Getting Started with Basics

Welcome to the web development course!

- You will learn all aspects of front-end web development. Starting from the very basics to the advanced.

- The best part of this journey is you'll get hands-on experience as you'll be doing some projects along the course, which will reinforce and strengthen your skills.

- We'll learn how to structure our webpage, design it and add interactivity to it. HTML, CSS, and JS

This course is divided into two modules:

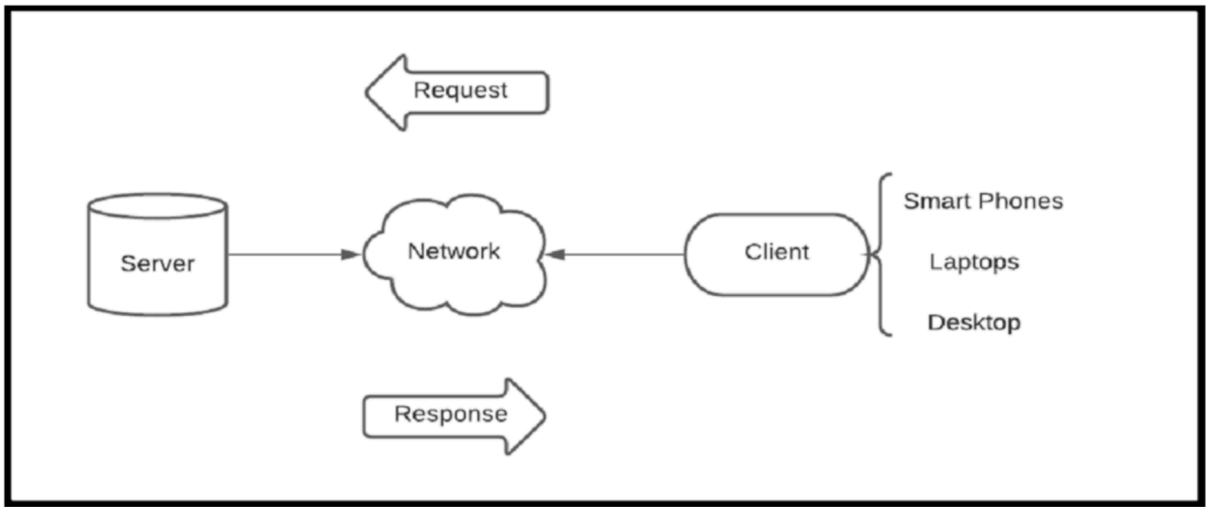
1. **Front-end Web Development** – Here you will understand the principles of user interface design and user experience.
   You'll construct new, appealing, responsive websites with HTML, CSS, JavaScript, jQuery, and Bootstrap. This part of the course focuses on core concepts and lays a strong programming foundation.

2. **Back-end Web Development** – In this section, you'll learn how to build and maintain the technologies that power the administrative side of websites.
   It's a deep dive that will help you understand a web application's operation with databases to evolve simple, static websites into dynamic, database-driven web applications. You'll go through a wide range of topics like NodeJS, event handling, file uploads, social authentication, deployment, etc.

Client-Server Architecture

Like in the outside world, the Client means any person or organization that uses a particular service. Similarly, in the world of computer networks, a client is a host (any computer or a device that is connected to a computer network) that receives some service from certain service providers (Servers).

As the word suggests, a Server is a remote computer that provides some service/information to its clients. So, a client requests some information, and the Server responds by serving it to the Client. Servers store and retrieve the information from the databases. The client can be any device like a mobile phone, laptop, desktop, etc.



Front end and Back end

There are two faces of a website: front-end and back-end. These are the most popular andcrucial topics in web development.

Front-end:

- The front-end is the part of a website with which the users interact directly. It's also called the 'client side' of a website.
- It includes everything the users see: text, colours, styles, images, buttons, menus, etc.
  HTML, CSS, and JavaScript are generally used for front-end development.
  Front-end developers build the front-end part of a website.
  Being a front-end developer, you will have to ensure that your websites look good on all devices for a smooth user experience.

- There are many front-end frameworks and libraries which developers use, like React.js, jQuery, Angular.js, etc.

Backend:

- The back end is the server side of a website.
- A back-end web developer uses it to store and manage data and make sure that the client side of the website works without any issues.
- The website's users or clients cannot see and interact with the back end of a website.The backend part of the software is abstracted from the users.
- Every line of code you write for the backend will be used on the frontend side anyways.
- In simple words, we can say that everything that happens in the background can be credited to the backend.
- Like frontend, there are many backend libraries that developers use like Express, Rails, Django, etc.

Static Vs Dynamic Websites:

- A static website is a website that displays the same content to each user, and it's usually written using simple HTML and CSS.
- Whereas a dynamic website displays different content to different users depending on user data and preferences and provides user interaction. In addition to HTML, making dynamic websites require advanced technologies, frameworks, and databases to store user data.
- Usually, when static websites run on a browser, the content shown is the same for every person accessing the website. An excellent example of a static website would be a simple Blog page.
- A dynamic website on the other hand is more functional. Here the users are required to interact with the website as user interaction plays a big role in

dynamic websites.

- As the website is dynamic, the content shown on the website will vary according to specific users. The information shown on the page will not be the same as Facebook, where the content is relevant and related to the specific user.

What happens when you visit a website?

Every website has its IP address, using which we can access a particular website. But, as humans, are not good at remembering numbers, we use domain names, which a user-friendly way to access a website's IP address. As the user enters the URL of the website, the browser sends a request for the domain name of the website. The DNS (Domain Name Server), which is like a phonebook of the internet, connects domain names with IP addresses and responds with the web server's IP address. Then the browser sends an HTTP request to the web server's IP (which is provided by DNS). The Server sends over the website's necessary files, which are then correctly rendered by the browser and displayed as a website.

What is DNS?

- The Domain Name System (DNS) is the phonebook of the Internet. E.g. when you want to call your friend, search for the friend's name in the phone directory and call them, but in an actual call on their mobile number. Similarly, Domain Name System (DNS) does this same process but for domain names and IP addresses.

- Humans access information online through domain names, like google.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

What happens when you enter google.com in the web browser?

Steps:

- Initial Typing: When you start typing google.com in the browser, the browser will start looking for your browser cache first if the content is fresh and present in the cache display the same.

- URL Parsing: If not, the browser checks if the IP address of the URL is present in the browser cache if not then requests the OS to do a DNS lookup using UDP to get the corresponding IP address of the URL from the DNS server to establish a new TCP connection.

- A new TCP connection is set between the browser and server.

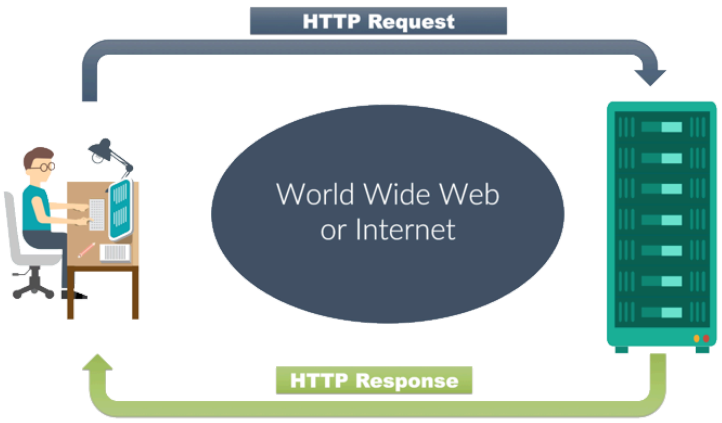
- An HTTP request is sent to the server using the TCP connection.

- The web servers running on the Servers handle the incoming HTTP request and send HTTP responses.

- The browser processes the HTTP response sent by the server and may close the TCP connection or reuse the same for future requests.

- If the response data is cacheable then browsers cache the same.

- Browser decodes the response and renders the content.



What is HTTP?

- HTTP is a protocol that allows the fetching of resources, such as HTML documents.

- It is the foundation of any data exchange on the Web, and it is a client-server protocol. Your server will receive requests from the browser that follows HTTP.

- It then responds with an HTTP response that all browsers can parse.

What is a web browser, and how does it work?

The web browser is an application that provides access to the web server, sends a network request to the URL, obtains resources, and interactively represents them.

E.g., Google Chrome, Firefox, Safari, Internet Explorer and Opera.

Conclusion:

In conclusion, embarking on the web development course provides a comprehensive journey through the fundamentals and advanced aspects of front-end and back-end web development. The course structure, divided into two modules, ensures a well-rounded learning experience. In the front-end module, participants dive into user interface design principles, honing their skills in creating visually appealing and responsive websites using HTML, CSS, JavaScript, jQuery, and Bootstrap. The subsequent back-end module delves into the intricacies of web application functionality, covering technologies such as NodeJS, event handling, file uploads, social authentication, deployment, and more. The hands-on approach, involving project work, promises to reinforce and solidify the acquired skills, making this course a valuable resource for individuals looking to build a strong foundation in web development.

Exploring essential concepts like client-server architecture, the distinction between front-end and back-end development, and the contrast between static and dynamic websites, the course sets the stage for a holistic understanding of web development. Participants gain insights into the client-server relationship, understanding how browsers interact with servers through protocols like HTTP and how DNS plays a pivotal role in translating user-friendly domain names into IP addresses. The notes on the client-server architecture highlight the integral role of servers in serving information to clients, shaping the foundation of the internet. Overall, the course provides a comprehensive introduction to the world of web development, demystifying key concepts and empowering learners to navigate the dynamic landscape of creating interactive and engaging web applications.

References:

- https://www.cloudflare.com/en-gb/learning/dns/what-is-dns/
- https://www.cloudflare.com/en-gb/learning/ddos/glossary/hypertext-transfer-protocol-http/
- https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work