Technical Orientation: Oral Session

What is a Stack?

The **stack** is a set of frameworks and tools used to develop a software product. This set of frameworks and tools are very specifically chosen to work together in creating a well-functioning software.

Here are some examples of widely used web development technology stacks in use today:

- **MERN** (MongoDB, ExpressJS, ReactJS, NodeJS)
- **MEAN** (MongoDB, ExpressJS, AngularJS, NodeJS)
- **LAMP** (Linux, Apache, MySQL, PHP)

Introduction to MERN Stack

MERN Stack:

MERN Stack is a Javascript Stack that is used for easier and faster deployment of full-stack web applications. MERN Stack comprises 4 technologies namely: MongoDB, ExpressJS, ReactJS and NodeJS. It is designed to make the development process smoother and easier.

Each of these 4 powerful technologies provides an end-to-end framework for the developers to work in and each of these technologies play a big part in the development of web applications.

MongoDB, Express, React, and Node are all used in tandem to build web applications and make up the MERN stack. In this lineup, Node and Express bind the web backend together, MongoDB serves as the NoSQL database, and React makes the frontend that the user sees and interacts with.

All four of these technologies are free, open-source, cross-platform, and JavaScript-based, with extensive community and industry support. Each technology has a unique set of attributes, which, when integrated together, make a simple but effective full JavaScript stack for web development.

NodeJS:

Node was developed as a JavaScript runtime environment built on Chrome's V8 JavaScript engine. Node made it possible to start using JavaScript on the server-side to build a variety of tools and applications beyond previous use cases that were limited to being within a browser. Node has an event-driven architecture capable of asynchronous, non-blocking I/O (short for Input/Output). Its unique non-blocking I/O model eliminates the waiting approach to serving requests. This allows you to build scalable and lightweight real-time web applications that can efficiently handle many requests. Node's default package management system, the Node Package Manager or npm, comes bundled with the Node installation. npm gives you access to hundreds of thousands of reusable Node packages built by developers all over the world and boasts that it is currently the largest ecosystem of open source libraries in the world.

However, npm isn't the only package management system at your disposal. Yarn is a newer package manager developed by Facebook and has been gaining popularity in recent years. It can be used as an alternative to npm, with access to all the same modules from the npm registry and more features that are not yet available with npm.

Learn more about Node at https://nodejs.org/en/, and browse through the available npm registry at https://www.npmjs.com/.

Learn more about Yarn and its features at https://yarnpkg.com

ExpressJS:

Node will enable us to build and run complete full-stack JavaScript applications. However, to implement an extensible server-side application with web application-specific features such as API routing, we will use the Express module on top of Node.

Express is a simple server-side web framework(MVC) for building web applications with Node. It complements Node with a layer of rudimentary web application features that provide HTTP utility methods and **middleware** functionality. In general terms, **middleware** functionality in any application enables different components to be added on to work together.

In any web application developed with Node, Express can be used as an API routing and middleware web framework. It is possible to insert almost any compatible middleware of your choice into the request handling chain, in almost any order, making Express very flexible to work with.

In the MERN-based applications that we will develop, Express can be used to handle API routing on the server-side, serve static files to the client, restrict access to resources with authentication integration, implement error handling, and, essentially, add on any middleware package that will extend the web application functionality as required.

MongoDB:

A crucial functionality in any complete web application is the **data storage** system. The Express module does not define requirements or put restrictions on integrating databases with a Node-Express web application. Therefore, this gives you the flexibility to choose any database option, be it a relational database such as PostgreSQL/MySQL/Oracle or a NoSQL database such as MongoDB.

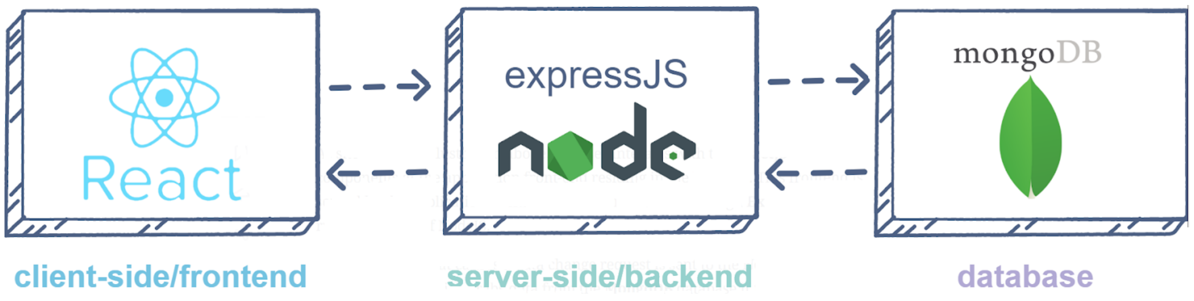
MongoDB is a top choice when deciding on a NoSQL database for any application. It is a document-oriented database that stores data in flexible, JSON-like documents. This means that fields can vary from document to document and data models can evolve over time in response to changing application requirements. Applications that place a high priority on availability and scalability benefit from MongoDB's distributed architecture features. It comes with built-in support for high availability, horizontal scaling using sharding and multi-data centre scalability across geographic distributions. MongoDB has an expressive query language, enabling ad hoc queries, indexing for fast lookups, and real-time aggregation that provides powerful ways to access and analyze data while maintaining performance even when data size grows exponentially.

Choosing MongoDB as the database for a Node and Express web application will make a fully JavaScript-based and standalone server-side application.

ReactJS:

A library created by Facebook. It is used to build UI components that create the user interface of the single page web application. React is a declarative and component-based JavaScript library for building user interfaces. Its declarative and modular nature makes it easy for developers to create and maintain reusable, interactive, and complex user interfaces.

Large applications that display a lot of changing data can be fast and responsive if built with React, as it takes care of efficiently updating and rendering just the right user interface components when specific data changes. React does this efficient rendering with its notable implementation of a virtual DOM, setting React apart from other web user interface libraries that handle page updates with expensive manipulations directly in the browser's DOM. Developing user interfaces using React also forces frontend programmers to write well-reasoned, modular code that is reusable and easier to debug, test, and extend.



Why MERN Stack?

Since all four technologies are JavaScript-based, these are inherently optimized for integration.

Consistency

As JavaScript is used throughout and across the technology stack, developers don't need to learn and change gears frequently to work with very different technologies. This also enables better communication and understanding across teams working on different parts of the web application.

Takes less time to learn, develop, deploy, and extend

Consistency across the stack also makes it easy to learn and work with MERN, reducing the overhead of adopting a new stack and the time to develop a working product. Once the working base of a MERN application is set up and a workflow established, it takes less effort to replicate, further develop, and extend any application.

Widely adopted in the industry

Organizations of all sizes have been adopting the technologies in this stack based on their needs because they can build applications faster, handle highly diverse requirements, and manage applications more efficiently at scale.

Community support and growth

Developer communities surrounding the very popular MERN stack technologies are quite diverse and are growing on a regular basis. With lots of people continuously using, fixing, updating, and willing to help grow these technologies, the support system will remain strong for the foreseeable future. These technologies will continue to be maintained, and resources are very likely to be available in terms of documentation, add-on libraries, and technical support. The ease and benefits of using these technologies are already widely recognized. Because of the high-profile companies that continue adoption and adaptation, and the growing number of people contributing to the codebases, providing support, and creating resources, the technologies in the MERN stack will continue to be relevant for a long time to come.

Development Environment Setup

System Requirements

Operating System: Ubuntu/Any Linux Distro or MacBook Pro

https://ubuntu.com/download/desktop

If you are running Windows, Setup WSL in Windows

https://docs.microsoft.com/en-us/windows/wsl/install-win10

https://ubuntu.com/wsl

CPU: Intel or AMD processor with 64-bit support; Recommended: i5/i7

Minimum system memory (RAM): 8GB/16GB

Selecting Development Tools

There are plenty of options available when it comes to selecting basic development tools such as text editors or IDEs, version control software, and even the development workspace itself. In this section, we will go over the options and recommendations that are relevant to web development with the MERN stack.

IDE or text editors

You can pick any based on your preference as a programmer and then customize it for MERN development.

The following popular options can each be customized as required:

- **Visual Studio Code (**https://code.visualstudio.com/**):** A feature-rich **source code editor** by Microsoft with extensive support for modern web application development workflow, including support for MERN stack technologies. (**Recommended**)

- **Atom (https://atom.io/):** A free, open-source text editor for GitHub that has many packages relevant to the MERN stack from other developers
- **SublimeText (https://www.sublimetext.com/):** A proprietary, cross-platform text editor that also has many packages relevant to the MERN stack, along with support for JavaScript development
- **WebStorm (https://www.jetbrains.com/webstorm/):** A full-fledged JavaScript IDE by JetBrains, with support for MERN stack development

Web Browser

Google Chrome

https://www.google.com/intl/en/chrome/

Chrome Developer Tools

Loading, viewing, and debugging the frontend is a very crucial part of the web development process. Chrome DevTools https://developers.google.com/web/ tools/chrome-devtools, which is a part of the Chrome browser, has many great features that allow debugging, testing, and experimenting with the frontend code and the look, feel, responsiveness, and performance of the UI.

Git Installation

https://git-scm.com/downloads

MongoDB Installation

The guide for MongoDB installation on your local machine is at https://docs.mongodb.com/manual/administration/install-community/

Node.js Installation

https://nodejs.org/en/download/

NPM Installation

https://docs.npmjs.com/downloading-and-installing-node-js-and-npm

POSTMAN Installation

API Testing tool

https://www.postman.com/downloads/

How does the Internet work?

- **"Inter-Network"** forms "Internet".
- The **Internet** sometimes called simply **"the Net"**, is a worldwide system of computer networks of specialized computers called "Routers".
- Each router's job is to know how to move packets along from their source to their destination. When it jumps in between routers, it's called a "**hop**".

Steps explaining how the Internet works:

1. '**Computer A**' sends a message to '**Computer B's**' IP Address.
2. The message is broken into smaller pieces called **Packets**.
3. Packets are sent along the Packet Routing Network to correct **IP addresses**.
4. Packets are subject to **Transfer Control Protocol** to maintain quality.
5. Packets are received and reassembled at 'Computer B's' IP Address.

Key-points:

- Group of Interconnected computer systems.
- To communicate and exchange data.
- Laptops, Desktops, Mobiles, TV's etc.,
- Computer systems = nodes(in network)
- Interconnected via cable media or wireless media.

Resource: https://youtu.be/x3c1ih2NJEg

World Wide Web

The World Wide Web (WWW) is a combination of all resources and users on the Internet that are using the **Hypertext Transfer Protocol (HTTP)**.

- A system that **interconnects** resources on the internet.
- Inter-connection via hyperlinks, referenced with **URI's.**
- We use **Web browsers**, (i.e., Chrome, Safari, Firefox,..,)to have access to web resources.
- We create a web page using HTML, CSS, Scripting Languages.

Resources:

- http://info.cern.ch/hypertext/WWW/TheProject.html
- http://info.cern.ch/
- https://www.history.com/news/the-worlds-first-web-site

Request and Response Cycle

HTTP Protocol

Hyper**t**ext **T**ransfer **P**rotocol is an application protocol that allows web-based applications to communicate and exchange data.

- **HTTP** works as a **request-response** protocol between a client and a server.
- A **client** (browser) sends an **HTTP request** to the **server;** then the server returns a **response** to the client. The **response** contains status information about the **request** and may also contain the requested content.

Just for your Information:

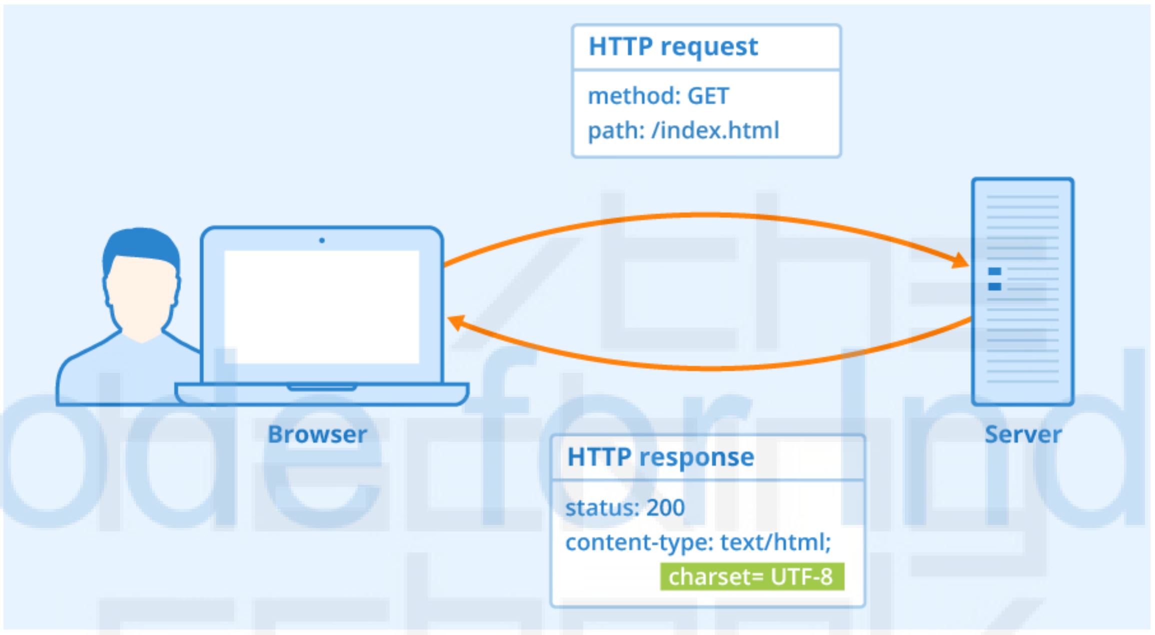
- **Http** is **connectionless:**

  After making the request, if the client disconnects from the server when the response is ready then the server re-establish the connection again and delivers the response.

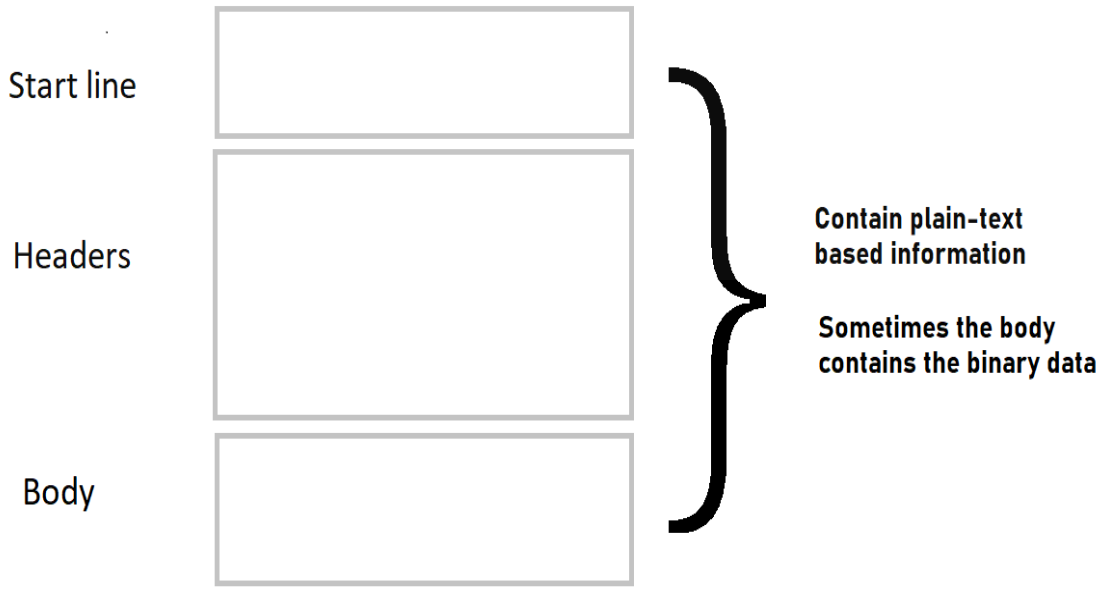
- The **Http** can deliver any type of content as long as two computers are able to read it.
- The **Http** is **stateless:**

  The client and server know about each other just during the current request. If it closes and if the two computers want to connect again, they need to provide the information to each other anew and the connection is handled as the very first one.

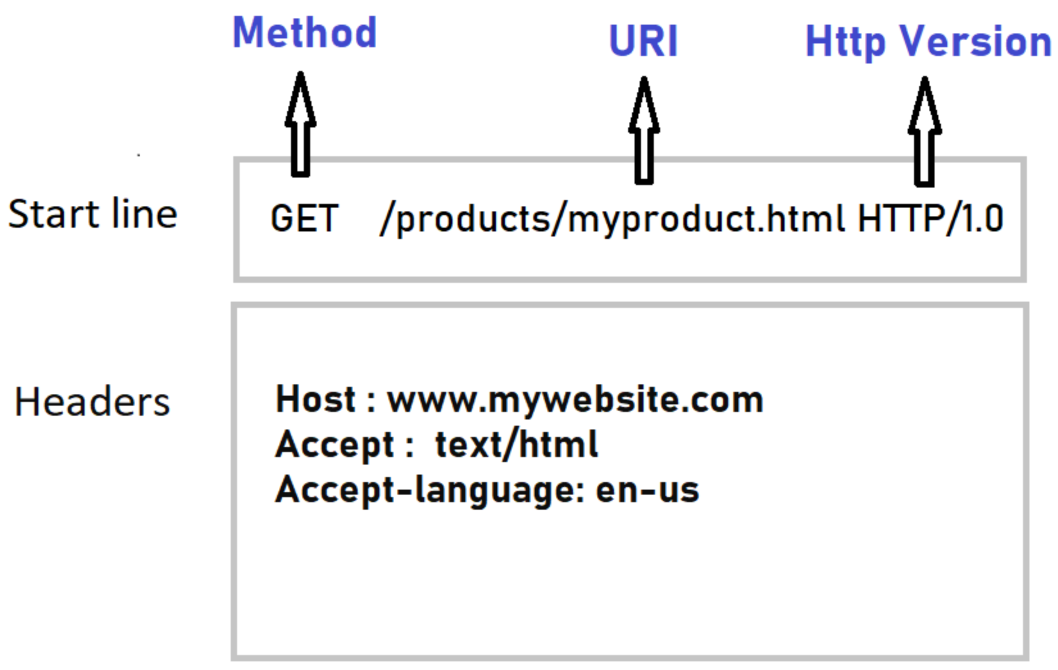
  Below is an **Http Request and Response view:**



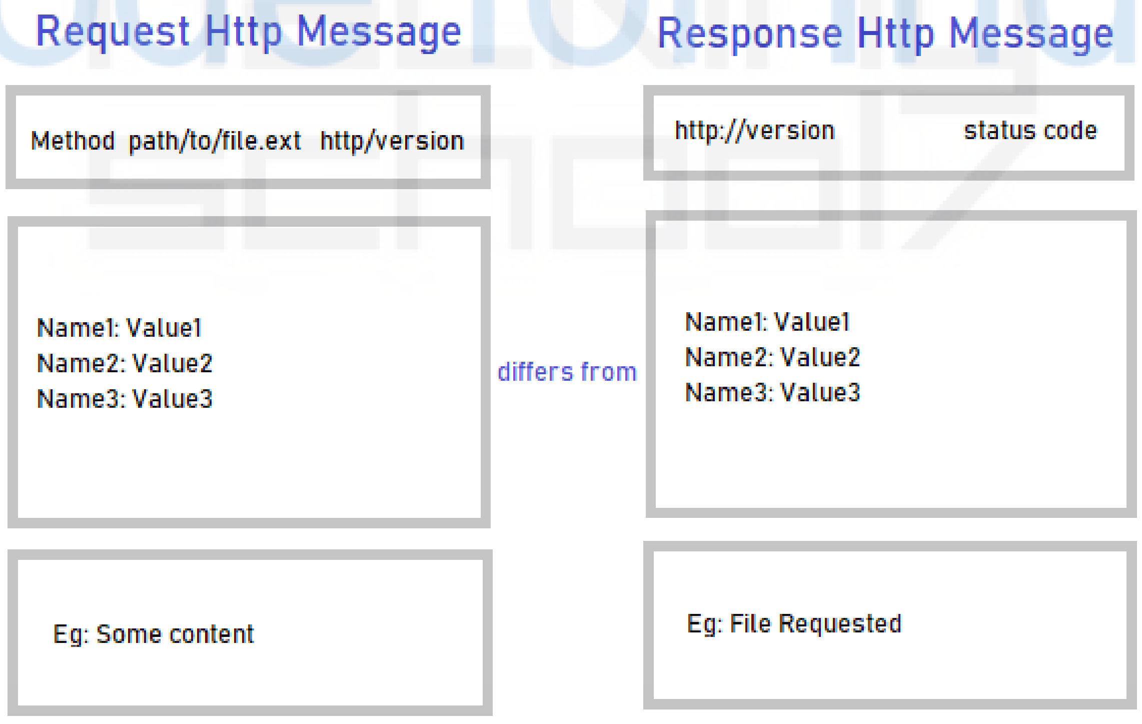
Below is a typical **http** message:


Start line
Headers
Body
Contain plain-text based information
Sometimes the body contains the binary data

Below is a representation of the protocol:



The information in the three sections vary depending on the http message, whether it is a **request** or a **response.**



Now let's set up a simple http server to understand http request-response messages.

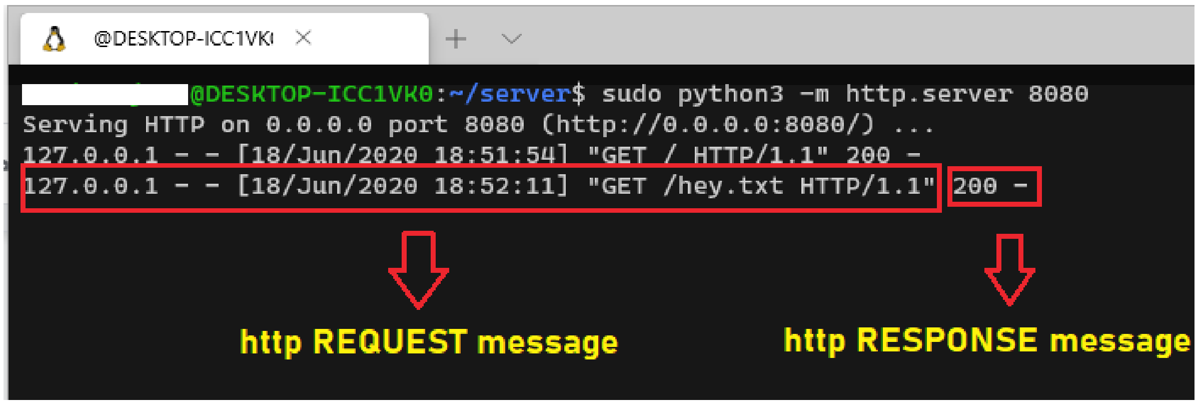
To start the Python Web Server:

In Python 2.7

```
sudo python -m SimpleHTTPServer 8080
```

In Python 3

```
sudo python3 -m http.server 8080
```



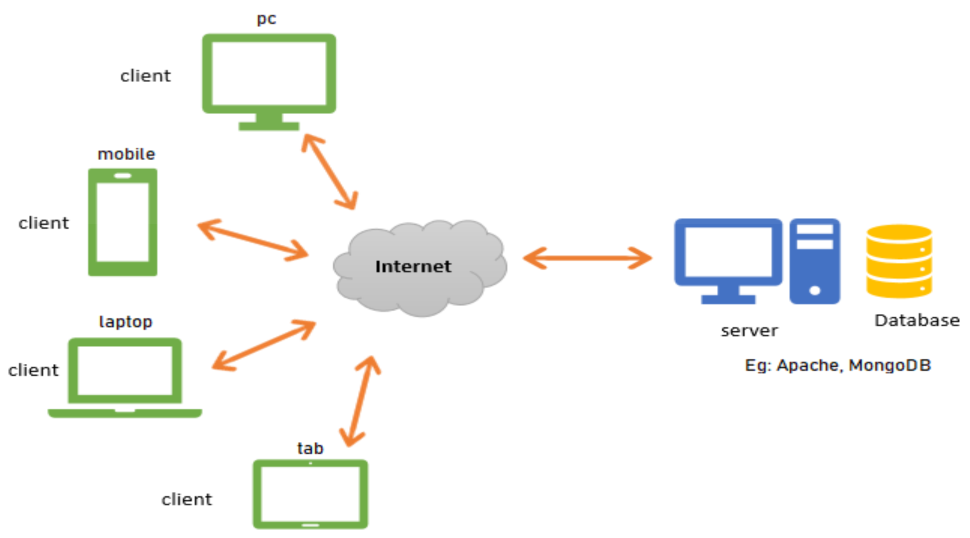
```
[redacted]@DESKTOP-ICC1VK0:~/server$ sudo python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
127.0.0.1 - - [18/Jun/2020 18:51:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Jun/2020 18:52:11] "GET /hey.txt HTTP/1.1" 200 -
```

Key-points:

- The HTTP is the messenger of the web.
- It is a TCP/IP based protocol.
- It is used to deliver the contents for example images, audio, videos, documents etc.,
- The computers that communicate via the Http must speak the Http protocol.

Client-Server Architecture

Client-Server Architecture is a computing model in which the **server** hosts, delivers and manages most of the resources and the **client** consumes the services.



Client:

- A Client can be a machine or a model.
- A Client program is a program that allows the user to make requests.
- A Client, whether it is a machine or a program, is an appliance and a way to make requests through the web.
- **Example:** PC, Laptop, Mobiles, Tabs etc.,

Server:

- A Server is a computer program, not a device.
- High-Performance computers are called servers because they run server programs.
- Servers provide functionality and serve other programs called clients.
- **Example:** Apache, Database

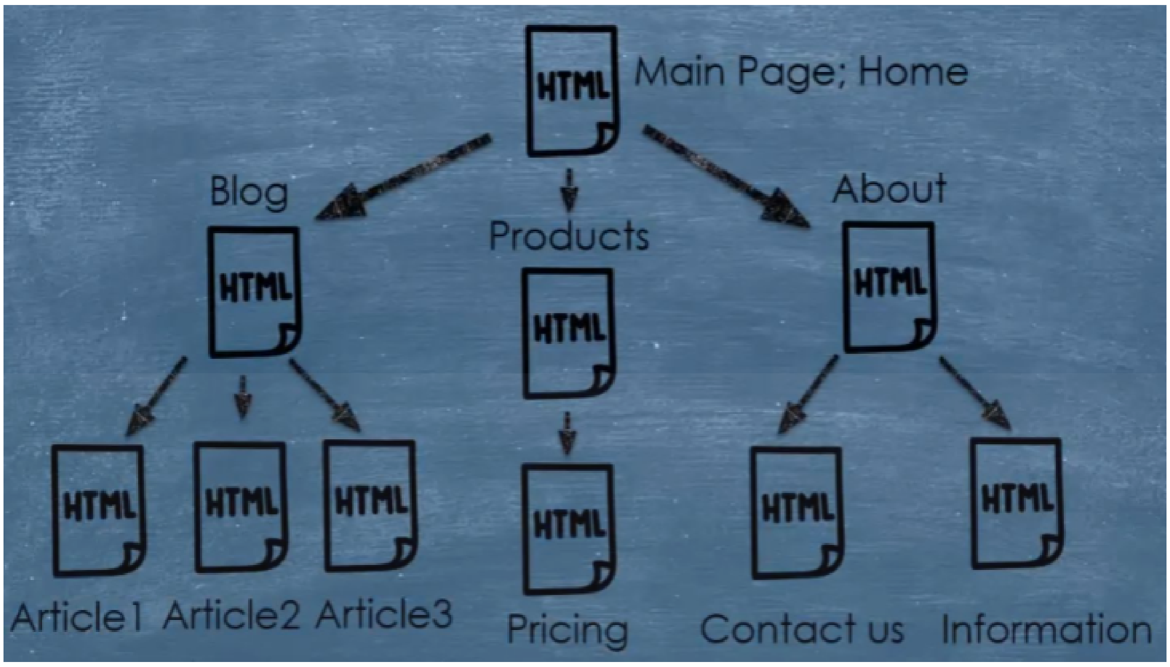
Just for your Information

- A single server can serve **multiple** clients at the same time.
- We can run multiple servers on one single machine, they are called **virtual servers.**

Web pages:

- Web Page is a single document written using **Hypertext Markup Language**.
- **Hypertext** is a text that contains hyperlinks that lead to each other documents, usually texts.
- A **hyperlink** is a text, word, sentence, button, icon, sometimes even a picture, when you click or hover on it, you will be directed to another web page.
- We access a web page by entering its **URL(Uniform Resource Locator)** address using a web browser.
- This is how a URL looks like http://www.example.com/index.html

- A web may contain text, graphics, hyperlinks and many other resources.



Websites:

A group of interlinked and well-structured web pages that exist on the same domain.

There are two types of websites.

1. Static Website
2. Dynamic Website

1. Static Website

- Static Website contains only static web pages.
- A document that, every time it is requested, displays on the browser exactly as it is stored in the server.
- It doesn't change at all unless the creator changes it manually.
- The URL address must end with a document extension such as .html, .php, like http://www.example.com/index.html

2. Dynamic Web pages:

- Websites that contain **Dynamic** web pages.

- Web pages that are generated using scripting languages and programs interacting with a database on the server-side.
- Programs that add, restrict, edit or remove data without the creator's intervention.
- Eg:b The current data and time which you are viewing the web page. The webpage can display your profile information every time you login.

Web Applications:

- A software program that exists on the server and runs using a web browser, through a web page.
- Created using a combination of programming languages and web application frameworks.
- It may use the RAM, allow user interactivity and it's designed for many uses.
- Example: Google+, Facebook, Youtube etc.,

What is URI, URN, URL?

URI: (Uniform Resource Identifier)

- It is a string of characters used to identify a resource on the internet either by location or by name, or both.

URN: (Uniform Resource Name)

- It is a string of characters which is a part of a URI that indicates the name.

URL: (Uniform Resource Locator)

- It is a string of characters but it refers to just the address.
- It is the most used way to locate resources on the web.

URI = URL + URN

Exercise: Find URI, URL, URN in the following link?

https://thehackingschool.com/full-stack-web-development-curriculum.html

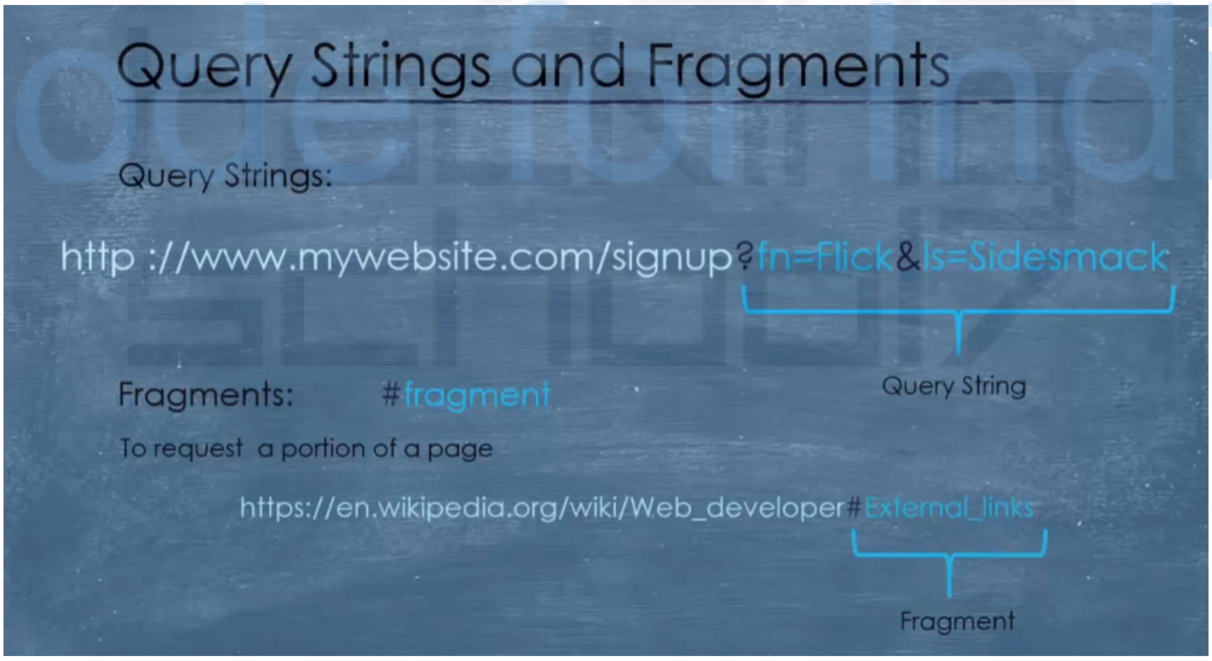
URL Fragment:

A fragment is an internal page reference, sometimes called a named anchor. It usually appears at the end of a URL and begins with a hash (#) character followed by an identifier. It refers to a section within a web page.

Example: http://www.example.com/foo.html#bar

#bar is a fragment.

URL Query String: A query string is a part of a uniform resource locator (URL) that assigns values to specified parameters. A query string commonly includes fields added to a base URL by a Web browser or other client application, for example as part of an HTML form when you click the form to submit.


Query Strings and Fragments
Query Strings:
http ://www.mywebsite.com/signup?fn=Flick&ls=Sidesmack
Query String
Fragments: #fragment
To request a portion of a page
https://en.wikipedia.org/wiki/Web_developer#External_links
Fragment

Resources:

https://stackoverflow.com/questions/4913343/what-is-the-difference-between-uri-url-and-urn

DNS Architecture:

DNS: The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like example.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

For most situations, DNS is concerned with a domain name being translated into the appropriate IP address. To learn how this process works, it helps to follow the path of a DNS lookup as it travels from a web browser, through the DNS lookup process, and back again. Let's take a look at the steps.

Note: Often DNS lookup information will be cached either locally inside the querying computer or remotely in the DNS infrastructure. There are typically 8 steps in a DNS lookup. When DNS information is cached, steps are skipped from the DNS lookup process which makes it quicker. The example below outlines all 8 steps when nothing is cached.

The 8 steps in a DNS lookup:

- A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
- The resolver then queries a DNS root name server (.).
- The root server then responds to the resolver with the address of a Top-Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
- The resolver then makes a request to the .com TLD.

- The TLD server then responds with the IP address of the domain’s nameserver, example.com.
- Lastly, the recursive resolver sends a query to the domain’s nameserver.
- The IP address for example.com is then returned to the resolver from the nameserver.
- The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

What is Root server?

Root servers are an essential part of the infrastructure of the Internet; web browsers and many other internet tools would not work without them. There are 13 different IP addresses that serve the DNS root zone, and hundreds of redundant root servers exist around the globe to handle requests to the root zone.

There are 13 root servers across the globe. The 13 root name servers are operated by 12 independent organisations.

https://root-servers.org/

Why are there only 13 DNS root server addresses?

A common misconception is that there are only 13 root servers in the world. In reality, there are many more, but still, only 13 IP addresses used to query the different root server networks. Limitations in the original architecture of DNS require there to be a maximum of 13 server addresses in the root zone. In the early days of the Internet, there was only one server for each of the 13 IP addresses, most of which were located in the United States.

Today each of the 13 IP addresses has several servers, which use Anycast routing to distribute requests based on load and proximity. Right now there are over 600 different DNS root servers distributed across every populated continent on earth.

MVC Architecture:

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update its data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

Tor (anonymity network)

https://en.wikipedia.org/wiki/Tor_(anonymity_network)

Download Tor

https://www.torproject.org/download/

VPN (A virtual private network)

(VPN) gives you online privacy and anonymity by creating a private network from a public internet connection. VPNs mask your internet protocol (IP) address so your online actions are virtually untraceable. Most important, VPN services establish secure and encrypted connections to provide greater privacy than even a secured Wi-Fi hotspot.

Free VPN Service Providers

https://chrome.google.com/webstore/detail/setupvpn-lifetime-free-vp/oofgbpoabipfcfjapgnbbjjaenockbdp/related

Web Scraping

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

Tools :

https://www.httrack.com/

Command Line

```
sudo wget -m URL
```