1. How internet works?

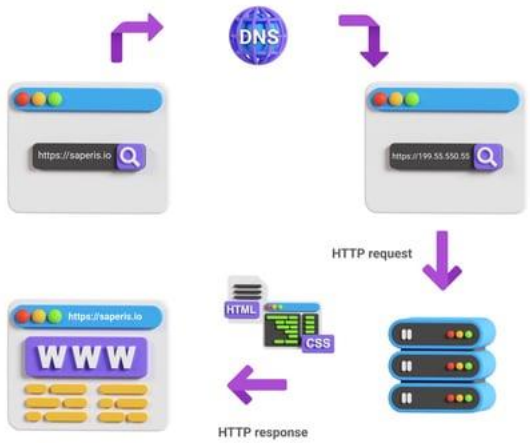
   There are two main concepts that are fundamental to the way the Internet functions: *packets* and *protocols*.

   In networking, a packet is a small segment of a larger message. Each packet contains both data and information about that data. The information about the packet's contents is known as the "header," and it goes at the front of the packet so that the receiving machine knows what to do with the packet. To understand the purpose of a packet header, think of how some consumer products come with assembly instructions.

   Connecting two computers, both of which may use different hardware and run different software, is one of the main challenges that the creators of the Internet had to solve. It requires the use of communications techniques that are understandable by all connected computers, just as two people who grew up in different parts of the world may need to speak a common language to understand each other.

2. How browser works?

   Let’s say you want to access the website saperis.io. That website, just like any other, is hosted on a server. The browser goes to the Domain Name System server, also known as DNS server, to translate the domain name into an IP address to locate the server where the web site is hosted. The browser then sends a request to the server using the Hypertext Transfer Protocol, also known as HTTP, asking it to send a copy of the website. Usually, a website is a collection of HTML, CSS, JavaScript files, images, and more.



The web server answers the browser’s request by using HTTP to send a copy of the files. The HTML file contains all the information the browser needs to be able to correctly put together and display the text, images, and so on.
The browser uses a piece of software called a rendering engine to translate the HTML into text and images. So for example, if in the HTML there is a hyperlink for an image, the browser will use that link, or URL, to access the image and display it on the website. The end result is that you see a complete website in your browser.

3. What is server?

   A Server is a program or a device that provides functionality for called clients which are other programs or devices. This architecture is called the client-server model. A single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities called services. These services include sharing data or resources among multiple clients or performing computations for a client. Multiple clients can be served by a single server, and a single client can use multiple servers.

4. What are the types of server available?

   1. Application Server
   These servers host web apps (computer programs that run inside a web browser) allowing users in the network to run and use them preventing the installation of a copy on their own computers.
   2. Catalog Server
   These servers maintain an index or table of contents of information that can be found across a large distributed network.
   3. Communication Server
   These servers maintain an environment needed for one communication endpoint to find other endpoints and then communicate with them.
   4. Computing Server
   These servers share vast amounts of computing resources which include CPU and random-access memory over a network.
   5. Database Server
   These servers maintain and share any form of database over a network.
   6. Fax Server
   These servers share one or more fax machines over a network which eliminates the hassle of physical access.
   7. File Server
   Shares files and folders, storage space to hold files and folders, or both, over a network.
   8. Game Server
   These servers enable several computers or gaming devices to play multiplayer games.
   9. Mail Server
   These servers make email communication possible in the same way as a post office makes snail mail communication possible.
   10. Print Server
   These servers share one or more printers over a network which eliminates the hassle of physical access.
   11. Proxy Server
   This server acts as an intermediary between a client and a server accepting incoming traffic from the client and sending it to the server.
   12. Web Server
   These servers host web pages. A web server is responsible for making the World Wide Web possible.

5. What is SEO? Importance of SEO?

   SEO stands for “search engine optimization.” In simple terms, SEO means the process of

improving your website to increase its visibility in Google, Microsoft Bing, and other search engines whenever people search for:

- Products you sell.
- Services you provide.
- Information on topics in which you have deep expertise and/or experience.

SEO is a critical marketing channel. First, and foremost: organic search delivers 53% of all website traffic. That's one big reason why the global SEO industry is forecast to reach a staggering $122.11 billion by 2028. SEO drives real business results for brands, businesses and organizations of all sizes. Whenever people want to go somewhere, do something, find information, research or buy a product/service – their journey typically begins with a search. But today, search is incredibly fragmented. Users may search on traditional web search engines (e.g., Google, Microsoft Bing), social platforms (e.g., YouTube, TikTok) or retailer websites (e.g., Amazon). In fact, 61% of U.S. online shoppers start their product search on Amazon, compared to 49% who start on a search engine like Google.

6. What is Accessibility?

   Accessibility is the practice of making your websites usable by as many people as possible. We traditionally think of this as being about people with disabilities, but the practice of making sites accessible also benefits other groups such as those using mobile devices, or those with slow network connections.

7. What is Markup Language?

   A markup language is a set of rules that defines how the layout and presentation of text and images should appear in a digital document. It allows structuring documents, adding formatting, and specifying how different elements should be displayed (or "rendered") on webpages.

8. What is HTML?

   To understand "HTML" from front to back, let's look at each word that makes up the abbreviation: Hypertext: text (often with embeds such as images, too) that is organized in order to connect related items. Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format. Language: a language that a computer system understands and uses to interpret commands. HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive.

9. What is Browser Engine?

The underlying software that turns HTML pages into the Web page the user sees. A browser engine includes the programming interface (API) and the rendering engine, which converts HTML and JavaScript into text and images for the screen and printer. Also called a "layout engine," a browser engine is also used by email programs that support HTML (most do), as well as other applications that render Web content.

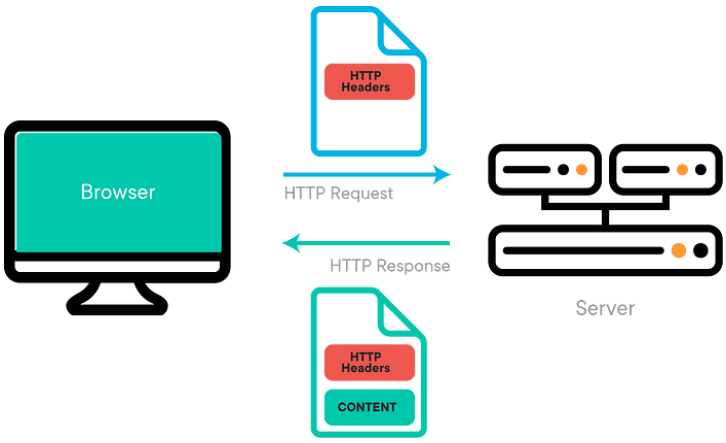
10. What is rendering engine? Share the available rendering engine?

    It's responsible for displaying the requested web resources by parsing the contents. By default it can parse html, xml, and images. It uses different plugins and/or extensions to display other type of data such as flash, PDF, etc.
    There are different rendering engines such as Gecko, WebKit, and Trident. Most widely used rendering engine is WebKit or its variant version. Gecko and WebKit are open source rendering engines while Trident is not. Firefox uses Gecko, Safari uses WebKit, Internet Explorer uses Trident, Chrome and Opera uses Blink, which is a variant of WebKit.

11. What is JavaScript Engine? Share the available JS Engine? Purpose of JS Engine?

    A JavaScript engine is a computer program that executes JavaScript code and converts it into computer understandable language.
    Available JS Engines are: V8, Chakra, Spider Monkey, Javascript Core Webkit.
    Purpose of JS Engine is These engines help to convert our JavaScript program into computer-understandable language.

12. How website works?



    All websites are identified by a unique address, which tells browsers where they are located.

    Simply put, when you enter a web address into your browser, it goes to that specified location and retrieves the web page. The process of retrieving this information is carried out by a web service using technologies such as Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP), which essentially define how information and files are transmitted over the web.

The browser then displays the retrieved web page on your screen using technologies such as HyperText Markup Language (HTML) and Cascading Style Sheet (CSS). They structure and present the web page's information to the visitor by telling the browser exactly where to put what on the screen.

Fortunately, you don't need to know how to write computer code or have in-depth knowledge of these technologies to make a website. In fact, you can turn your website idea into a fully functional site in a matter of minutes using a website builder.

13. What is Data Structure?

    A data structure is a particular way of organising data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks.

14. Explain Tree Data Structure?

    Tree data structure is a specialized data structure to store data in hierarchical manner. It is used to organize and store data in the computer to be used more effectively. It consists of a central node, structural nodes, and sub-nodes, which are connected via edges.

15. What is user agent? Share the list and its purpose?

    The User Agent is a request header that allows a characteristic string that allows network protocol peers to identify the Operating System and Browser of the web-server. Your browser sends the user agent to every website you connect to. There is no conventional way of writing a user agent string as different browsers use different formats and many web browsers load a lot of information onto their user agents. When your browser is connected to a website, a User-Agent field is included in the HTTP header. The data of the header field varies from browser to browser. This information is used to serve different websites to different web browsers and different operating systems.

    Purpose of user agent:
    Enabling users to access and interact with online content and services in a convenient and secure manner.

    List of user agents:
    - Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
    - Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:124.0) Gecko/20100101 Firefox/124.0
    - Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.2420.81
    - Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 OPR/109.0.0.0
    - Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
    - Mozilla/5.0 (Macintosh; Intel Mac OS X 14.4; rv:124.0) Gecko/20100101 Firefox/124.0
    - Mozilla/5.0 (Macintosh; Intel Mac OS X 14_4_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15
    - Mozilla/5.0 (Macintosh; Intel Mac OS X 14_4_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 OPR/109.0.0.0

- Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
- Mozilla/5.0 (X11; Linux i686; rv:124.0) Gecko/20100101 Firefox/124.0

Types of user agents:

- Web browsers: These are the most common type of user agents. Examples include Google Chrome, Mozilla Firefox, and Microsoft Edge.
- Mobile apps: Many mobile apps act as user agents, making requests to servers and displaying the responses to the user.
- Desktop applications: Some desktop applications, such as email clients and chat clients, act as user agents when they communicate with servers.
- Bots and crawlers: These are automated user agents that are often used for tasks such as web indexing[2], site monitoring, and data mining.
- Internet of Things (IoT) devices: These are user agents that represent devices connected to the Internet, such as smart thermostats, security cameras, and home appliances.
- Custom user agents: Some user agents are designed specifically for a particular purpose and are not generally available to the public. For example, a company might develop a user agent to perform automated testing of its website.

16. What is Hypertest?

    HyperTest is a cutting-edge testing tool that has gained prominence in the field of software testing. HyperTest is an API test automation platform that helps teams generate and run integration tests for their microservices without ever writing a single line of code. HyperTest is a no-code test automation tool that regresses all your APIs by auto-generating integration tests using your network traffic, also giving a way to reproduce these failures inside actual user-journeys.

17. What is HTML Tags?

    HTML tags are essential building blocks that define the structure and content of a webpage. HTML tags are composed of an opening tag, content, and a closing tag. The opening tag marks the beginning of an element, and the closing tag marks the end. The content is the information or structure that falls between the opening and closing tags. Here's the basic structure of an HTML tag:
    <tagname> Content... </tagname>

18. What is HTML attributes?

    HTML attributes provide additional information about elements within an HTML document. Every HTML element can have attributes. Attributes are always defined in the start tag. They are specified using a name/value pair format, where the attribute name defines the property, and its value provides specific details, like name="value". These attributes impact content display and interaction on web pages.

19. What is HTML Elements?

    An HTML element is defined by a start tag, some content, and an end tag.

    The HTML element is everything from the start tag to the end tag:

    <tagname>Content goes here...</tagname>

    Examples of some HTML elements:

    <h1>My First Heading</h1>

    My first paragraph.

20. How to convert elements to tree?

    A tree view represents a hierarchical view of information, where each item can have a number of subitems.

    Click on the arrow(s) to open or close the tree branches.

    Step 1) Add HTML

    Step 2) Add CSS

    Step 3) Add JavaScript

21. What is DOCTYPE?

    The HTML document type declaration, also known as DOCTYPE, is the first line of code required in every HTML or XHTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in. This ensures that the web page is parsed the same way by different web browsers.

22. What are the ways we can save html file?

In Notepad:
1. Open Notepad.
2. Write the HTML Code.
3. Save the HTML page: Once the code is written, you can click on Files and then Save or directly press Ctrl + S.

In Chrome:
1. Navigate to the web page, right-click on the page and select Save as...
2. Select or create a new folder to save the file, images, and associated items from the web page.
3. Enter a file name and select Webpage, Complete (*.htm;*html) for the Save as type
4. Click the Save button.

In FireFox:
1. Navigate to the web page and select File, Save Page As... or right-click and select Save Page As...
2. Select or create a new folder to save the file, images, and associated items from the web page.
3. Enter a file name and select Web Page, complete (*.htm;*.html) for the Save as type:
4. Click Save

In Internet Explorer
1. Navigate to the web page and File, Save as...
2. Enter a file name and select Web Page, complete (*.htm;*.html) for the Save as type:
3. Click Save

23. What is charset? Why we need to use this?

The charset attribute specifies the character encoding for the HTML document.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

Syntax: <meta charset="character_set">

24. What is metadata? What is the purpose of it?

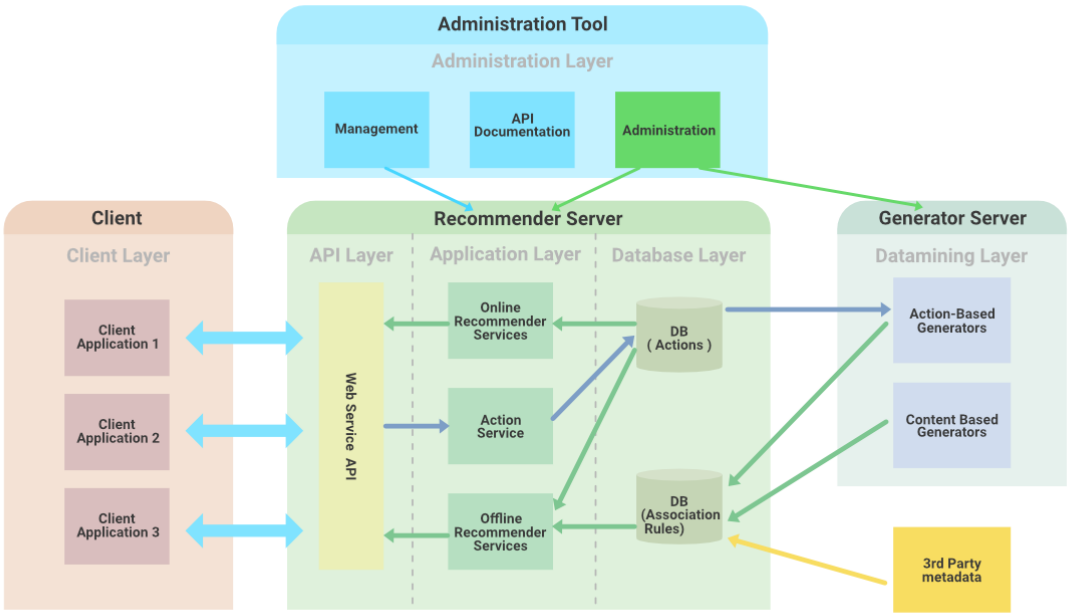
Metadata is frequently described as “data about other data.” Whether detailing the contents of a web page, the technical details of an image, or information about an asset's usage rights, metadata provides additional information that facilitates data management so assets can be located and used more efficiently. It provides context with details such as the source, type, owner, and relationships to other data sets. So, it can help you understand the relevance of a particular data set and guide you on how to use it.

25. Explain Web Architecture?

Web application architecture is a mechanism that gives us a clarification that how the connection is established between the client and the server. It determines how the

components in an application communicate with each other. It doesn't matter what is the size and the complexity level of the application is, they all follow the same principle only the details may differ.

How Does Web Application Architecture Work?



All the web applications run on the client-side and the server-side. When a user makes a request there are mainly two programs run on both sides.

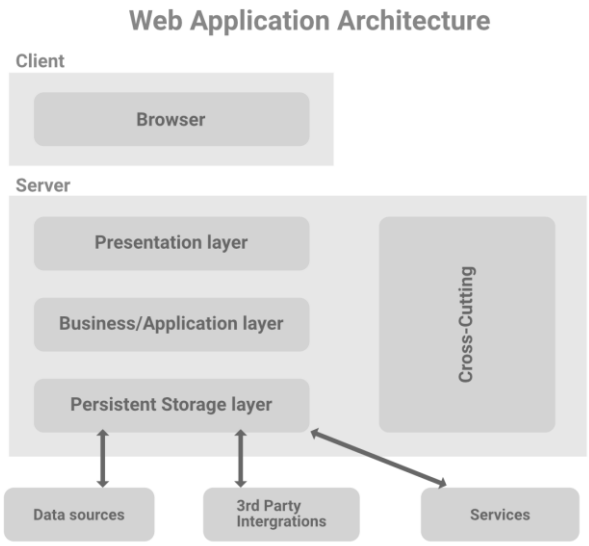
- Code that runs in the browser and works as per the inputs of the user.
- Code in the server which responds to the HTTP requests

While working on the web application, a web developer decides the functions of the code on the server and the functions of the code on the browser. They also define how these two will function in relation to each other. Server-side code can be written using the languages Python, JavaScript, C#, PHP, Ruby on Rails, etc. Any code can have the capability to run on the server if it can respond to HTTP requests. The server-side code is mainly responsible for creating the page which the user has requested. It also stores different types of data such as user profiles, tweets, pages, etc. Server-side code can not be seen by the end-user (except within a rare malfunction)

Client-side languages include the combination of HTML, CSS, and JavaScript. This code is parsed by the browser, and it can be seen as well as edited by the user. Only through HTTP requests, client-side code can communicate with the server. Also, it cannot read files off a server directly.

Web Application Three Tier Architecture Layers

Web application architectural patterns are separated into many different layers or tiers which is called Multi- or Three-Tier Architecture. You can easily replace and upgrade each layer independently.



Presentation Layer: This layer is accessible to the client via a browser and it includes user interface components and UI process components. As we have already discussed that these UI components are built with HTML, CSS, and JavaScript (and its frameworks or library) where each of them plays a different role in building the user interface.

Business Layer: It is also referred to as a Business Logic or Domain Logic or Application Layer. It accepts the user's request from the browser, processes it, and regulates the routes through which the data will be accessed. The whole workflow is encoded in this layer. You can take the example of booking a hotel on a website. A traveler will go through a sequence of events to book the hotel room and the whole workflow will be taken care of by the business logic.

Persistence Layer: It is also referred to as a storage or data access layer. This layer collects all the data calls and provides access to the persistent storage of an application. The business layer is closely attached to the persistence layer, so the logic knows which database to talk to and the process of retrieving data becomes more optimized. A server and a database management system software exist in data storage infrastructure which is used to communicate with the database itself, applications, and user interfaces to retrieve data and parse it. You can store the data in hardware servers or in the cloud.

Some other parts of the web application which is separated from the main layers that exist in the architecture are…

- Cross-cutting code: This part handles communications, operational management, and security. It affects all parts of the system but should never mix with them.
- Third-party integrations: Using third-party APIs we can integrate payment gateways, social logins, GDSs in travel websites, etc.