Web Development (WD) (3151606)

Unit-01

Introduction to WEB

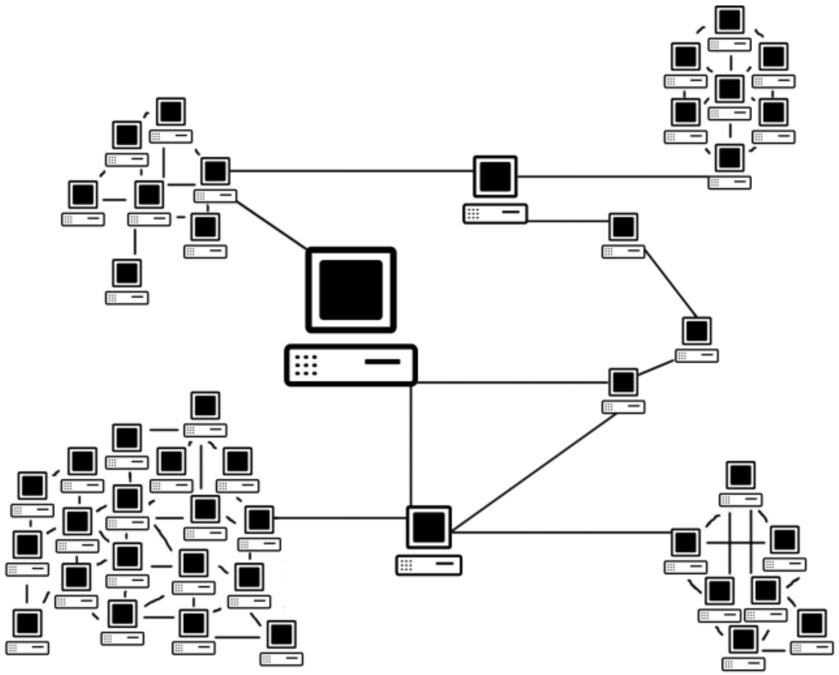
Prof. Mayur Prajapati
Computer Engineering Department
SVBIT, Gandhinagar

Outline

- ✓ Introduction to WEB
  - ✓ What is Internet?
  - ✓ What is WWW?
- ✓ How the WEB works?
- ✓ HTTP Protocol
- ✓ Client Server architecture
- ✓ Web Servers
- ✓ Installing Web Server
- ✓ Configuring Web Server
- ✓ Web Security
- ✓ CORS
- ✓ SEO

What is Internet?

- The Internet is a massive **network of networks**, a networking infrastructure.
- It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet.
- Information that travels over the Internet uses many different set of rules which are known as **protocols**.



What is WWW?

- WWW is stands for World Wide Web.

- The World Wide Web (WWW) is a global information medium which users can read and write via computer connected to the internet.

- It is a collection of websites or web pages stored in web servers and connected to local computers through the internet.

- These websites contain text pages, digital images, audios, videos, etc. Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc.

- The WWW, along with internet, enables the retrieval and display of text and media to your device.

What is WWW?

- A technical definition of the WWW is – All the resources and users on the Internet that are using HTTP.

- HTTP stands for **H**yper**t**ext **T**ransfer **P**rotocol which is a text-based request-response protocol.

- HTTP is an application layer protocol that allows web-based applications to communicate and exchange the data.

- HTTP is a TCP/IP based protocol, so it also is a connectionless and stateless protocol.
  - After making the request, the client disconnects from the server, then when the response is ready the server re-establish the connection again and deliver the response.

- HTTP is the protocol being used to transfer hypertext documents that makes the World Wide Web possible.

What is WWW?

- The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP.

- These links are electronic connections that link related pieces of information so that users can access the desired information quickly.

- Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

What is WWW?

- A web page is given an online address called a Uniform Resource Locator (URL).
- A particular collection of web pages that belong to a specific URL is called a website, e.g., *www.facebook.com*, *www.google.com*, etc.
- So, the World Wide Web is like a huge electronic book whose pages are stored on multiple servers across the world.
- Small websites store all of their WebPages on a single server, but big websites or organizations place their WebPages on different servers in different countries so that when users of a country search their site they could get the information quickly from the nearest server.

What is WWW?

- So, the web provides a communication platform for users to retrieve and exchange information over the internet.

- Unlike a book, where we move from one page to another in a sequence, on World Wide Web we follow a web of hypertext links to visit a web page and from that web page to move to other web pages.

- You need a browser, which is installed on your computer, to access the Web.

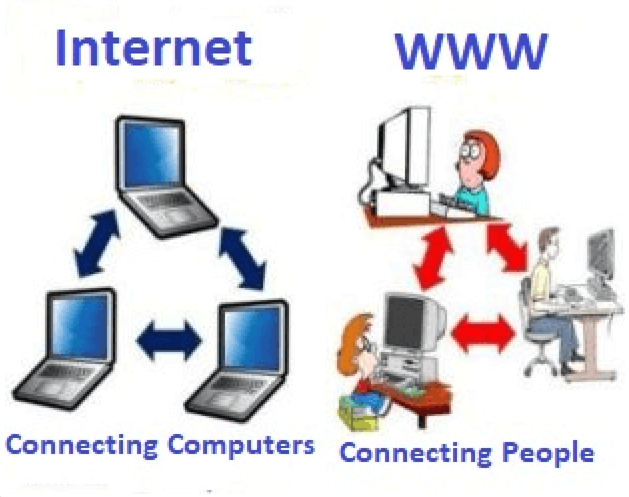
Difference between Internet and WWW:

- Internet is entirely different from WWW.

- It is a worldwide network of devices like computers, laptops, tablets, etc.

- It enables users to send emails to other users and chat with them online.

- For example, when you send an email or chatting with someone online, you are using the internet.

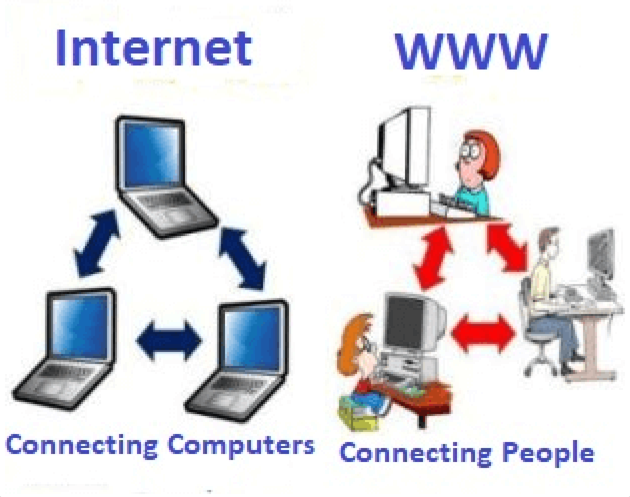


Difference between Internet and WWW:

- But, when you have opened a website like google.com for information, you are using the World Wide Web; a network of servers over the internet.

- You request a webpage from your computer using a browser, and the server renders that page to your browser.

- Your computer is called a client who runs a program (web browser), and asks the other computer (server) for the information it needs.



HTTP

- HTTP stands for Hypertext Transfer Protocol.

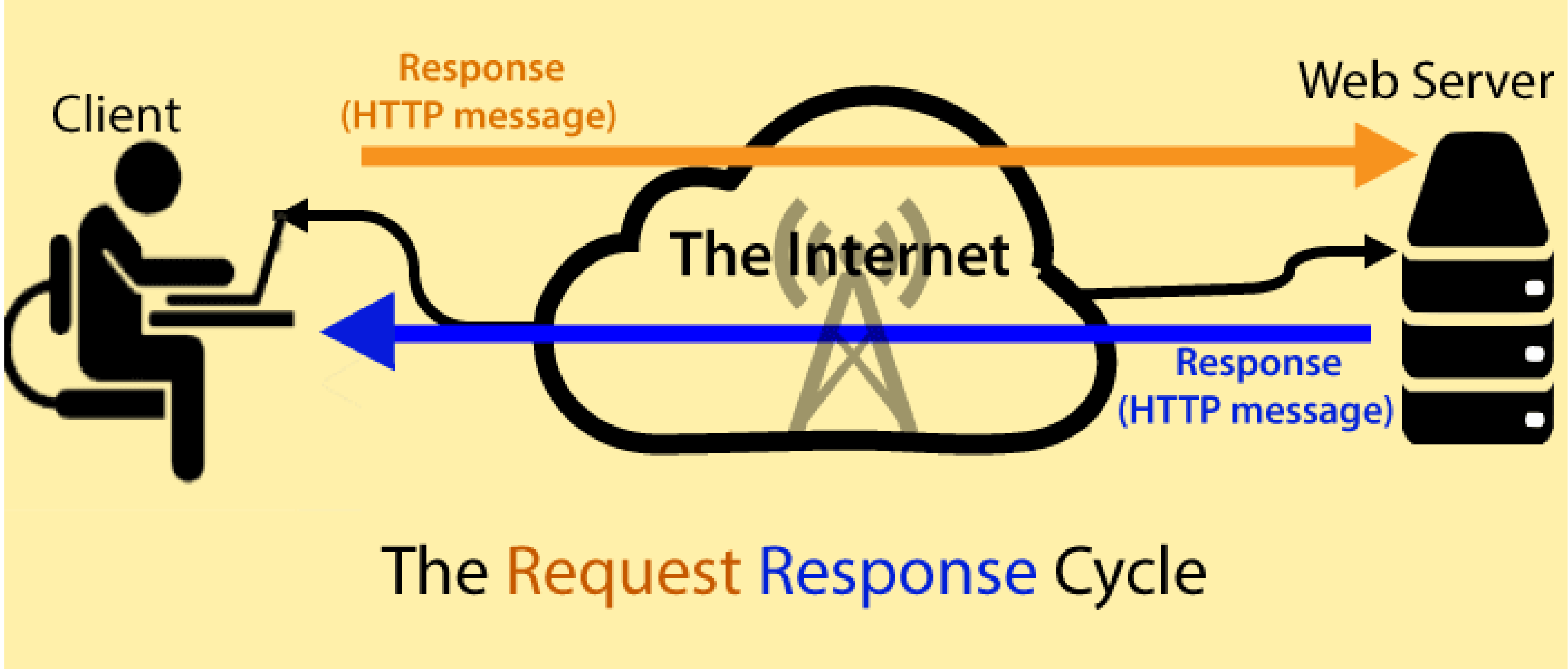
- HTTP is based on the client-server architecture model and a stateless request/response protocol that operates by exchanging messages across a reliable TCP/IP connection.

- An HTTP "client" is a program (Web browser) that establishes a connection to a server for the purpose of sending one or more HTTP request messages.

- An HTTP "server" is a program (generally a web server like Apache Web Server) that accepts connections in order to serve HTTP requests by sending HTTP response messages.

How the Web Works?

- We will have a client on the left side and server on the right side. A user wants to see a website, like www.javatpoint.com/java-tutorial.



How the Web Works?

- The user types the URL of a page using a client program, usually a browser.
- But first, the computer of the user and the web server need to be physically connected.
- That is the job of the internet.
- Using the TCP/IP protocol, it establishes a connection using a combination of cable media or wireless media and does all the necessary work to prepare the environment for the two computers to talk via the HTTP protocol.
- When the connection establishes, the client sends a request called the HTTP message, but because the HTTP is a connectionless protocol, so the client disconnects from the server and waits for the response.

How the Web Works?

- On the other side, the server processes the request, prepare the response, establish the connection again, and send it back the response and again in the form of an HTTP message to the client. Then the two computers completely disconnect.

Features of HTTP

- **HTTP is communication protocol** used between web browser and web server.

- **HTTP is connectionless**: After a request is made, the client disconnects from the server and waits for a response. The server must re-establish the connection after it process the request.

- **HTTP is media independent**: Any type of data can be sent by HTTP as long as both the client and server know how to handle the data content.

- **HTTP is stateless**: This is a direct result of HTTP's being connectionless. The server and client are aware of each other only during a request. Afterwards, each forgets the other.

For this reason neither the client nor the browser can retain information between different request across the web pages.

Features of HTTP

- HTTP has Cache control: It automatically saves recently visited web pages. So, if user visit the same page again, then it can be displayed from cache.

HTTP request

- The HTTP request message consist of following,
  - A request line (e.g. GET /index.php HTTP1.1)
  - Request header fields (e.g. Accept-Language: en)
  - An empty line (CRLF)
  - An optional message body

- The request line and other header fields must end with CRLF (Carriage return, Line Feed) (/r/n)

- A **request line** contains the **method** of request followed by the **resource** we want and at the end protocol **version** used.
  - HTTP Request Methods: GET, POST, PUT, DELETE etc...

HTTP request

- There are many **request header fields** available with HTTP Request, some of are listed below
  - **Accept** : Media type(s) that is/are acceptable for the response. (e.g. Accept: text/html)
  - **Accept-Charset**: Character sets that are acceptable. (e.g. Accept-Charset: utf-8)
  - **Date**: The date and time at which the message was originated (e.g. Date: Tue, 15 Nov 1994 08:12:31 GMT)
  - **Host**: The domain name of the server (e.g. Host: darshan.ac.in)
  - **User-Agent**: details of the browser used (e.g. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36)

HTTP Request Message Structure

- Like most network protocols, HTTP uses the client-server model: An HTTP client opens a connection and sends a request message to an HTTP server; the server then returns a response message, usually containing the resource that was requested.

- After delivering the response, the server closes the connection.

- The format of the request and response messages are similar and will have following structure:
- Start line
- Header lines
- A blank line
- An optional message body like file, query data or query output

HTTP Request Message Structure

- Start Line:
- It consists three parts separated by single space.
- Request Method
- Request- URI
- HTTP Version

- The Request Method indicates the method to be performed on the resource identified by the given Request-URI.
- The method is case-sensitive and should always be mentioned in uppercase.

HTTP Request Message Structure

- The Request-URI is a Uniform Resource Identifier and identifies the resource upon which to apply the request.
- The URI is combination of URL (Uniform Resource Locator) and URN (Uniform Resource Name).
- The web address denotes URL and specific name of place or a person or a item denotes URN.

- Every URI consists two parts:
- The part before colon denotes scheme
- The part after colon depend upon scheme
- The URI are case sensitive , but generally written in lowercase.

- HTTP Version:
- Official version of HTTP is HTTP 1.1.

HTTP Request (Example)

GET /index.php HTTP/1.1	→	index.php is requested from server using GET method of HTTP version 1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36	→	Version 5.0 of mozila browser is used on windows 10 (64 bit) while requesting the page from the server.
Host: www.bapugkv.ac.in	→	Host of the requested page is www.bapugkv.ac.in
Accept-Language: en-us	→	Client accepts US English locale while receiving the response from the server.
Accept: text/html	→	Client accepts text file containing the HTML in it while receiving the response from the server.

HTTP Response

- After receiving and interpreting a HTTP request message, a server responds with an HTTP response message.
- The HTTP response message consist of following,
  - Status-Line (format= HTTP-Version SP Status-Code SP Reason-Phrase CRLF)
  - *((general-header | response-header | entity-header) CRLF)
  - An empty line (CRLF)
  - An optional message body
- Status-Line consist of
  - HTTP-Version, which can be HTTP/1.1
  - Status-Code is a 3 digit code which is in below format
    - 1xx: **Informational** - Request received, continuing process
    - 2xx: **Success** - The action was successfully received, understood, and accepted
    - 3xx: **Redirection** - Further action must be taken in order to complete the request
    - 4xx: **Client Error** - The request contains bad syntax or cannot be fulfilled
    - 5xx: **Server Error** - The server failed to fulfill an apparently valid request
  - Reason-Phase is a textual representation of the status code in human readable format.

HTTP Status Codes with reason phrase

"100": Continue

"101": Switching Protocols

"200": OK

"201": Created

"202": Accepted

"203": Non-Authoritative Information

"204": No Content

"205": Reset Content

"206": Partial Content

"300": Multiple Choices

"301": Moved Permanently

"302": Found

"303": See Other

"304": Not Modified

"305": Use Proxy

"307": Temporary Redirect

"400": Bad Request

"401": Unauthorized

"402": Payment Required

"403": Forbidden

"404": Not Found

"405": Method Not Allowed

"406": Not Acceptable

"407": Proxy Authentication Required

"408": Request Time-out

"409": Conflict

"410": Gone

"411": Length Required

"412": Precondition Failed

"413": Request Entity Too Large

"414": Request-URI Too Large

"415": Unsupported Media Type

"416": Requested range not satisfiable

"417": Expectation Failed

"500": Internal Server Error

"501": Not Implemented

"502": Bad Gateway

"503": Service Unavailable

"504": Gateway Time-out

"505": HTTP Version not supported

Refer:
https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html for more details on HTTP Status Code

HTTP Response (Example)

HTTP/1.1 200 OK	→	Response is 200 status code with OK message using HTTP1.1
Date: Mon, 27 Jul 2009 12:28:53 GMT	→	Response Date & Time
Server: Apache/2.2.14 (Win32)	→	Webserver used by server is Apache and version is 2.2.14 built for 32bit OS
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT	→	Last modified at Date & Time.
Content-Length: 88	→	Content size of the response is 88 bytes
Content-Type: text/html	→	Content type of the response is text file containing HTML

Server

- A server commonly refers to a computer program that receives and responds to requests made over a network.

- It receives the request for a web document from the client and sends the requested information to the client computer on the Internet.

- A device can be both a client and a server at the same time, as an individual system has the ability to provide resources and use them from another system in one go.
- There are different types of servers, including mail servers, virtual servers, and web servers.
- The primary function of a web server is to store, process and deliver web pages to clients.
- Commonly used web servers,
  - For PHP: Apache
  - For ASP: IIS
  - For JSP: Tomcat, Glassfish

Server

- The primary function of a web server is to store, process and deliver web pages to clients.

- a web server may run Microsoft IIS or Apache HTTP Server, which offers users access to the information from web pages or websites over the internet.

- A mail server is able to run a program like iMail or Exim that provides services of SMTP (Simple Mail Transfer Protocol) for sending and receiving email.

- Commonly used web servers,
  - For PHP: Apache
  - For ASP: IIS
  - For JSP: Tomcat, Glassfish

Type of Server

- There are many types of servers, which are as follows:
- Webserver
- Application server
- Blade server
- Cloud server
- Database server
- Dedicated server
- Proxy server
- File server
- Mail server
- Standalone server
- Domain name service

Web Server

- A web server offers web pages or other content to the web browser by loading the information from a disc and transfer files by using a network to the user's web browser.

- It is used by a computer or collection of computers to provide content to several users over the internet.

- This exchange was done with the help of HTTP communicating between the browser and the server.

Web Browser

- A browser is a software program that is used to explore, retrieve, and display the information available on the World Wide Web.

- This information may be in the form of pictures, web pages, videos, and other files that all are connected via hyperlinks and categorized with the help of URLs (Uniform Resource Identifiers). For example, you are viewing this page by using a browser.

- A browser is a client program as it runs on a user computer or mobile device and contacts the webserver for the information requested by the user.

- The web server sends the data back to the browser that displays the results on internet supported devices. On behalf of the users, the browser sends requests to web servers all over the internet by using HTTP (Hypertext Transfer Protocol). A browser requires a smartphone, computer, or tablet and internet to work.

Web Browser

- A browser is a software program that is used to explore, retrieve, and display the information available on the World Wide Web.

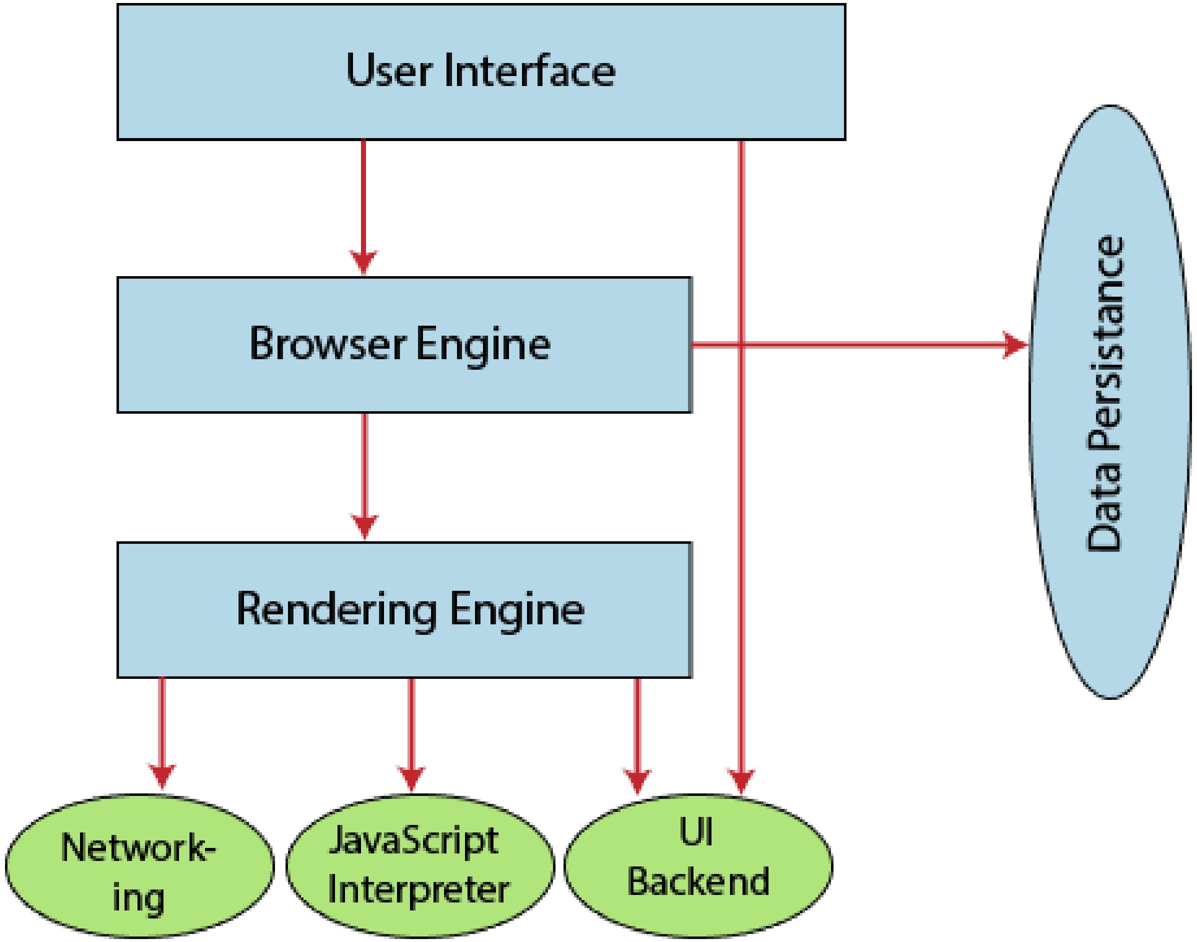
- This information may be in the form of pictures, web pages, videos, and other files that all are connected via hyperlinks and categorized with the help of URLs (Uniform Resource Identifiers). For example, you are viewing this page by using a browser.

- A browser is a client program as it runs on a user computer or mobile device and contacts the webserver for the information requested by the user.

- The web server sends the data back to the browser that displays the results on internet supported devices. On behalf of the users, the browser sends requests to web servers all over the internet by using HTTP (Hypertext Transfer Protocol). A browser requires a smartphone, computer, or tablet and internet to work.

Architecture of Web Browser

- The Architecture of a web browser is shown in the below figure:



Architecture of Web Browser

- **User Interface:** The user interface is an area where the user can use several options like address bar, back and forward button, menu, bookmarking, and many other options to interact with the browser.

- **Browser Engine:** It connects the UI (User Interface) and the rendering engine as a bridge.
- It queries and manipulates the rendering engine based on inputs from several user interfaces.

- **Rendering Engine:** It is responsible for displaying the requested content on the browser screen. It translates the HTML, XML files, and images, which are formatted by using the CSS.
- It generates the layout of the content and displays it on the browser screen.

Architecture of Web Browser

- **Networking:** It retrieves the URLs by using internet protocols like HTTP or FTP.
- It is responsible for maintaining all aspects of Internet communication and security.
- Furthermore, it may be used to cache a retrieved document to reduce network traffic.

- **JavaScript Interpreter:** As the name suggests, JavaScript Interpreter translates and executes the JavaScript code, which is included in a website.
- The translated results are sent to the rendering engine to display results on the device screen.

- **UI Backend:** It is used to draw basic combo boxes and Windows (widgets).
- It specifies a generic interface, which is not platform-specific.

Architecture of Web Browser

- **Data Storage:** The data storage is a persistence layer that is used by the browser to store all sorts of information locally, like cookies.

- A browser also supports different storage mechanisms such as IndexedDB, WebSQL, localStorage, and FileSystem.

- It is a database stored on the local drive of your computer where the browser is installed. It handles user data like cache, bookmarks, cookies, and preferences.

How does a web browser work?

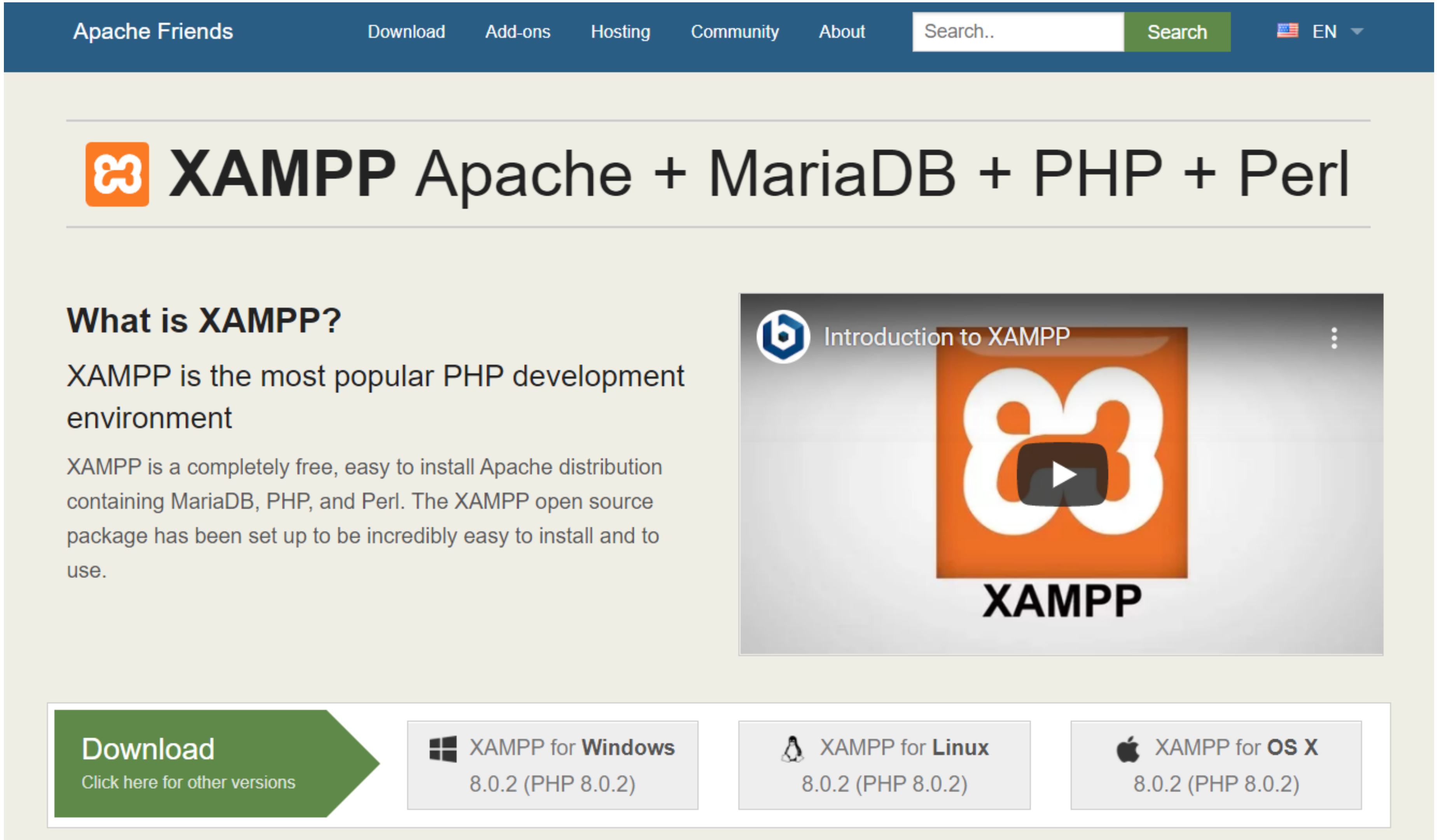
- When a user enters something (like javatpoint.com) in the browser. This request goes to a domain name server.
- The browser sends the user request to the server using an IP address, which is described by the domain name server.
- The domain name server sends an IP address to the web server that hosts the website.
- The server sends the information back to the IP address, which is defined by the browser at the time of the request. The requested page may include links to other files on the same server, like images, for which the browser also requests the server.
- The browser gathers all the information requested by the user, and displays on your device screen in the form of web pages.

Apache web server

- The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996.
- We can download and install apache web server separately from http://httpd.apache.org/download.cgi
- We can also download Apache web server with bundle like XAMPP, WAMP or LAMP.
  - XAMPP is the most popular PHP development environment consist of Apache, MariaDB, PHP, Perl and many other packages.
  - WAMP is popular PHP development environment for Windows OS.
  - LAMP is popular PHP development environment for Linux based OS.
- We are going to use XAMPP bundle as it has installable versions of Windows as well as Linux.
- We can download XAMPP and install XAMPP from https://www.apachefriends.org/index.html
- In the next few slides we are going to see how to install and configure the XAMPP.

Installing XAMPP

- Step 01: Download XAMPP installable from https://www.apachefriends.org/index.html



Apache Friends Download Add-ons Hosting Community About Search.. Search EN

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

Download
Click here for other versions

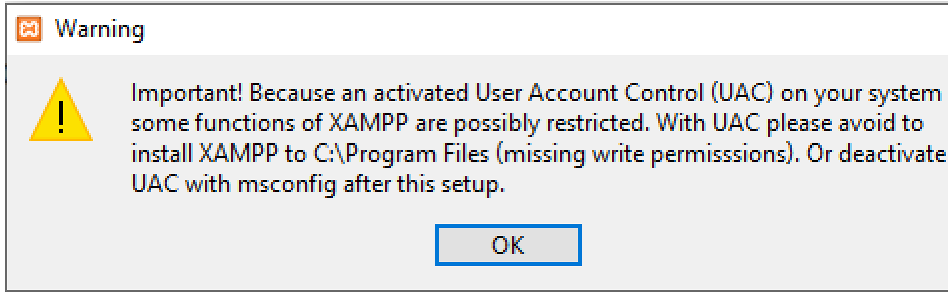
XAMPP for **Windows**
8.0.2 (PHP 8.0.2)

XAMPP for **Linux**
8.0.2 (PHP 8.0.2)

XAMPP for **OS X**
8.0.2 (PHP 8.0.2)

Installing XAMPP (Cont.)

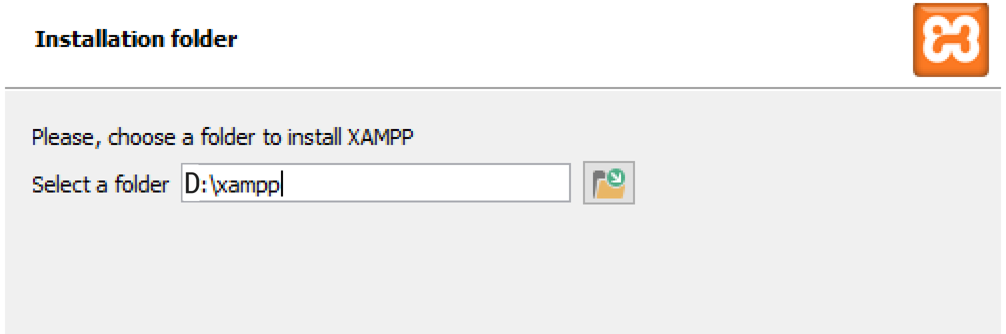
- Step 02: start the installation by double clicking on the downloaded package
- Note: you might get this warning before installation begins, which suggest that UAC policy may restrict XAMPP to perform some functions if we install it in C drive.



Warning

Important! Because an activated User Account Control (UAC) on your system some functions of XAMPP are possibly restricted. With UAC please avoid to install XAMPP to C:\Program Files (missing write permisssions). Or deactivate UAC with msconfig after this setup.

- Press OK in this warning, as we are not going to install XAMPP on C drive.
- After some screen you will get below screen, change the destination to other then C Drive.



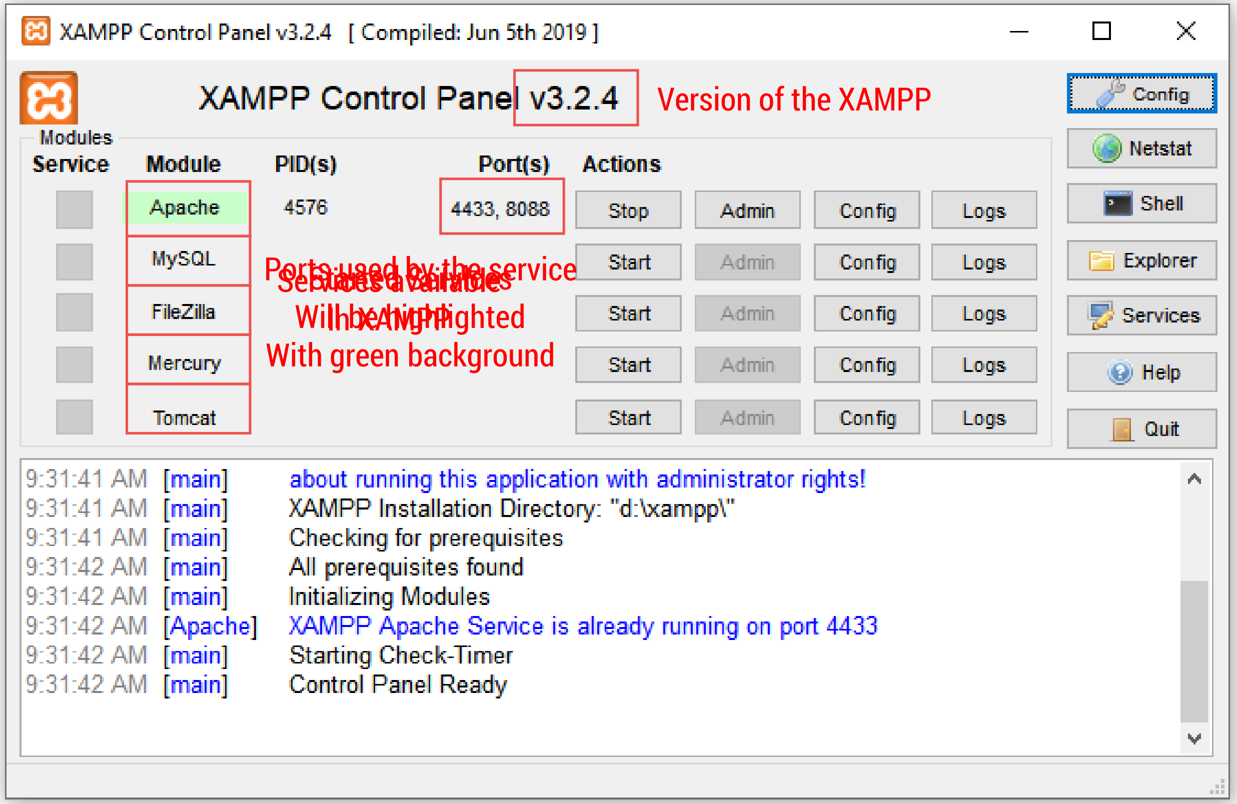
Installation folder

Please, choose a folder to install XAMPP

Select a folder D:\xampp

- After changing the destination of installation, just follow the default installation process.

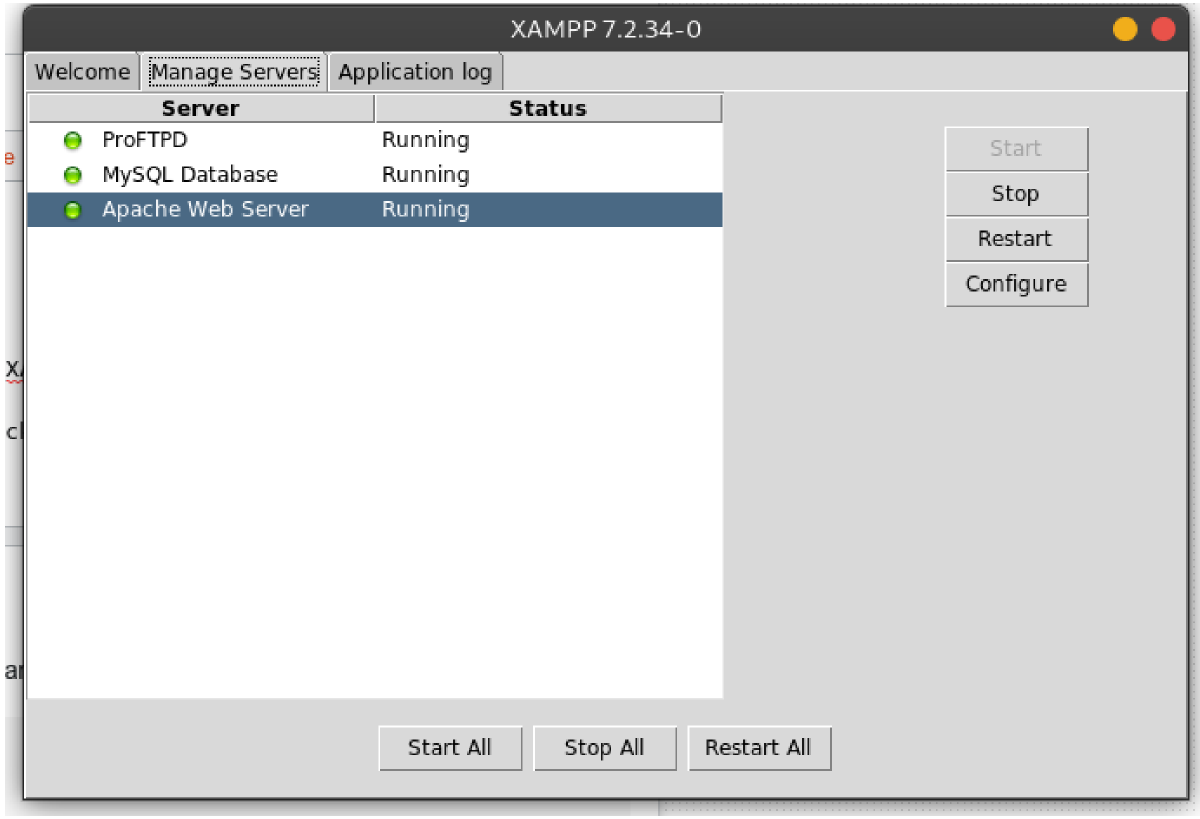
XAMPP Control Panel



Service	Module	PID(s)	Port(s)	Actions			
□	Apache	4576	4433, 8088	Stop	Admin	Config	Logs
□	MySQL			Start	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
9:31:41 AM  [main]     about running this application with administrator rights!
9:31:41 AM  [main]     XAMPP Installation Directory: "d:\xampp\"
9:31:41 AM  [main]     Checking for prerequisites
9:31:42 AM  [main]     All prerequisites found
9:31:42 AM  [main]     Initializing Modules
9:31:42 AM  [Apache]   XAMPP Apache Service is already running on port 4433
9:31:42 AM  [main]     Starting Check-Timer
9:31:42 AM  [main]     Control Panel Ready
```

XAMPP Control Panel (Windows)



Server	Status
● ProFTPD	Running
● MySQL Database	Running
● Apache Web Server	Running

XAMPP Control Panel (Linux)

Configuring Server (Apache)

- Sometimes we need to change the server configuration in some specific conditions like,
  - Port is occupied
  - To add/load modules
  - To change max upload size
  - To change root directory
  - Etc…
- We can configure Apache server using **httpd.conf** file located at **xampp/apache/conf** folder.
- We can configure some parameters of the server using **php.ini** file located at **xampp/php** folder.
- Important settings in **httpd.conf** file are,
  - Changing **port number** for apache : to change the port number for apache go to httpd.conf file and change Listen:80 to any other available port number, we can also change port for https using httpd-ssl.conf file.
  - **Loading/Unloading Modules** in apache : to load/unload modules in apache we can go to httpd.conf and remove/add the comment code(#) from the module we want to load/unload.
  - **Changing Root Directory**: to change root directory we can change **DocumentRoot** and **<Directory>** tag to point new directory.

Configuring Server (Apache) (Cont.)

- Important settings in **php.ini** file are,
  - **`upload_max_filesize`** setting is for the maximum allowed size for uploaded files in the scripts.
  - **`post_max_size`** setting is for the maximum allowed size of POST data that PHP will accept.
  - **`error_reporting = E_ALL & ~E_NOTICE`** setting has default values as `E_ALL` and `~E_NOTICE` which shows all errors except notices.
  - **`max_execution_time`** = 30, Maximum execution time is set to seconds for any script to limit the time in production servers.
  - **`mysql.default_host = hostname`** setting is done to connect to MySQL with default server if no other server host is mentioned.
  - **`mysql.default_user = username`** setting is done to connect MySQL with default username, if no other name is mentioned.
  - **`mysql.default_password = password`** setting is done to connect MySQL with default password if no other password is mentioned.

Web Security

- Web security is also known as “Cybersecurity”.

- It basically means protecting a website or web application by detecting, preventing and responding to cyber threats.

- It is a system of protection measures and protocols that can protect your website or web application from being hacked or entered by unauthorized personnel.

- This integral division of Information Security is vital to the protection of websites, web applications, and web services.

- Anything that is applied over the Internet should have some form of web security to protect it.

Web Security

- There are a lot of factors that go into web security and web protection.

- Any website or application that is secure is surely backed by different types of checkpoints and techniques for keeping it safe.
- There are different types of technologies available for maintaining the best security standards. Some popular technical solutions for testing, building, and preventing threats include:

- Black box testing tools
- Fuzzing tools
- White box testing tools
- Web application firewalls (WAF)
- Security or vulnerability scanners
- Password cracking tools

Web Security

- Web Security also protects the visitors from the below-mentioned points –

- **Stolen Data**: Cyber-criminals frequently hacks visitor’s data that is stored on a website like email addresses, payment information, and a few other details.

- **Phishing schemes**: This is not just related to email, but through phishing, hackers design a layout that looks exactly like the website to trick the user by compelling them to give their sensitive details.

- **Session hijacking**: Certain cyber attackers can take over a user’s session and compel them to take undesired actions on a site.

Web Security

- Thus, web security is easy to install and it also helps the business people to make their website safe and secure.

- A web application firewall prevents automated attacks that usually target small or lesser-known websites.

- These attacks are born out by malicious bots or malware that automatically scan for vulnerabilities they can misuse, or cause DDoS attacks that slow down or crash your website.

- Thus, Web security is extremely important, especially for websites or web applications that deal with confidential, private, or protected information.
- Security methods are evolving to match the different types of vulnerabilities that come into existence.

Cross-Origin Resource Sharing (CORS)

- Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to tell browsers to give a web application running at one origin, access to selected resources from a different origin.

- A web application executes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, or port) from its own.

- An example of a cross-origin request: the front-end JavaScript code served from https://domain-a.com uses XMLHttpRequest to make a request for https://domain-b.com/data.json.

- For security reasons, browsers restrict cross-origin HTTP requests initiated from scripts. For example, XMLHttpRequest and the Fetch API follow the same-origin policy.

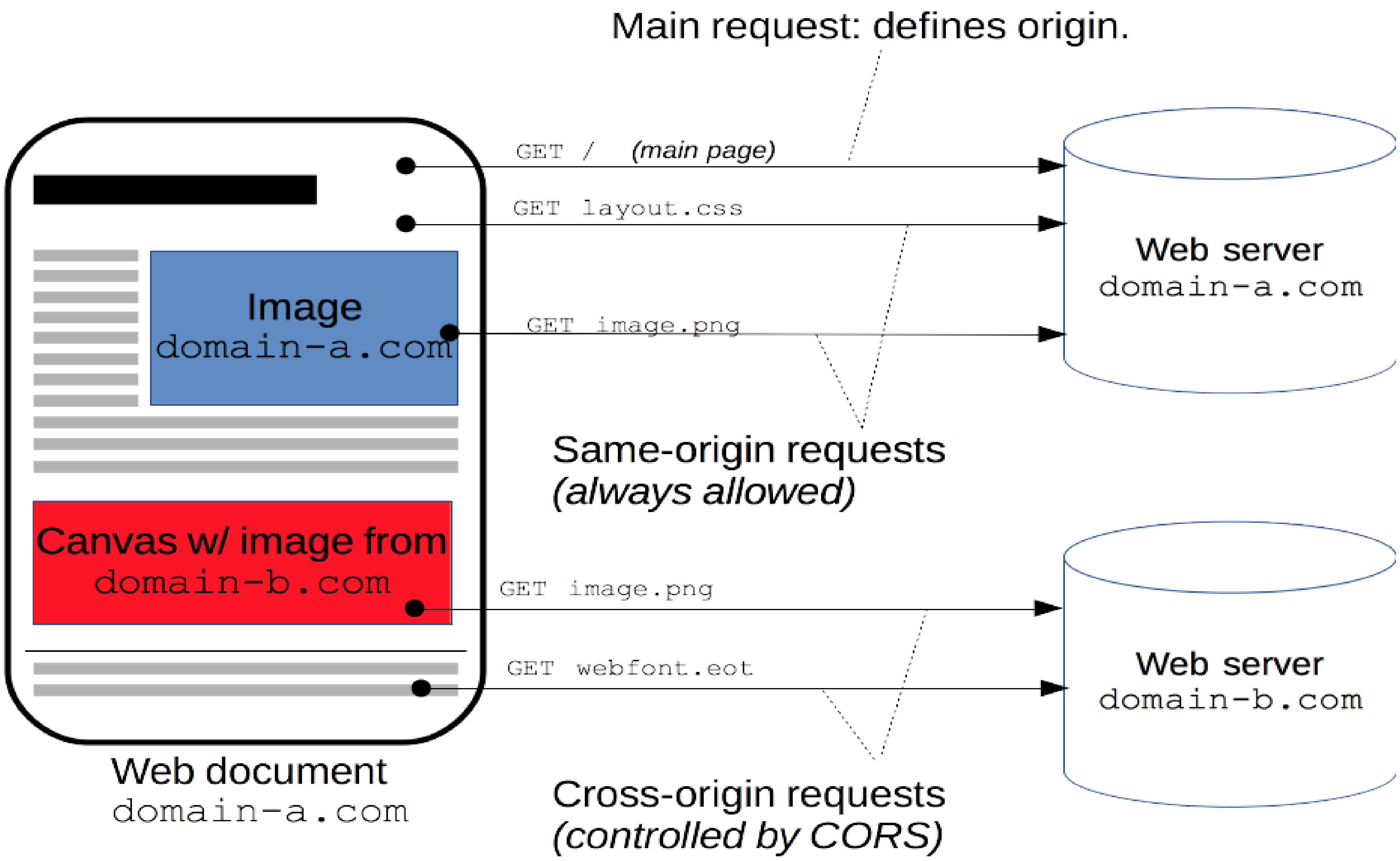
Cross-Origin Resource Sharing (CORS)

- This means that a web application using those APIs can only request resources from the same origin the application was loaded from unless the response from other origins includes the right CORS headers.

- The CORS mechanism supports secure cross-origin requests and data transfers between browsers and servers.

- Modern browsers use CORS in APIs such as XMLHttpRequest or Fetch to mitigate the risks of cross-origin HTTP requests.

Cross-Origin Resource Sharing (CORS)



What requests use CORS?

- This cross-origin sharing standard can enable cross-site HTTP requests for:
- Invocations of the XMLHttpRequest or Fetch APIs, as discussed above.
- Web Fonts (for cross-domain font usage in @font-face within CSS), so that servers can deploy TrueType fonts that can only be cross-site loaded and used by web sites that are permitted to do so.
- Images/video frames drawn to a canvas using drawImage().
- CSS Shapes from images.
- WebGL textures.

THANK YOU

Prof. Mayur Prajapati
Computer Engineering Department
SVBIT, Gandhinagar