**Networking and Internet**

* **The Internet**
  + **Internet by the Numbers**
    - **>4 billion users**
    - **46% of all world population has access**
    - **> 1 billion websites**
    - **10x increases from 1999 to 2013**
* **Internet History**
  + **1958 – Bell Labs – Model (Modulation – Digital to Analog)**
  + **1961 – MIT – Packet Switching**
  + **1968 – ARPANET**
  + **1972 – University Internet Nodes, UCLA Chat**
  + **1974 – Vint Cerf – TCP**
* **How the Web works**
  + **Servers**
  + **Clients**
  + **ISPs**
  + **DNS**
  + **Datacenters**
* **Domain Name Server**
  + **Phone book of the Internet**
  + **Maps domain (**[**www.google.com,www.yahoo.com**](http://www.google.com,www.yahoo.com)**) to IP addresses (112.123.21.22,8.8.22.56)**
  + **Humans remember domains**
  + **Computers work with IP**
* **Internet Service Provider**
  + **Company that provides access to user**
  + **Internet can be over DSL, Phone Line, Cable, Fiber, Wireless and other**
* **Web Terminologies**
  + **Protocols**
  + **Addresses**
  + **Packets**
* **Web Protocols**
  + **TCP/IP**
  + **HTTP**
  + **DHCP**
  + **FTP**
  + **SMTP**
  + **SSH**
  + **Telnet**
* **Web Addresses**
  + **Domain Names**
  + **IP Address**
  + **MAC Address**
* **Domain Names**
  + **Human-readable web address**
  + **<subdomain>, <domain>, <TDL>**
  + **TDL – Top level domain**
    - **.org**
    - **.com**
    - **.net**
    - **Country based - .uk, .in,**
    - **Purpose based - .edu, .aero, .info**
* **IP Addresses**
  + **32-bit, 4-word address (IPv4)**
  + **Uniquely defines a server, a client, a node or a router.**
  + **IPv6 Address – Default in future – 128-bit (16 octet)**
  + **IP allows –**
    - **Subnets**
    - **Gateways**
    - **Private IPs**
* **MAC Address**
  + **Media Access Control**
  + **An ID unique to a hardware Network Interface**
  + **Is not dynamic like IP. Is fixed for a hardware device**
  + **Used by all IEEE 802 Network Technologies**
* **Components of Web**
  + **Web Pages**
  + **Web Sites**
  + **Web Servers**
  + **Search Engines**
* **Web Pages**
  + **A document that can be viewed over web**
  + **Transported over Internet**
  + **Viewed on a browser**
  + **Uses markup (HTML) and styling. Can contain scripts**
* **Web Site**
  + **Collection of webpages**
  + **Also can include other media (audio, images, video)**
* **Web Servers**
  + **A hardware (or software) that hosts the website**
  + **One website can be spread over multiple servers**
  + **One server can host multiple websites**
* **Search Engines**
  + **A website that indexes other websites/webpages**
  + **Helps you find websites**
  + **Uses techniques like ‘crawling’ to catch content for searching.**
  + **Google.com, Bing.com, Yahoo.com**
* **How do Web Technologies work?**
  + **Server**
    - **Hardware**
    - **OS**
    - **Server Framework**
    - **Containers/Servlets**
    - **Server Applications**
  + **Server Side Frameworks**
    - **Ruby**
      * **Rails**
    - **PHP**
      * **Codeigniter**
      * **Laravel**
    - **Node.js**
      * **Express**
      * **Hapi.js**
  + **Client Side Technologies**
    - **HTML (Markup)**
    - **CSS (Styling)**
      * **SASS**
      * **LESS**
    - **Javascript (Scripting/Events)**
      * **jQuery**
      * **Angular**
      * **React**
      * **Backbone**
      * **Knockout**
  + **Server Side Databases**
    - **RDBMS**
      * **MySQL**
      * **Postgres**
      * **Oracle**
      * **MS Database**
    - **NoSQL**
      * **MongoDB**
      * **Crouch DB**
      * **Memcache**
      * **Redis**
  + **Client Side Storage**
    - **localStorage**
    - **sessionStorage**
    - **Cookies**
    - **indexedDB**
    - **cache**
  + **Types of websites**
    - **Static Websites**
    - **Dynamic Websites**
    - **Responsive**
  + **RESTful APIs**
    - **A contract between servers and clients**
    - **GET**
    - **POST**
    - **PUT**
    - **DELETE**
    - **PATCH**
  + **Data exchange formats**
    - **JSON**
    - **XML**
  + **Website design principles**
    - **Reactive websites**
    - **Single-Page Applications**
    - **MVC, MVP, MVVM and MV\* architectures**
    - **Web Application Frameworks**
  + **Latest Developments**
    - **VirtualDOM, ShadowDOM**
    - **Sockets**
    - **Pub/Sub, Push Notifications**
    - **Browser Native APIs (Locations, User data)**
  + **Git Repository**
    - [**https://github.com/coding-blocks-archives/FullStack\_NodeJS\_Live\_March2020**](https://github.com/coding-blocks-archives/FullStack_NodeJS_Live_March2020)