



GENERAL SIR JOHN KOTELAWALA DEFENCE UNIVERSITY

# Communication Networks

ET 3102



## Classroom > Communication Networks - ET 3102



Photonic and Laser Engine...

Stream

Classwork

People

Grades



Next Generation Cellular N...

Communication Technology

Communication Theory

Communication Systems

Deep Learning

Machine Learning

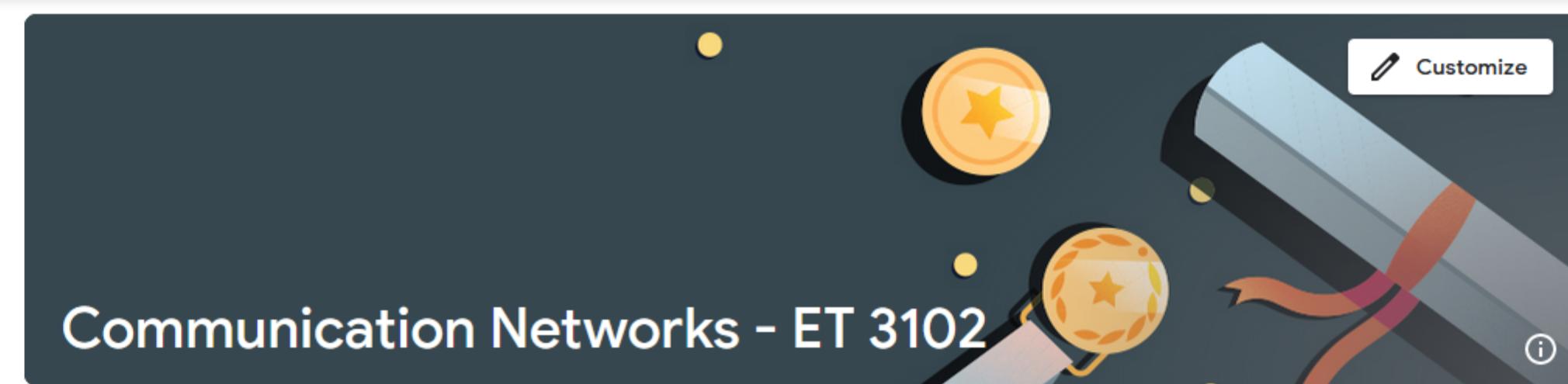
35th Intake : EE & ET

Individual Design Project (...)

Random Signals and Proce...

Archived classes

Settings



Class code



nhqhb2c



Announce something to your class



Upcoming

No work due soon

View all



This is where you can talk to your class

Use the stream to share announcements, post assignments, and respond to student questions



Stream settings

# Outline

Overview on ISO/OSI reference model for open systems, packet and distributed systems and Topologies.

Physical and Data Link Layers.

Network (IP) and Transport Layers (TCP/UDP).

Session Layer, Presentation and Application Layer.

Local Area Network and Wide Area Networks.

# Software Requirements – VS Code Community Edition

The screenshot shows the Visual Studio Code interface with several extensions installed for browser development:

- Run and Debug**: Shows a list of files: Feature\_Engineering2.ipynb, Feature\_Engineering2.html, Untitled-1.html, and exampleDotCom.html.
- VARIABLES**: A sidebar showing variables from the current file.
- WATCH**: A sidebar showing watched files.
- CALL STACK**: A sidebar showing the call stack for the current file.
- LOADED SCRIPTS**: A sidebar showing loaded scripts.
- BREAKPOINTS**: A sidebar showing caught and uncaught exceptions.
- BROWSER BREAKPOINTS**: A sidebar showing breakpoints in browser code.
- JULIA: COMPILED CODE**: A sidebar showing Julia compiled code.

The main editor area displays a large amount of JavaScript code, likely a browser extension or script, with syntax highlighting. The bottom status bar shows the file path E:\KDU Lectures\UG\ET3102ComNetworks\Untitled-1.html, line 17, column 394, and other status indicators like "Julia env: [loading]" and "General Sir John Kotelawala Defence University".



# Session Layer, Presentation and Application Layer

# OSI Protocol Layer Functional Summary

Layer	Name	Functions
7	Application	User level data.
6	Presentation	Standardized data appearance, blocking, text compression.
5	Session	Sessions or logical connections between parts of an application; message sequencing, recovery.
4	Transport	Flow control, end-to-end error detection and correction, priority service.
3	Network	Routing, message blocking into uniformly sized packets.
2	Data Link	Reliable data delivery over physical medium, transmission error recovery, separating packets into uniformly sized frames.
1	Physical	Actual communication across physical medium, individual bit transmission.

# APPLICATION LAYER

- The application layer involves **all protocols that actuate remote process communications in their application form**, although the original OSI model actually separates some functionalities in the Session and Presentation Layers.
- Now that we have reviewed all the underlying structure, we are ready to discuss some examples of Application Layer elements, which represent some familiar applications to the everyday computer user.
- They are the Domain Name Server (DNS), the World Wide Web (WWW), and the email exchange through the Simple Mail Transfer Protocol (SMTP).

# Internet Communication Layers Functional Summary

Layer	Functions	Responsibilities
Application	Prepare messages from user interactions.	User interaction, addressing.
Transport	Convert messages to packets.	Sequencing, reliability (integrity), error correction.
Internet	Convert packets to datagrams.	Flow control, routing.
Physical	Transmit datagrams as individual bits.	Data communication

# Internet Communication Layer Protocol Summary

Layer	TCP Protocols	UDP Protocols
Application	<p>HTTP (Hypertext Transfer Protocol): Used for communicating webpages.</p> <p>SMTP (Simple Mail Transfer Protocol): Used for communicating e-mail.</p> <p>FTP (File Transfer Protocol): Used for receiving or sending files</p> <p>Telnet (Terminal Emulation Protocol): Used for performing remote operations as if directly connected to the host from a terminal and others.</p>	<p>SNMP (Simple Network Monitoring Protocol): Used for controlling network devices.</p> <p>Syslog (System Audit Log ): Used for entering records in the system log.</p> <p>Time: Used for communication and synchronizing time among network devices and others.</p>
Transport	TCP.	UDP.
Internet	IP.	IP.
Physical	Data Communication.	Data communication.

# EXAMPLE

When we enter a URL in a browser :

- Browser checks cache for DNS entry to find the corresponding IP address of website.
- It looks for following cache. If not found in one, then continues checking to the next until found.
  - Browser Cache
  - Operating Systems Cache
  - Router Cache
  - ISP Cache
- If not found in cache, ISP's (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.
- The requests are sent using small data packets that contain information content of request and IP address it is destined for.
- Browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
- Browser sends an HTTP request to the web server. GET or POST request.
- Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, XML and HTML.
- Server sends out an HTTP response along with the status of response.
- Browser displays HTML content

# URL AND ITS ANATOMY

## Uniform Resource Locator

- it has the **location of the resources** which we want to access.
- It is an address of the place where we want to go to interact with or find information.

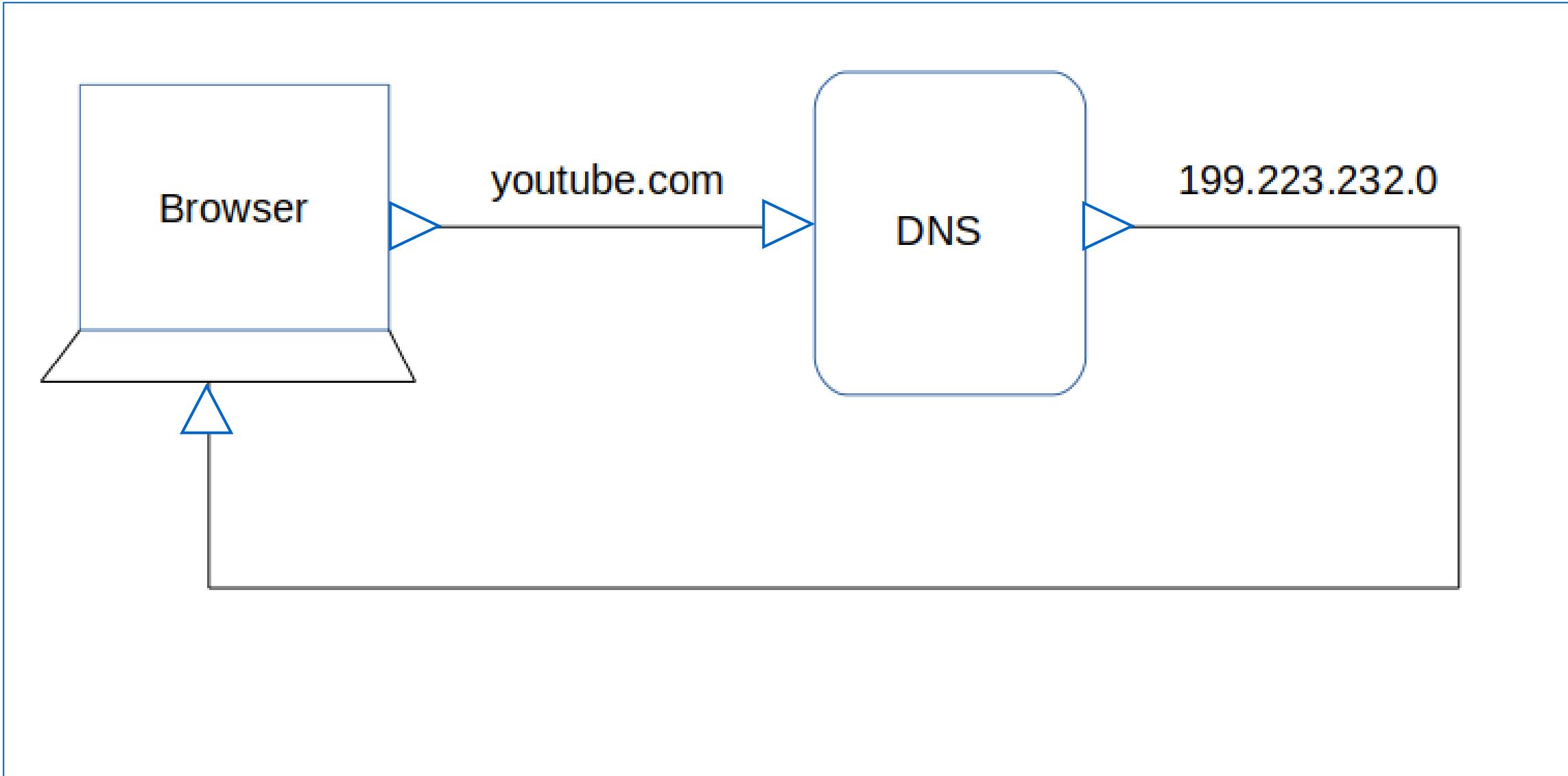
▪ **https://www.kdu.ac.lk/page1**

- This basically tells the browser which **protocol** it should use.
- It can be http, https, ftp, etc.
- A protocol is a set of rules that browser use to communicate over the network.
- 'https' is basically a secure version, i.e. information is exchanged in a secured process.

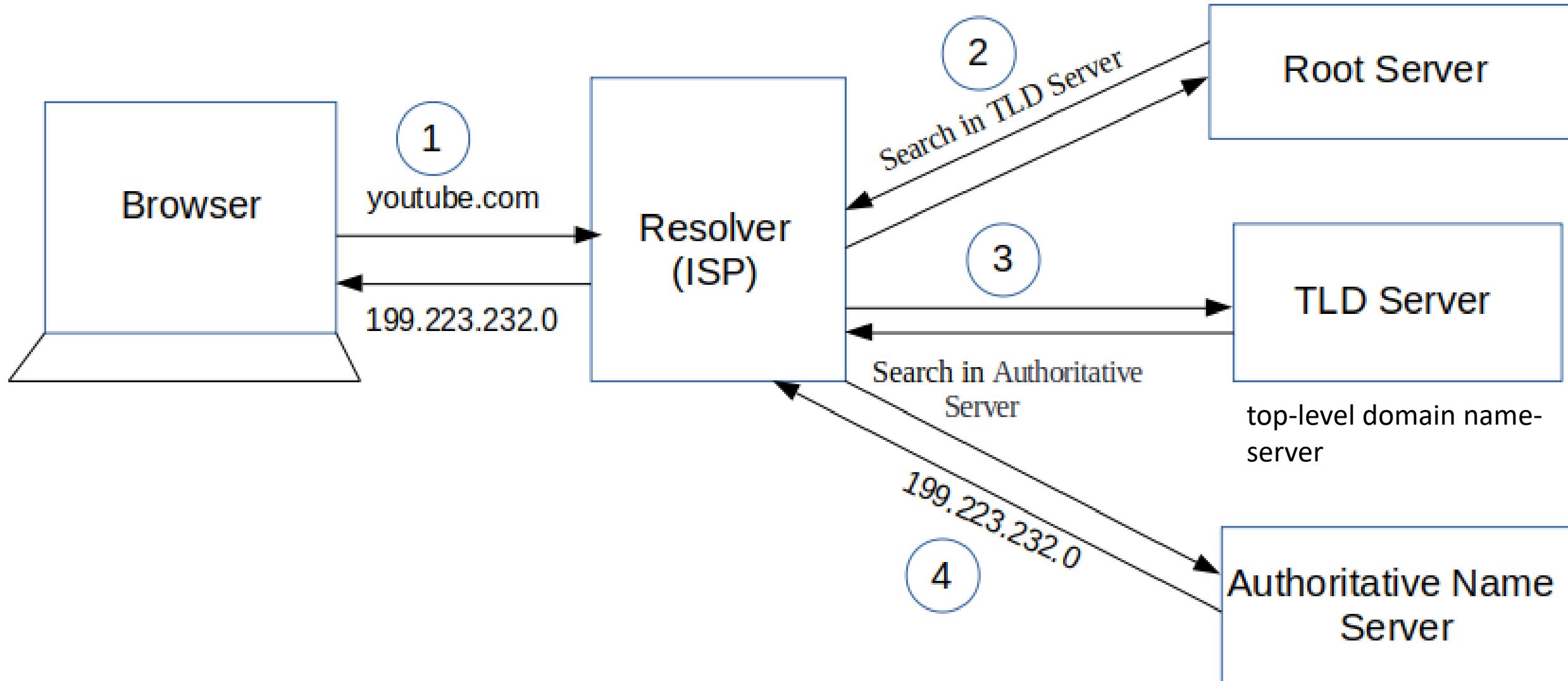
# URL AND ITS ANATOMY

- <https://www.kdu.ac.lk/page1>
  - www.kdu.ac.lk is a domain name.
  - the domain name is the address of the website.
  - provides a **unique identity to the website**
  - When we type any URL, then we are actually transmitting the IP address of the computer which is responsible for **serving** the website content (**hosting**).

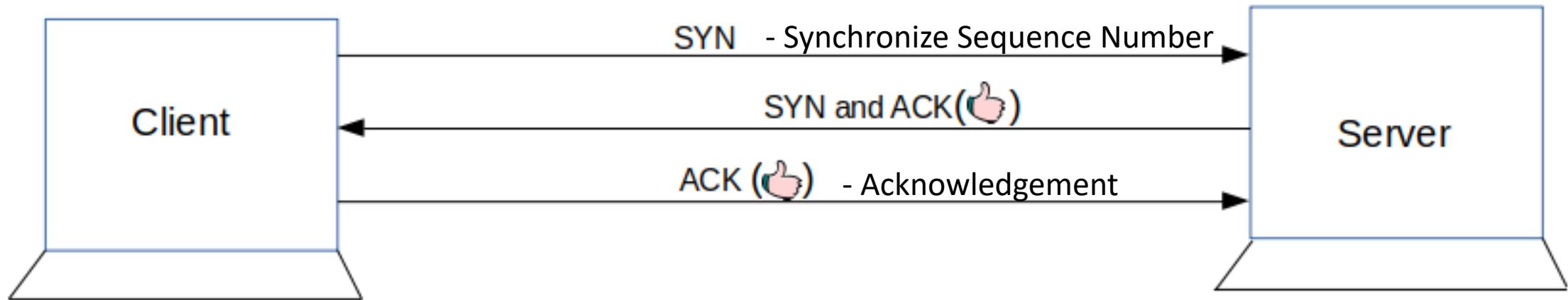
# EXAMPLE



# EXAMPLE

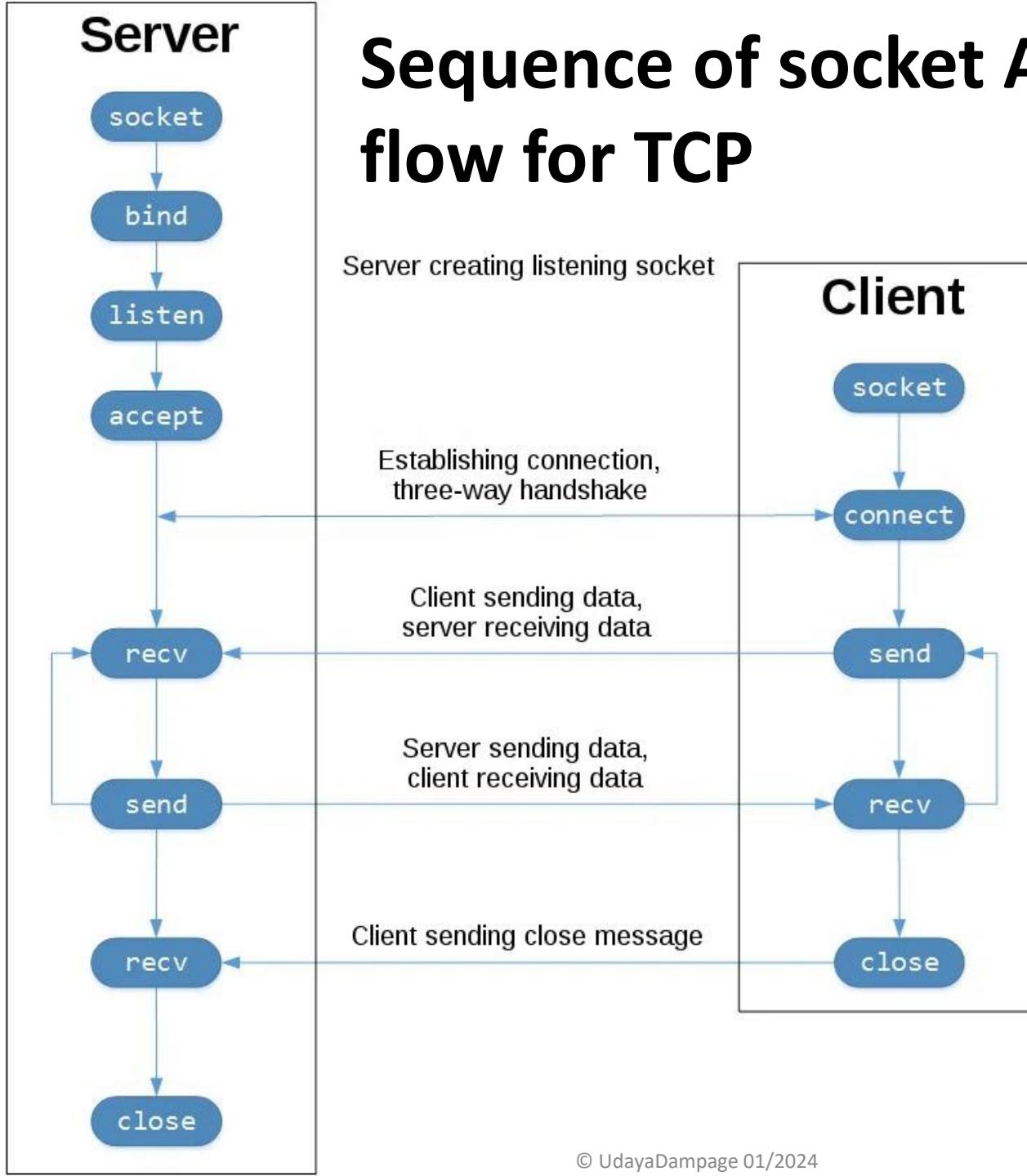


# EXAMPLE



More information in next  
Sub-Module

# EXAMPLE



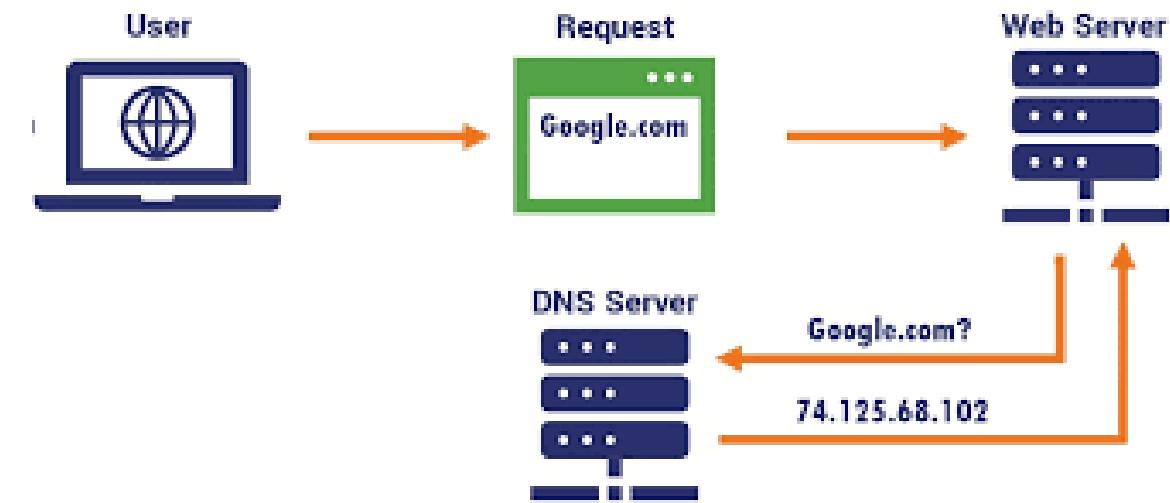
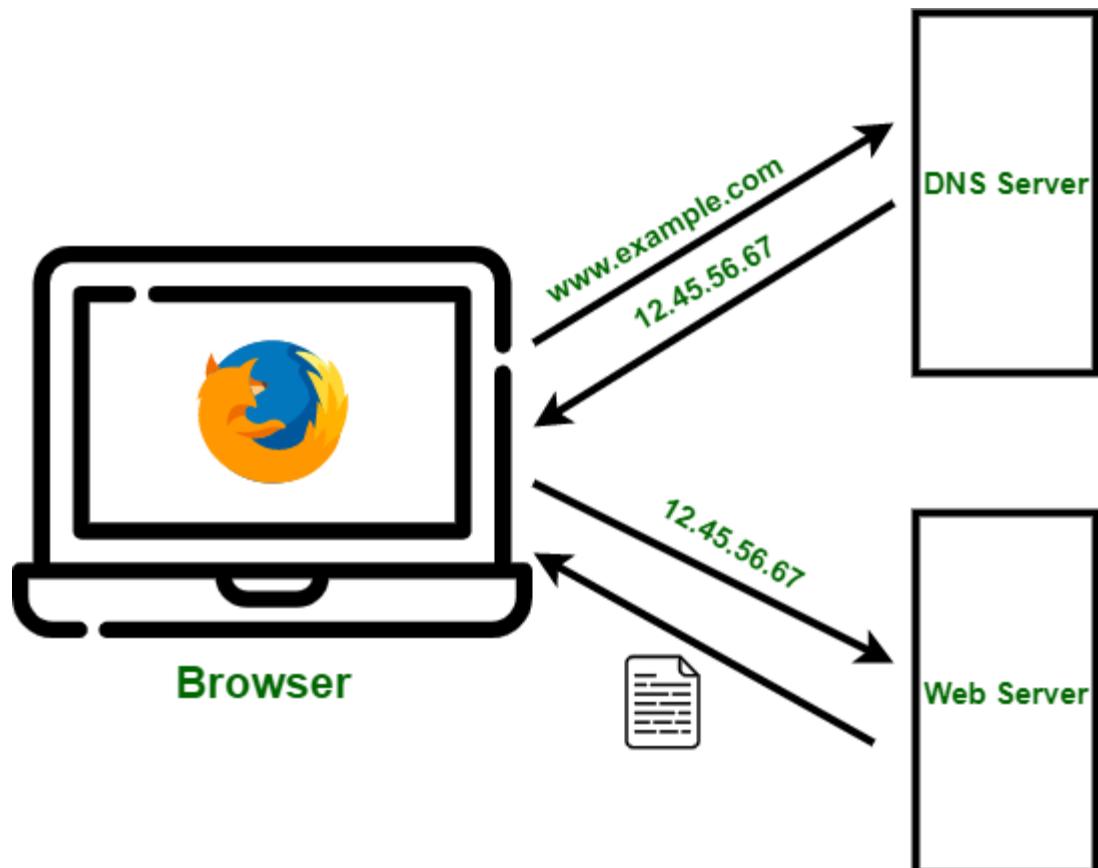
# Sequence of socket API calls and data flow for TCP

More information in next Sub-Module

# Domain Name Server

- At the application layer, DNS is a system used to **assign identifiers for hosts** connected to the Internet.
- The main purpose of DNS is to **translate numerical identifiers into meaningful alphanumerical names**.
- The latter are easier to remember for human users, and are more commonly adopted to denote the host in the everyday use.
- Also the reverse translation is provided – from a name to a numerical address.
- The DNS serves as the “phone book” for the Internet, as it translates the alphanumerical *hostname* of a computer into its IP address.
- An example of name provided by DNS is *google.com*, which is the translation of the IP address 142.251.46.206, as we examined in last week practical sessions.

# Domain Name Server



# Practical 1

```
ipconfig /displaydns
```

```
import socket
```

```
addr1 = socket.gethostbyname('google.com')  
addr2 = socket.gethostbyname('yahoo.com')
```

```
print(addr1, addr2)
```

# HTTP (Hypertext Transfer Protocol)

- HTTP is the foundation of data communication on the World Wide Web.
- It is a **request-response protocol** that enables **clients** (usually web browsers) to **request resources** (such as web pages, images, and videos) from **servers**.
- HTTP uses a **stateless connection**, meaning each request and response pair is independent and doesn't rely on previous connections.
- The protocol operates primarily over **TCP**, using **port 80** by default.

# HTTPS (Hypertext Transfer Protocol Secure)

- HTTPS is a **secure version of HTTP** that uses **encryption** to ensure the confidentiality and integrity of data transmitted between the client and server.
- It employs **Transport Layer Security (TLS)** or its predecessor, **Secure Sockets Layer (SSL)**, to encrypt the data.
- HTTPS operates over **TCP**, using port **443** by default, and is widely used for sensitive data transmission, such as online banking, e-commerce, and login pages.

# Practical 2

1. Type <Cmd> in the search box.
2. Open <Command Prompt>
3. Command: <pip install requests>

# Practical 2

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install requests
Requirement already satisfied: requests in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (2.28.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from requests) (3.0.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from requests) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.10_qbz5n2kfra8p0\localcache\local-packages\python310\site-packages (from requests) (2022.12.7)

[notice] A new release of pip is available: 23.3.1 -> 23.3.2
[notice] To update, run: C:\Users\DELL\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip

C:\WINDOWS\system32>
```

# Practical 2

Administrator: Command Prompt - python

```
C:\WINDOWS\system32>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>> response = requests.get('https://www.google.co.uk')
>>> print(response.text)
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-LK"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="p-v8s0L81moPm4PV-K72xQ">(function(){var _g={kEI:'gDqtZbrWBeGZseMP_q6ymA4',kEXPI:'0,798230,3,562886,4348,207,2414,2390,1132070,870536,327244,380711,16114,28684,23792,12317,17582,4998,17075,6885,31559,2872,2891,4140,4208,3406,606,30668,19390,10632,16335,1014,1,16916,2652,4,57401,2216,27038,6636,7596,1,42154,2,16737,23024,5679,1020,31122,4568,6259,23418,1252,33064,2,2,1,6959,3998,12868,10962,23350,8703,11804,7,1922,52238,20199,68055,5123,2265,765,15816,1804,7734,15103,12432,11813,1428,87,118,289,13205,16258,26609,5209125,1014,2,366,586,691,5992113,1209,2798624,8042,7475465,20540004,16672,43887,3,1603,3,262,3,234,3,2121276,2585,22636438,392913,6869,1294,4636,8408,8028,2728,5331,2,576,13024,4426,10577,1955,3923,17458,13534,7439,1774,2,1296,2052,27,6,5,15,48,217,4832,703,662,210,2863,9,2468,8506,1043,1622,5492,4795,1251,961,149,2428,1246,4,1532,1,6,2815,1063,3073,3334,4,443,218,3,2103,2,210,40,270,2481,378,295,3823,5,1232,208,692,32,805,1388,51,4963,99,464,1017,303,89,24,1830,854,624,255,874,902,1078,2776,1322,3,1629,1343,600,722,1252,108,581,196,80,293,951,332,13,1,604,3,746,61,139,23,1,122,1402,327,212,82,839,1376,4,112,3015,680,1330,134,572,3609,285,128,157,2180,137,1264,1077,1171,47,574,3,953,52,431,3,870,127,216,107,72,3,301,118,683,591,799,427,599,54,111,364,421,405,306,1399,320,4,100,510,43,198,19,266,3600,862,413,4,21355795,360645,218,4285,2335,594,364,19',kBL:'2u_V',kOPI:89978449};(function(){var a;(null==(a=window.google)?0:a.stvsc)?google.kEI=_g.kEI:window.google=_g;}).call(this));}(function(){google.sn='webhp';google.kHL='en-LK';});(function(){
var h=this||self;function l(){return void 0!==window.google&&void 0!=window.google.kOPI&&0!=window.google.kOPI?window.google.kOPI:null};var m,n=[];function p(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||m}function q(a){for(var b=null;a&&(!a.getAttribute||!(b=a.getAttribute("leid")));)a=a.parentNode;return b}function r(a){/^http:/i.test(a)&&"https:"==window.location.protocol&&(google.ml&google.ml(Error("a"),!1,{src:a,glmm:1}),a="");return a}function t(a,b,c,d,k){var e="";-1==b.search("&ei=")&&(e="&ei=")+p(d),-1==b.search("&lei=")&&(d=q(d))&&(e+="&lei=")+d;d="";var g=-1==b.search("&cshid=")&&"slh"!=a,f=[];f.push(["zx",Date.now().toString()]);h._cshid&&g&&f.push(["cshid",h._cshid]);c=c();null!=c&&f.push([{"opi":c.toString()}]);for(c=0;c<f.length;c++){if(0==c||0<c)d+="&";d+=f[c][0]+"+f[c][1]}return"/+(k|"gen_204")+"?atyp=i&ct="+String(a)+"&cad="+(b+e+d)};m=google.kEI;google.getEI=p;google.getLEI=q;google.ml=function(){return null};google.log=function(a,b,c,d,k,e){e=void 0==e;l:e;c||(c=t(a,b,e,d,k));if(c=r(c)){a=new Image;var g=n.length;n[g]=a;a.onerror=a.onload=a.onabort=function(){delete n[g]};a.src=c}};google.logUrl=function(a,b){b=void 0==b;l:b;return t("",a,b)};).call(this);(function(){google.y={};google.sy=[];google.x=function(a,b){if(a)var c=a.id;else{do c=Math.random();while(google.y[c])google.y[c]=[a,b];return!1};google.sx=function(a){google.sy.push(a)};google.lm=[];google.plm=function(a){google.lm.push.apply(google.lm,a)};google.lq=[];google.load=function(a,b,c){google.lq.push([a,b,c])};google.loadAll=function(a,b){google.lq.push([a,b])};google.bx=!1;google.lx=function(){var d=[];google.fce=function(a,b,c,e){d.push([a,b,c,e])}.call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse"));a="1"==c||"q"==c&&!a.elements.q.value?!0:!1}else a!=1;a&&(b.preventDefault(),b.stopPropagation()),!0);document.documentElement.addEventListener("click",function(b){var a;a:{for(a=b.target;a&&a!=document.documentElement;a=a.parentElement)if("A"==a.tagName){a="1"==a.getAttribute("data-nohref");break}a!=1}a&&b.preventDefault(),!0});}.call(this));</script><style>#gbar,#guser{font-size:13px;padding-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !important;text-align:right}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}@media all{.gb1{height:22px;margin-right:.5em;vertical-align:top}#gbar{float:left}}a.gb1,a.gb4{text-decoration:underline !important}a.gb1,a.gb4{color:#00c !important}.gb1 .gb4{color:#dd8e27 !important}.gbf .gb4{color:#900 !important}
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-height:17px}form{margin-bottom:20px}.h{color:#1967d2}em{font-weight:bold;font-style:normal}.lst{height:25px;width:496px}.gsfi,.lst{font:18px arial,sans-serif}.gsfs{font:17px arial,sans-serif}.ds{display:inline-box;display:inline-block;margin:3px 0 4px;margin-left:4px}input{font-family:inherit}body{background:#fff;color:#000}a{color:#681da8;text-decoration:none}a:hover,a:active{text-decoration:underline}.fl a{color:#1967d2}a:visited{color:#681da8}.sblc{padding-top:5px}.sblc a{display:block;margin:2px 0;margin-left:13px;font-size:11px}.lsbb{background:#f8f9fa;border:solid 1px;border-color:#dadce0 #70757a #70757a #dadce0;height:30px}.lsbb{display:block}#WqQANb a{display:inline-block;margin:0 12px}.lsb{background:url(/images/nav_logo229.png) 0 -261px repeat-x;color:#000;border:none;cursor:pointer;height:30px;margin:0;outline:0;font:15px arial,sans-serif;vertical-align:top}.lsb:active{background:#dadce0}.lst:focus{outline:none}</style><script nonce="p-v8s0L81moPm4PV-K72xQ">(function(){window.google.erl={jsr:1,bv:1940,de:true};
```

# Practical 2

The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, and a search bar. The top right corner features standard window control buttons. The left sidebar contains icons for RUN AND DEBUG, VARIABLES, WATCH, and CALL STACK, along with a help section. The main workspace displays a file named Feature\_Engineering2.ipynb, which contains a large block of HTML and JavaScript code. The code editor has syntax highlighting for HTML and JavaScript. The bottom status bar shows Julia env: [loading] and file details like Ln 17, Col 394, Spaces: 4, UTF-8, CRLF, HTML.

# Practical 3

<Html> <!-- This tag is compulsory for any HTML document. -->

<Head>

<!-- The Head tag is used to create a title of web page, CSS syntax for a web page, and helps in written a JavaScript code. -->

</Head>

<Body>

<!-- The Body tag is used to display the content on a web page. In this example we do not specify any content or any tag, so in

output nothing will be displayed on the web page. -->

</Body>

</Html>

# Practical 4

```
<Html> <!-- This tag is compulsory for any HTML document. -->  
  <Head> <!-- The Head tag is used to create a title of web page, CSS syntax for a web page, and helps in written a JavaScript code. -->  
    <title>  
      My Favourite Subject  
    </title> <!-- In this example the title tag is used to specify the title of the web page. -->  
  </Head>  
  <Body> <!-- The Body tag is used to display the content on a web page. In this example we do not specify any content or any tag, so in output  
nothing will display on the web page. -->  
  
  <h6> Communication Networks </h6>  
  <h5> During the Lecture Sessions, I am going to Learn: </h5>  
  <h4> To describe the basis and structure of an abstract layered protocol model. </h4>  
  <h3> To describe analyze and compare functions and applications of OSI layers. </h3>  
  <h2> To analyze of network layers within the simulate networking environment. </h2>  
  <h1> Communication Networks , My Most Favoutite Subject !</h1>  
  </Body>  
</Html>
```

# SMTP (Simple Mail Transfer Protocol)

- SMTP (Simple Mail Transfer Protocol) is **an Internet standard** for **email transmission** across IP networks.
- It is a **text-based protocol** that allows mail servers to **send, receive, and relay** email messages.
- SMTP operates over **TCP**, using port **25** by default, and provides the basic framework for email communication.
- It is often used in conjunction with other protocols like **IMAP** and **POP3** for **receiving and managing email**.

# IMAP (Internet Message Access Protocol)

- IMAP (Internet Message Access Protocol) is an Internet standard protocol used to access and **manage email on a remote mail server.**
- POP3 - **downloads** and **deletes** email from the server.
- IMAP allows users to **access and manipulate** their email **directly on the server**, making it more suitable for managing email across multiple devices.
- IMAP operates over **TCP**, using port **143** by default, or port **993** for **secure IMAP (IMAPS)** connections.

# Practical 5

1. Type <Cmd> in the search box.
2. Open <Command Prompt>
3. Command: <import smtplib>
1. Command: <from email.message import EmailMessage>

# Practical 6 - Sending a message

Create an EmailMessage object and set its properties:

```
msg = EmailMessage()  
msg.set_content('This is the body of the email')  
msg['Subject'] = 'Test Email'  
msg['From'] = 'sender@example.com'  
msg['To'] = 'recipient@example.com'
```

Connect to the SMTP server and send the email:

```
smtp_server = 'smtp.example.com'  
smtp_port = 587 # Use the appropriate port for your SMTP server  
with smtplib.SMTP(smtp_server, smtp_port) as server:  
    server.starttls() # Use TLS encryption  
    server.login('username', 'password') # Log in to the SMTP  
    server  
    server.send_message(msg) # Send the email
```

# Practical 7

1. Type <Cmd> in the search box.
2. Open <Command Prompt>
3. Command: <import imaplib>
1. Command: <import email>

# Practical 8 – Receiving a message

```
imap_server = 'imap.example.com'  
imap_port = 993 # Use the appropriate port for your IMAP server  
mail = imaplib.IMAP4_SSL(imap_server, imap_port)  
mail.login('username', 'password')  
Select the mailbox and search for emails:  
mail.select('inbox') # Select the mailbox  
_, data = mail.search(None, 'ALL') # Search for all emails in the  
mailbox  
email_ids = data[0].split() # Extract the email IDs from the search  
result
```

# Practical 9 – Fetch and parse messages

```
for email_id in email_ids:  
    _, data = mail.fetch(email_id, '(RFC822)') # Fetch the email  
    message  
    raw_email = data[0][1] # Extract the raw email data  
    msg = email.message_from_bytes(raw_email) # Parse the email  
    message  
    print('Subject:', msg['subject'])  
    print('From:', msg['from'])  
    print('To:', msg['to'])  
    print('Body:', msg.get_payload(decode=True).decode('utf-8'))  
    Log out and close the connection:  
    mail.logout()
```

# FTP (File Transfer Protocol)

- FTP (File Transfer Protocol) is a standard network protocol used to **transfer files** between a **client** and a **server** over a TCP-based network, such as the Internet.
- FTP uses a **client-server architecture** and employs **separate control** and **data connections** to facilitate the transfer of files, making it more efficient and reliable.
- The protocol operates over **TCP**, using ports **20** and **21** for data and control connections, respectively.

# Practical 10 – Connecting to FTP Server

1. Type <Cmd> in the search box.
2. Open <Command Prompt>
3. from ftplib import FTP
4. ftp = FTP()
5. ftp.connect('ftp.example.com', 21) # Connect to the FTP server
6. ftp.login('username', 'password') # Log in with your credentials
7. print(ftp.getwelcome()) # Print the server's welcome message

# Practical 11 - Listing Directories and Files

To list the contents of a directory, use the `dir()` method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server)
```

```
ftp.dir() # List the contents of the current directory
```

# Practical 12 - Changing Directories

To change the current working directory on the server, use the cwd() method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server)
```

```
ftp.cwd('/path/to/directory') # Change the current working directory
```

# Practical 13 - Creating and Removing Directories

To create a new directory, use the `mkd()` method. To remove a directory, use the `rmd()` method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server)
```

```
ftp.mkd('/path/to/new/directory') # Create a new directory
```

```
ftp.rmd('/path/to/directory') # Remove a directory
```

# Practical 14 -Uploading Files

To upload a file to the server, use the storbinary() method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server)
```

```
with open('local_file.txt', 'rb') as f:
```

```
ftp.storbinary('STOR remote_file.txt', f) # Upload the file to the server
```

# Practical 15 -Downloading Files

To download a file from the server, use the retrbinary() method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server)
```

```
with open('local_file.txt', 'wb') as f:
```

```
ftp.retrbinary('RETR remote_file.txt', f.write) # Download the file from the server
```

# Practical 16 -Disconnecting from Server

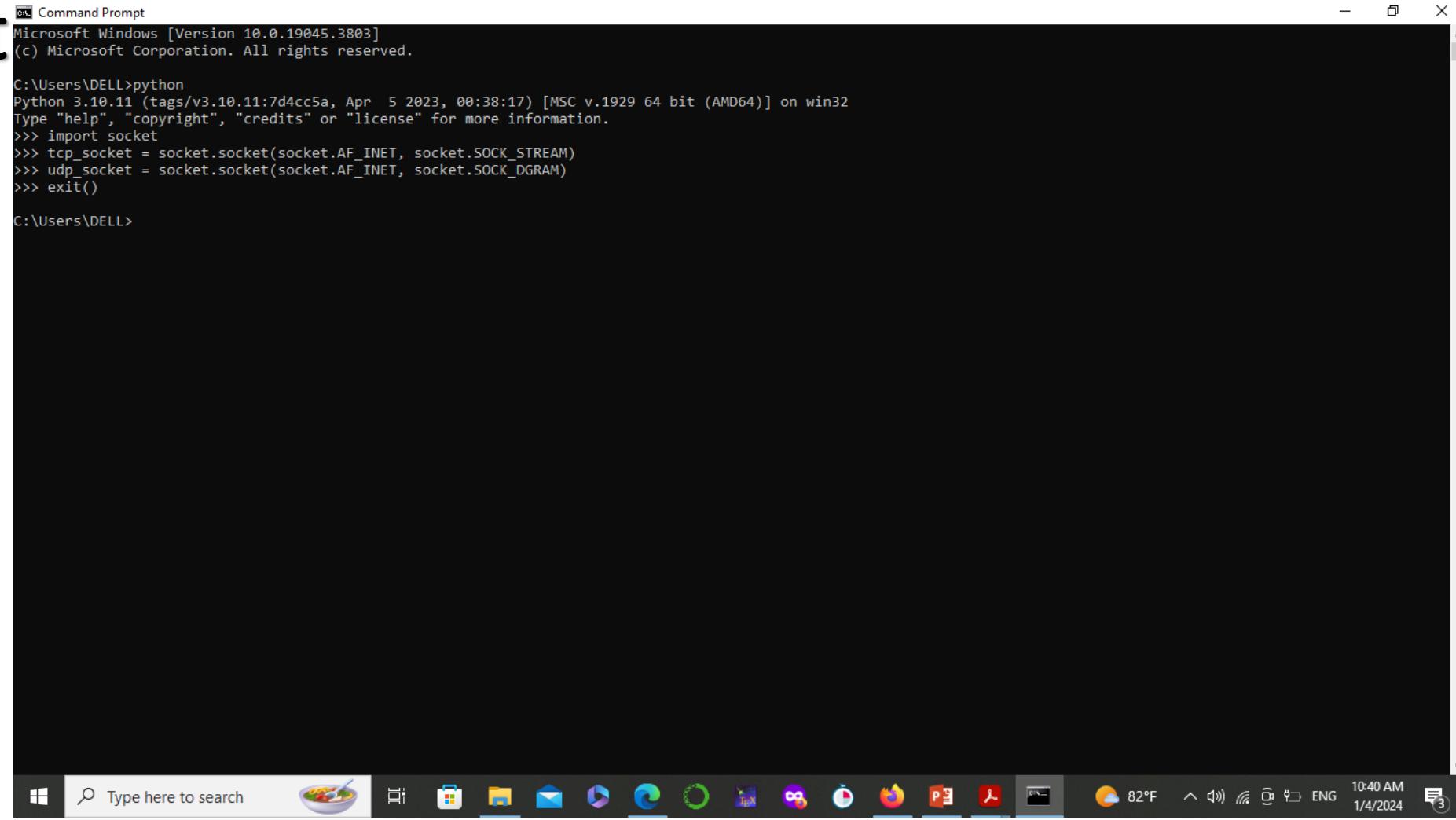
To close the connection to the server, use the quit() method:

```
from ftplib import FTP
```

```
# ... (connect and log in to the FTP server and perform operations)
```

```
ftp.quit() # Disconnect from the server
```

# Practical -Revisit



```
Command Prompt
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import socket
>>> tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
>>> udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
>>> exit()

C:\Users\DELL>
```

To create a TCP socket using IPv4, you would call: `socket.socket()` function as follows:  
`tcp_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)`

To create a UDP socket using IPv4, you would use the following code:  
`udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)`

To close a socket:

`tcp_socket.close()`

# References

1. Nevio Benvenuto and Michele Zorzi, (2011). Principles of Communications Networks and Systems, John Wiley.
2. Thomas Robertazzi, (2011). Basics of Computer Networking (Springer Briefs in Electrical and Computer Engineering), Springer.
3. <https://aws.amazon.com/blogs/mobile>



# Exercise One

