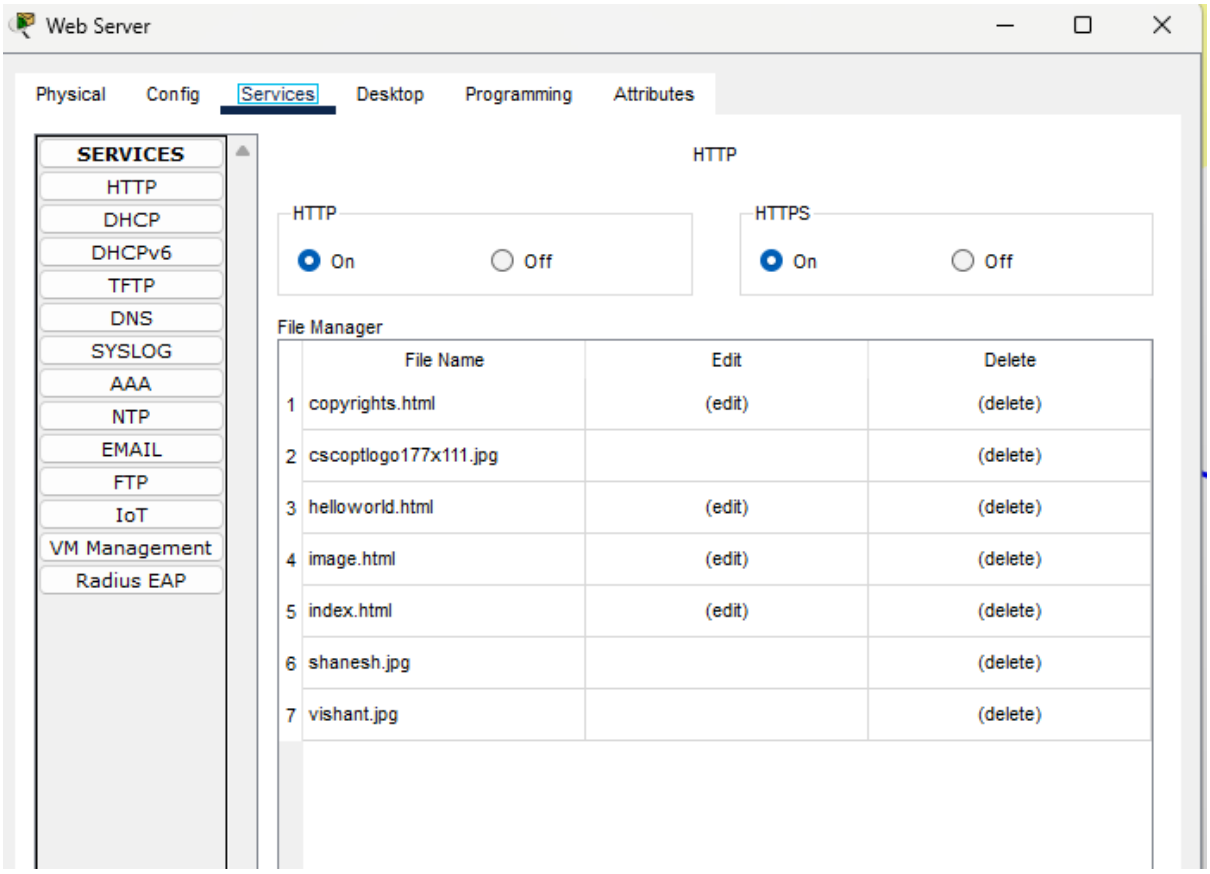



*Figure 17: HTTP services are available on the web server*

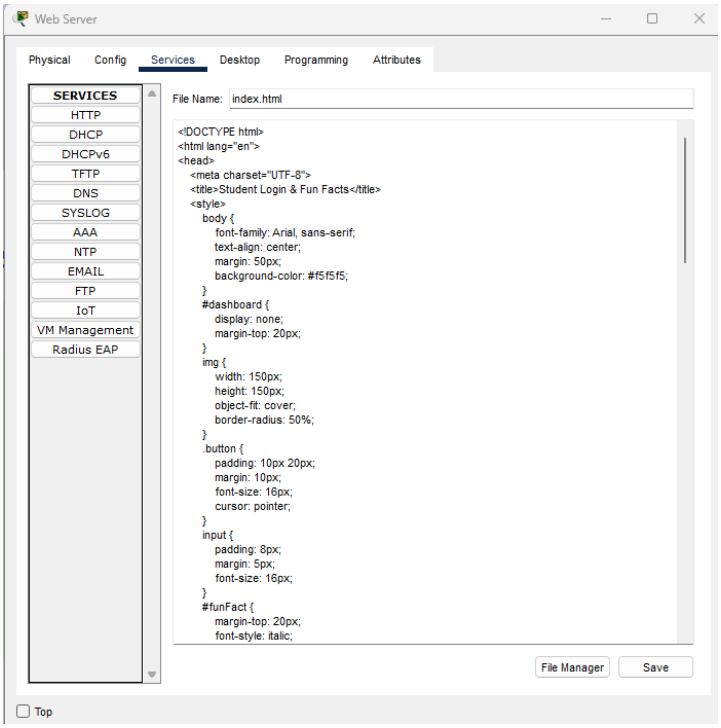



*Figure 18: Edit the index.html file to create a Login Page. Full code is available in the Appendix section*

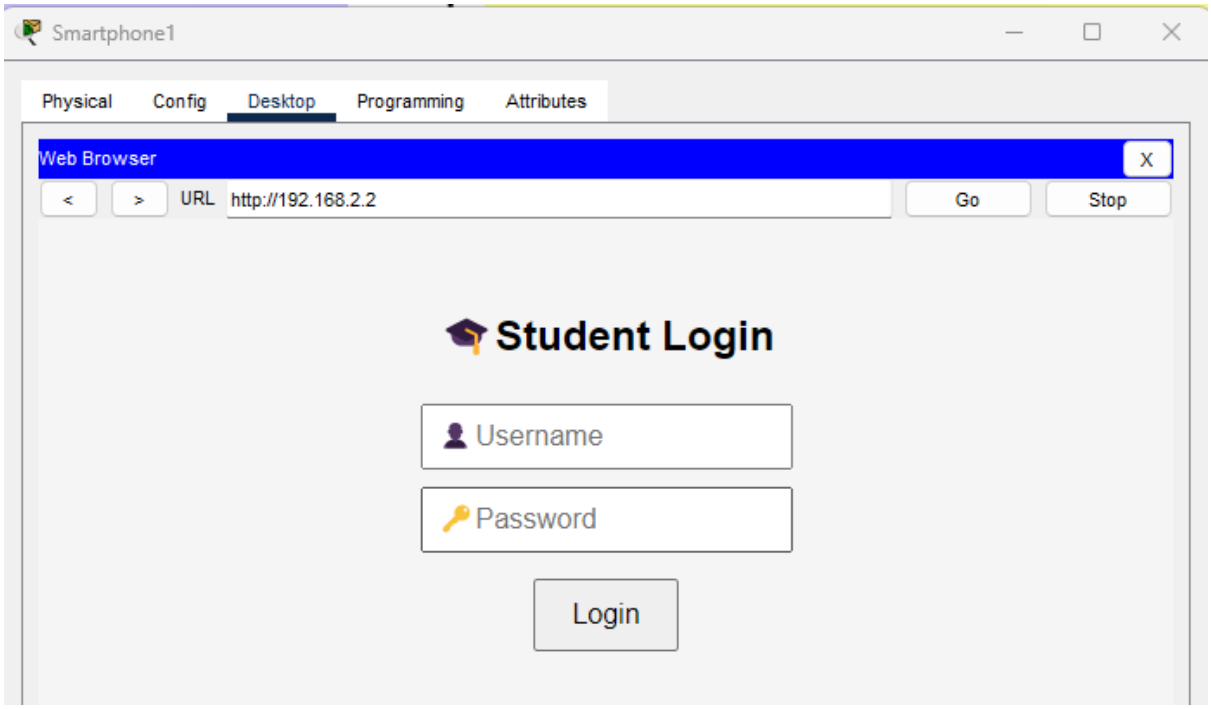



*Figure 19: Access the login page from the smartphone web browser*

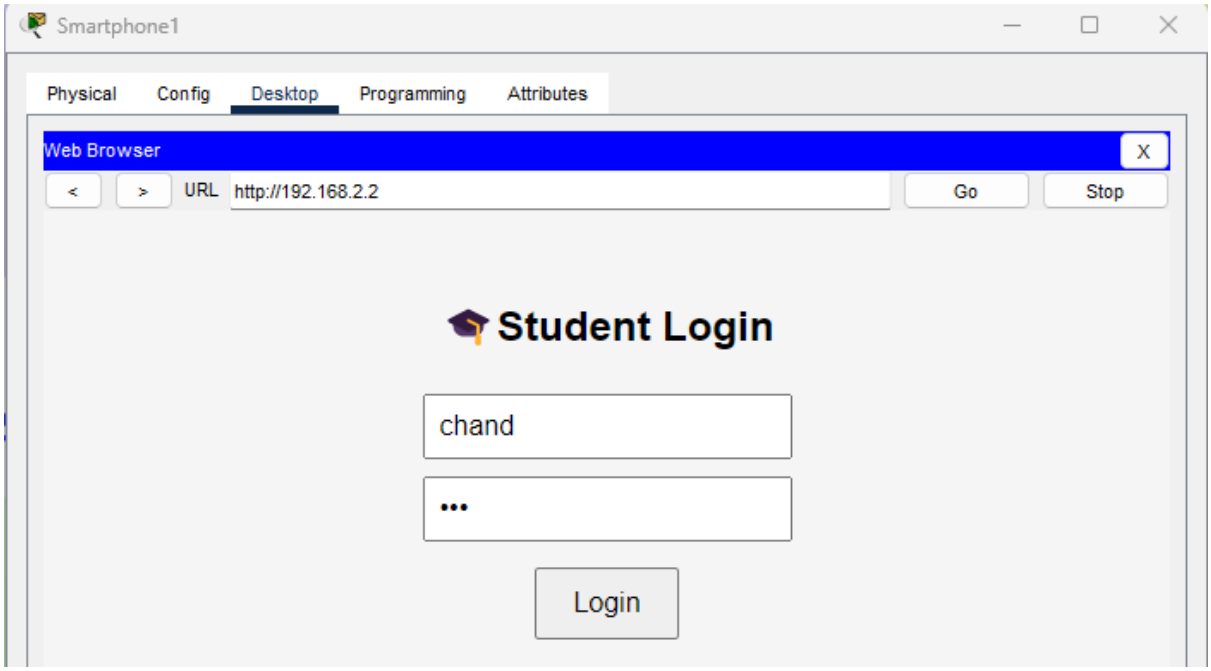



*Figure 20: Login with username and password*

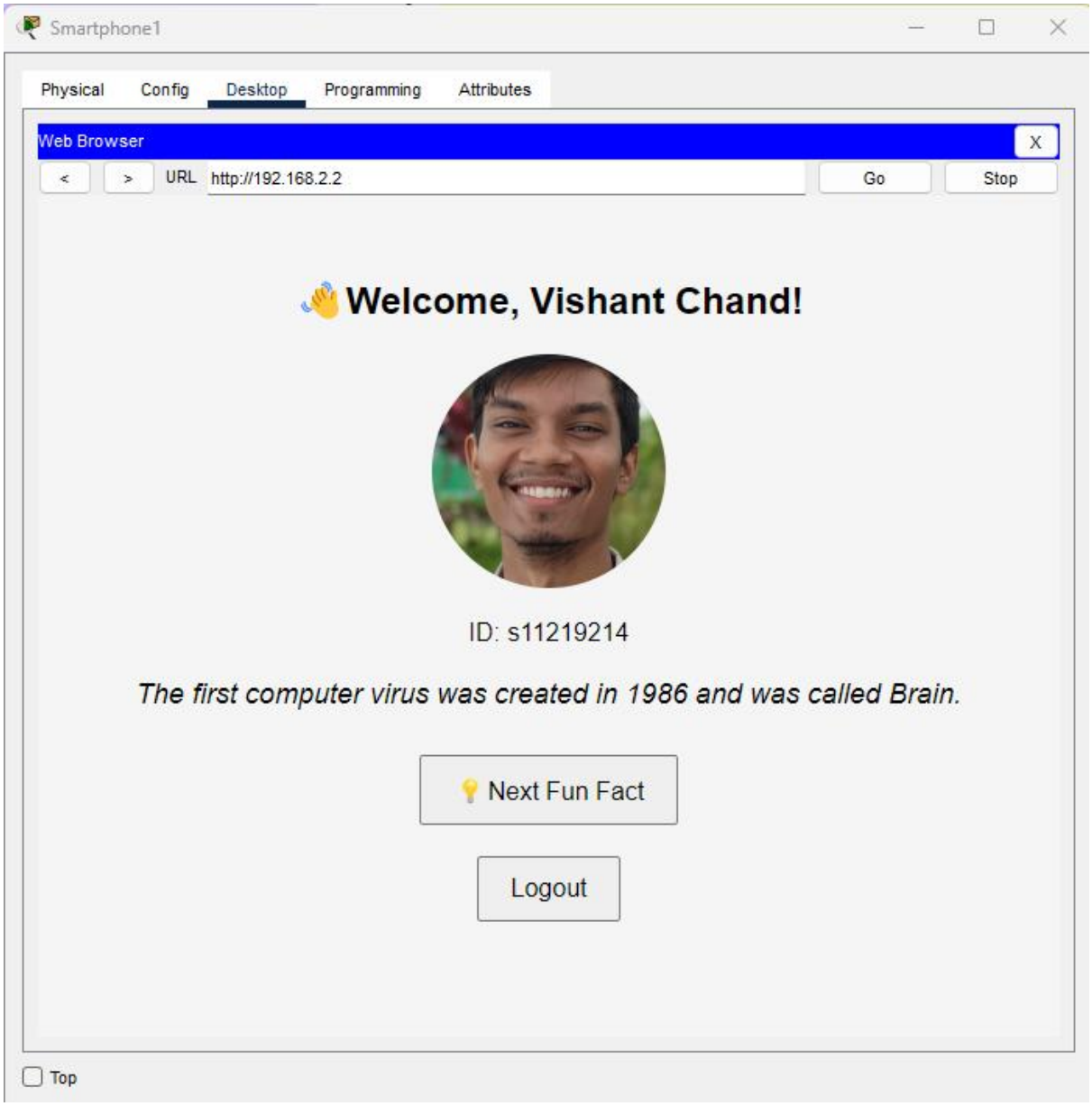



*Figure 21: Once successfully logged in, you can see the student details, fun facts, a 'Next Fun Fact' button and a logout button.*

# Conclusion

A mobile network topology was designed, configured and successfully implemented in packet tracer. The network simulates how mobile networks establish connectivity in home networks and foreign networks with connectivity. DHCP was configured on the router allowing IoT devices to receive IP addresses dynamically. All connectivity was successful between devices through ping, and the web server could be accessed and allowed students to login with the correct credentials. However, there was an issue encountered during the configuration of mobile devices, as the central server kept on assigning IP addresses dynamically to the wireless interface as well as the 3G/4G cellular interface, which often led to duplicate IP addresses. Also, turning the wireless interface off did not work because after restarting the network, IP addresses were again assigned to the wireless interface. Therefore, to resolve this issue my group assigned dummy IP addresses to the wireless interface which allowed the 3G/4G interface to work successfully with no issues.

# Part 2: Live Stream Production

## Introduction

This part requires my group to do a livestream showing how camera switching works. The livestream needs to be minimum of 10 minutes. The livestreaming software we have used is OBS studio. It is a free and open-source software used for video recording and live streaming (OBS, 2025).

## Setup & Methodology

My group decided to livestream a video game online while also showing the camera switching. This would include two separate scenes for the members face Cam and screensharing for the gameplay. The livestream was done on a MSI GF63 laptop with the following specs:

- i5-10500H 6 cores/12 threads
- 16GB DDR5 memory
- 500GB NVMe SSD
- GeForce RTX 3050
- Windows 10 Pro

Also, our choice of streaming platform was twitch.tv. It is a video streaming platform for gaming entertainment, music and more (twitch.tv, 2025).

## Live Broadcast – link (raw footage)

This is the raw footage livestream link available on twitch.tv - **click here**

Two cameras were setup for the facecam and were used for switching using hotkeys during the livestream. The screen shared the gameplay footage. There were some challenges faced during the livestream such as:

- Audio issues (audio isn't detected or delays in audio)
- Device compatibility (streaming on twitch requires your streaming device to have a dedicated GPU)
- Internet connectivity (streaming requires a reliable internet connectivity or else screen share.

## Post-Production (link to edited video)

The editing software used for the post-production video was Capcut. It is a video and graphic editing app (capcut, 2025). Below is the link for the edited video: click here

## Performance Analysis

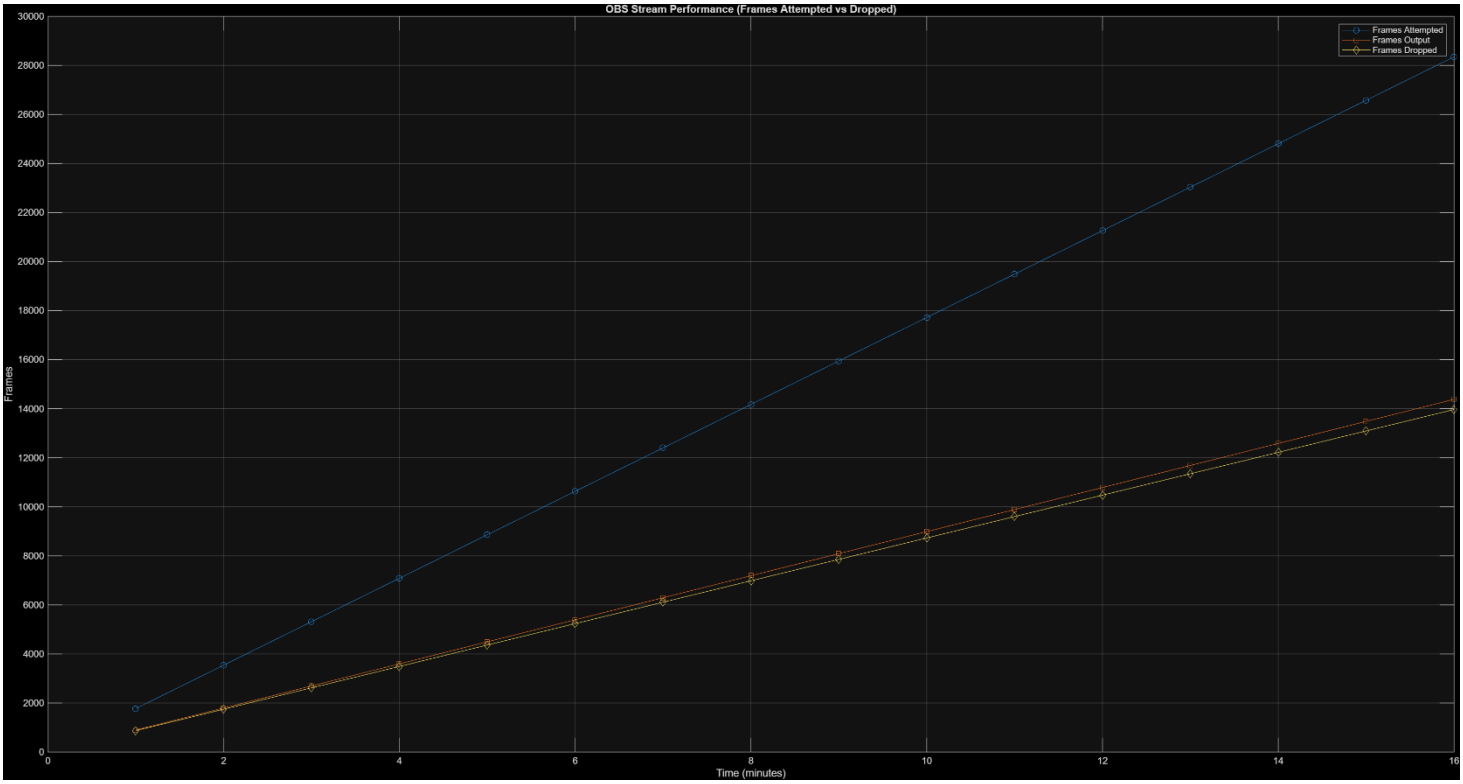



*Figure 22:Performance analysis graph showing frames sent, received and dropped*

The performance analysis was done using the main livestream session log created in OBS studio. Using the log file, my group created the graph in **MATLAB** which is mostly used for data analysis and mathematical computation (MATLAB, 2025).

The graph shows the total stream time which was around **~16 minutes**, frames attempted (**28,364),** successful frame output **(14,390)** and dropped frames **(13,974)** which was about **~49.3%.**

According to the logs created, the livestream resolution was set to 1280 by 720p at 60 frames per second, 1 frame lost due to rendering lag (GPU stalls) and 1/3328 frames skipped due to encoding lag (CPU stalls).

## Conclusion

In conclusion, the requirements for part two of this assignment was met. A livestream was done on twitch.tv using OBS studio and during the initial livestream, camera switching was done between the participants. The raw footage was edited and uploaded to YouTube, and a graph was made using MATLAB for the livestream analysis.

# Part 3: Mobile Application Development

## Introduction

In this part my group will be designing a prototype of a mobile application (mp3/mp4 player). It is important to consider UX/UI designing as UI refers to interactivity and UX focuses on the overall user's experience (Figma, n.d.). Also designing a cross-platform mobile application will allow both Android and iOS consumers to use the mobile app (jetbrains, 2025).

## App Concept

The mp3/mp4 player will be designed as a prototype (low fidelity), therefore the focus while designing will be the front-end considering the structure and optimal user flow experience. The app will have 5 screens including:

- **Homepage** - This will be the first screen the user interacts with and can also navigate to other screens.
- **Now Playing** - This screen shows details about the current song being played including the song name and artist. It also gives the user to play/pause and skip to next song or go back to the previous song.
- **Create New Playlist** - This screen allows users to create new playlists and add mp3 or mp4 file formats to the created playlist.
- Recently Played -Shows the user songs that have been played previously in an order.
- **Playlist Preview** -The user can navigate to this screen to view all playlists created and is currently available

The app mainly targets mainly students and young adults as it allows them to listen to songs while studying, create their own custom playlists, watch short videos during leisure. This mp3/mp4 player app solves the problem of disorganized media files by allowing users to create and manage playlists in one place. It also allows offline access to your content as they are stored locally on your device.

## UI/UX Design

The designing of the prototype was done through Figma which is a UI/UX design tool. It allows users to work collaboratively which was very convenient for me and my partner in this assignment The app mock-up design can be accessed via this link **Click here**

The user can navigate different screens on the app by using the available navigation options on the homepage. Once the user wants to exit the current screen, they can use the 'back to home button' made available on the top left corner of the screen. This gives users a simple and convenient structural flow while using this app.

## Functional Prototype

Below is a video link of a functional prototype of the mobile app (mp3/mp4 player) demonstrating the core features such as creating a playlist, adding media, and basic playback functionality.
Functional prototype link: **Click here**
Alternative link: **Click here**

## Middleware Analysis Report

Additionally, our chosen middleware framework is React native, it allows us to create native apps for both Android and iOS using its built-in class for JavaScript library (Native, 2025). Also, the alternative framework was Flutter which uses dart language, it is an open-source framework for Ui development by Google (flutter, 2025).

| Aspect | React Native | Flutter |
|---|---|---|
| **Language** | JavaScript (with React) | Dart |
| **Codebase** | Single (cross-platform) | Single (cross-platform) |
| **Performance** | Faster for apps that need native components | Fast due to single codebase for both UI and logic. |
| **UI/UX** | Uses native components | Uses its own rendering engine |
| **Development Speed** | Fast, large community and libraries | Fast but fewer supported library |
| **Use Case** | Best for larger development community | Best for single codebase and high customization |

## Role in the Mobile Computing Protocol Stack

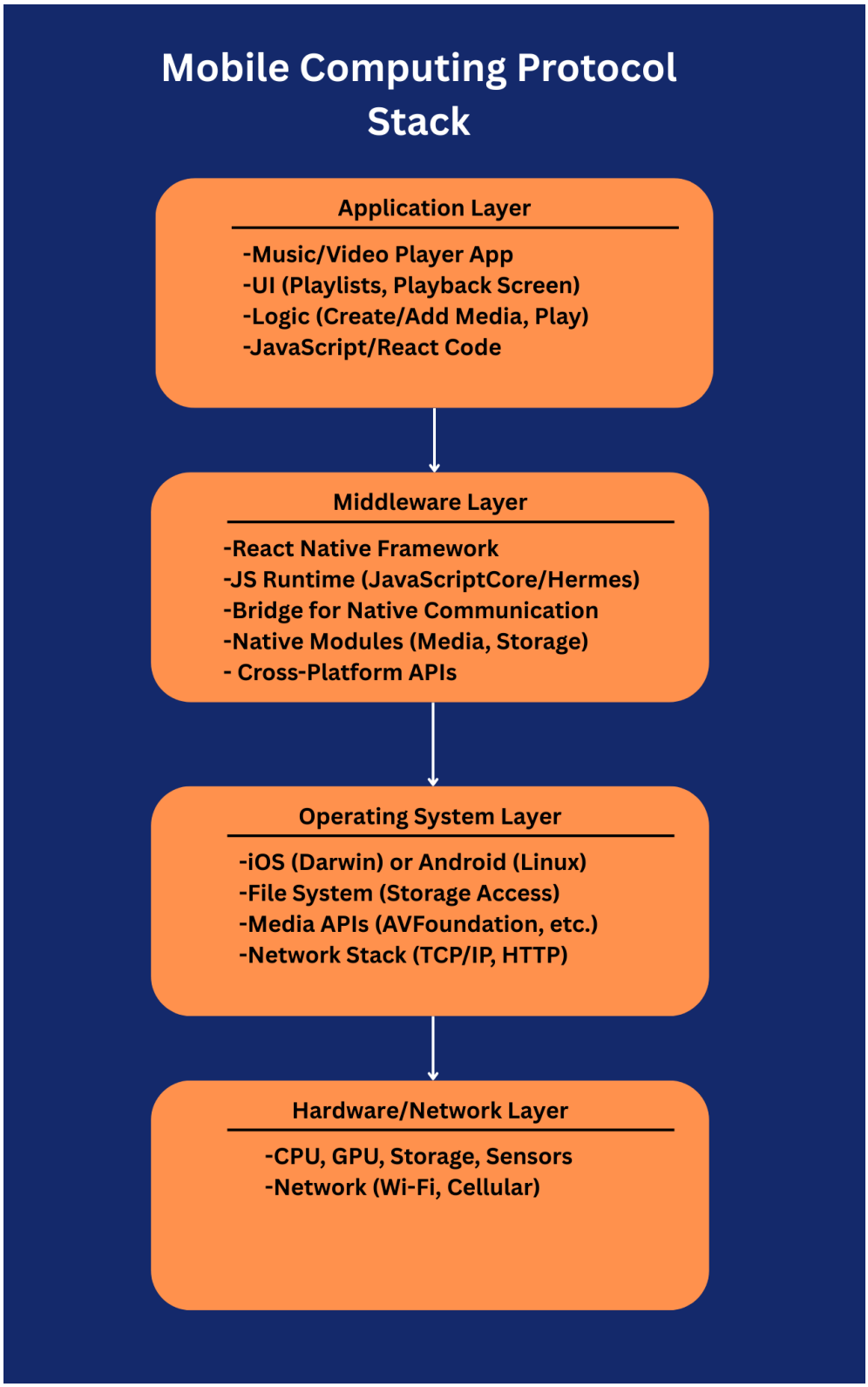



*Figure 23: Mobile Computing Protocol Stack designed in Canva.*

**Application Layer**

- Includes the app's logic written in JavaScript using the react native framework for the mp3/mp4 mobile app. It includes the app UI including the homepage, now playing, etc and other related functionalities.

**Middleware layer**

- The react native framework acts as a bridge and translates the JavaScript code into native instructions to allow the app to work on iOS and Android(cross platform).

**Operating System Layer**

- The react native framework calls the operating system of both Android and iOS indirectly as the OS is used to handle low level tasks such as file storage and handling network.

**Hardware/Network Layer**

- The OS interacts directly with the devices  hardware such as the GPU, CPU and storage for storing files as well as the networking features such as cellular or Wi-Fi connectivity

# Conclusion

In conclusion, the native react framework was chosen for our mp3/mp4 mobile app prototype development. The core features were implemented and tested during the video demonstration. The app supports cross platform  allowing both android and iOS to run the app. Figma was used for the overall design of the app and can be accessed via the link in the documentation.

# References

capcut. (2025, September 5). From https://www.capcut.com/

Figma. (n.d.). *What is the difference between UI and UX?* From https://www.figma.com/resource-library/difference-between-ui-and-ux/

flutter. (2025). From https://flutter.dev/

Fortinet. (n.d.). *What Is Dynamic Host Configuration Protocol (DHCP)?* From https://www.fortinet.com/uk/resources/cyberglossary/dynamic-host-configuration-protocol-dhcp

jetbrains. (2025, July 28). *What is cross-platform mobile development?* From https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-mobile-development.html

MATLAB. (2025, September 5). From https://www.mathworks.com/discovery/what-is-matlab.html

Native, R. (2025). From React Native: https://reactnative.dev/

OBS. (2025, September 5). From https://obsproject.com/

twitch.tv. (2025, September 5). From https://www.twitch.tv/p/en/about/