**OVERVIEW OF THE WORLD WIDE WEB**

* **What is the World Wide Web**
  + The **WWW** is a collection of **Web resources** and **applications** that serves as a tool for users to share and obtain information through the Internet.The Web is typically a representation of human knowledge around the globe.
  + Its Web resources are **Hypertext** and **Hypermedia** based which enables resources to be linked to other resources.
    - Hypertext - software system used to link one text-based resource to another
    - Hypermedia - an extended feature of Hypertext which enables handling of multimedia resources such as images and videos.
    - The terms Hypermedia and Hypertext were invented by **Ted Nelson**
  + Applications are software tools used by users to access the Web resources
    - Two forms of Application
      * Web Server
        + It hosts/houses electronic resources
        + Gives response to clients’ request
      * Web Client
        + Sends requests to the server to be able to access certain Web resources.
    - **“Spider”**- A web crawler application which performs **Web Indexing** to obtain all of the information for a certain topic in the Web.
* **Internet** 
  + A common misconception is that the WWW is the same thing as the Internet.
  + serves as a medium between the Web and the Web browser such that it allows users to access information with the use of Web browser.
  + Collection of computers connected to a network
    - Composed of the following:
      * Hosts
      * Media
      * protocols
* **Web Browser**
  + A typical tool used to interact with the WWW
    - Categories
      * Static
        + Information displayed does not change/alter
      * Dynamic
        + Generates/displays information that varies and depends on the client
* **The History of the World Wide Web**
  + 1989
    - The WWW was invented by Sir Tim Berners-Lee in 1989 with the purpose of allowing scientists to easily share scientific resources or documents to other researchers around the world.
      * **Sir Tim Berners-Lee** - Father of modern Web
    - It was developed at **CERN**, a company sir Tim Berners-Lee works for.
    - Sir Tim Berners-Lee proposed the three **Web Technologies**:
      * HTTP
      * HTML
      * URL
  + 1990
    - CERN hosted the first website developed by Sir Tim Berners-Lee in Switzerland
  + 1993
    - CERN made World Wide Web open source such that users around the world can alter it and improve it without a cost.
  + 1994
    - **The World Wide Web Consortium (W3C)** was founded by sir Tim Berners-Lee
      * **W3C**  - is an international organization that provides standards in developing the Web

**HYPERTEXT TRANSFER PROTOCOL**

* **What is HTTP**
  + A communication protocol used to fetch the hypertext and hypermedia resources such as HTML documents in the Web.
  + It serves as a tool that allows Web servers and Web browsers to communicate in such a way that Web servers are able to understand requests given by the Web browser.
  + It is base on a Client-Server Structure which means that interaction between the Web browser and the Web server is initiated by having the Web browser send requests.
  + In additional, it is a set of standards in formatting and transmitting resources such as documents,videos and graphic within the World Wide Web
* **HTTP Fundamentals**
  + HTTP is a stateless protocol which means that the server does not need to keep the information about the client’s request or session.
  + Based on the **Client-Server Architecture/Structure**
    - Client (a.k.a User Agent)
      * A tool such as the Web browser which is used by the users to perform certain tasks in the Web.
    - Web Server
      * Serves web resources such as documents as requested by the client.
  + Uses the **Request-Response** Protocol
    - A protocol in which the client first sends a request which is then sent to the server and the the server sends back a response.
  + HTTP is media-dependent which means that it allows transmission of information in various data types as long as both the Web Server and the Client understands the data type of the resource.
    - Both Client and Server uses MIME Types to identify the data type of the information being sent.
* **HTTP Features and Functionalities**
  + Cache Control
    - HTTP controls caching in such a way that it allows servers to indicate or specify specific resources to be cached at the same time, the Client is allowed to choose the actions to be done on the resources that are cached.
  + Content Media Type specification
    - Uses MIME extension to specify the media types allowed.
  + Language and Character set specification
  + Content/Transfer Codings
    - Allows servers and clients to specify how resources are transmitted
      * For example, a resource can be sent/transmitted in compressed or uncompressed form
  + Content Negotiation
    - Allows clients to negotiate with the server in such a way that it specifies the preferred format of the resource being sent.
      * For example, a client indicates that its preferred format for the documentation is pdf.
  + Request pipelining/multiplexing
    - Multiplexing means concurrency
  + Authentication and Authorization
    - It allows servers to specify the authentication and authorization in accessing its resources.
* **HTTP Version History**
  + HTTP 0.9 (1991)
    - It uses the GET method only
      * **GET** - HTTP method used to retrieve aspecified Web resource
    - It does not allow **persistent connections** which means that it can only establish one TCP connection
    - Server responses only comes in HTML form.
    - Clients can only send a single ASCII form of string
    - Client requests are terminated using CRLF
      * CRLF
        + Carriage Return Line Feedback
        + Used to indicate the termination of a line
  + HTTP 1.0
    - RFC 1945, May 1996
    - Improved HTTP 0.9 such that more methods are added:
      * **POST** - HTTP method used to allow clients to send an information
      * **HEAD** - Identical to the GET method but does not display the content of the response’s body. It only displays the Headers used in th response.
    - This protocol can already accommodate extensions
  + HTTP 1.1
    - RFC 2068, Jan 1997
    - RFC 2616, Jun 1999
    - RFC 7230-7235, Jun 2014
    - HTTP 1.1 is the protocol which is used up until the present
    - More methods are added:
      * **PUT** - Allows clients to replace the contents of the website with the data sent/uploaded by the client.
      * **DELETE** - Allows clients to remove the representation of all the contents.
      * **CONNECT** - Creates connection to the server identified by the given URI.
      * **OPTIONS** - Allows the identification of the HTTP methods allowed by the server
      * **TRACE -** HTTP method used to echo the contents of the Client’s response
    - HTTP 1.1 can already accommodate persistent connections such that it allows connections to be reused
    - HTTP 1.1 already supports data compression wherein content of the message body can already be compressed to enhance speed
    - Pipelining is already supported such that it lowers the transmission delay of the communication between the server and the client.
    - Added more mechanisms regarding Cache Control
  + HTTP 2.0
    - RFC 7540, May 2015
    - It is an enhanced version of HTTP 1.1
      * Parallel requests can already be handled which makes things concurrent
      * Can already accommodate Header compression
    - Added accommodation to server push
      * **Server Push** - Allows the server to automatically send information even if the web browser has not yet sent its request.
* **HTTP Messages**
  + **Request Messages**
    - Elements:
      * HTTP method - this defines the action the client wants to happen.
      * Protocol Version
      * Header - includes additional information regarding the client.
      * Message Body
        + “Payload” - contains the content the client wants to upload or send to the server. (This is commonly used with the use of POST method)
  + **Response Messages**
    - Elements:
      * HTTP protocol version
      * Status code - code that indicates the status of the client’s request
      * Status message - the description of the Status Code
      * Header
      * Message Body
* **HTTP Methods** 
  + **Request Methods (Standard Methods)**
    - GET - Retrieves data from the server
    - HEAD - Similar to the GET method’s function but it does not display the content of the Message Body.
    - POST
      * Creates/Adds new resources to the collection of resources in the server
      * The query string given by the client is included in the Message Body
    - PUT - Creates or replaces the state of target source/origin server
    - DELETE - removes the resource sent by the server
    - OPTIONS - Enables clients to identify the restricted methods or the methods that are only allowed for the resource
    - TRACE
      * Server echoes back the Client’s request
      * It also includes header information and length of the message body
    - CONNECT - Method used to establish tunnel connection between the user and the client to secure connection through a proxy which can then be secured using TLS.
  + **Method Properties**
    - Safe Method
      * Indicates no changes are done in the state of the server
      * GET, HEAD, OPTIONS, TRACE
    - Idempotent Methods
      * No matter how many times requests are repeated, the server’s state will still be the same
      * GET, HEAD, OPTIONS, TRACE, PUT, DELETE
    - Cacheable Methods
      * Allows storing of response to the methods for future use.
      * GET, HEAD, POST
* **HTTP Status Codes**
  + The first digit in a status code refers to the category of which the status code belongs.
  + Categories:
    - Informational (1xx)
      * 100 - Continue
      * 101 - Switching Protocol
    - Success (2xx)
      * 200 - Ok
      * 201 - Created
      * 202 - Accepted
      * 203 - Non-Authoritative Information
      * 204 - No Content
      * 205 - Reset Content
      * 206 - Partial Content
    - Redirection (3xx)
      * 300 - Multiple Choices
      * 301 - Moved Permanently
      * 302 - Found
      * 303 - See Other
      * 304 - Not Modified
      * 305 - Use Proxy
      * 307 - Temporary Redirect
    - Client Error (4xx)
      * 401 - Unauthorized
      * 402 - Payment Required
      * 403 - Forbidden
      * 404 - Not Found
      * 405 - Method Not Allowed
      * 406 - Not Acceptable
      * 407 - Proxy Authentication Required
      * 408 -Request Timeout
      * 409 - Conflict
      * 410 - Gone
      * 411 - Length Required
      * 412 - Precondition Failed
      * 413 - Request Entity Too Large
      * 414 - Request-URI Too Long
      * 415 - Unsupported Media Type
      * 416 - Requested Range Not Satisfiable
      * 417 - Expectation Failed
    - Server Error (5xx)
      * 500 Internal Server Error
      * 501 - Not Implemented
      * 502 - Bad Gateway
      * 503 - Service Unavailable
      * 504 - Gateway Timeout
      * 505 - HTTP Version Not Supported
* **HTTP Headers**
  + Types of Message Headers:
    - General
      * Headers are applied to both the Request and the Response
      * Headers:
        + Cache-Control
        + Connection
        + Date
        + Pragma
        + Trailer
        + Transfer-Encoding
    - Request
      * contains information of the requested resource or information about the client
      * Headers:
        + Accept
        + Authorization
        + Expect
        + From
        + Rane
        + User-Agent
        + Accept-Language
        + Accept-Charset
    - Response
      * contains information of the response or information about the server
      * Headers:
        + E-Tag

It is only present if there is caching

* + - * + Accept-Range
        + WWW-Authenticate
    - Entity
      * Contains information regarding the message body such as its content-length or the data types such as MIME Types the content used.
      * Heades:
        + Allow
        + Content-Encoding
        + Content-Language
* **HTTP Extensions**
  + Allows addition of new request methods, message headers, status codes.
  + WebDAV (RFC 4918)
    - Provision that allows method used for authoring the website remotely
  + Examples:
    - Request Methods
      * PROPFIND, MKCOL, MOVE, LOCK, UNLOCK
    - Message HEaders
      * DAV, DEPTH, DESTINATION, IF, OVERWRITE, TIMEOUT
    - Status Codes
      * 207 - Multi-status
      * 422 - Unprocessable Entity
      * 423 - Locked
      * 424 - Failed Dependency
      * 507 - Insufficient Storage

**HYPERTEXT MARKUP LANGUAGE**