What is web ?

The Web is **a computer system that links documents and pictures into a database that is stored in computers in many different parts of the world and that people everywhere can use**.

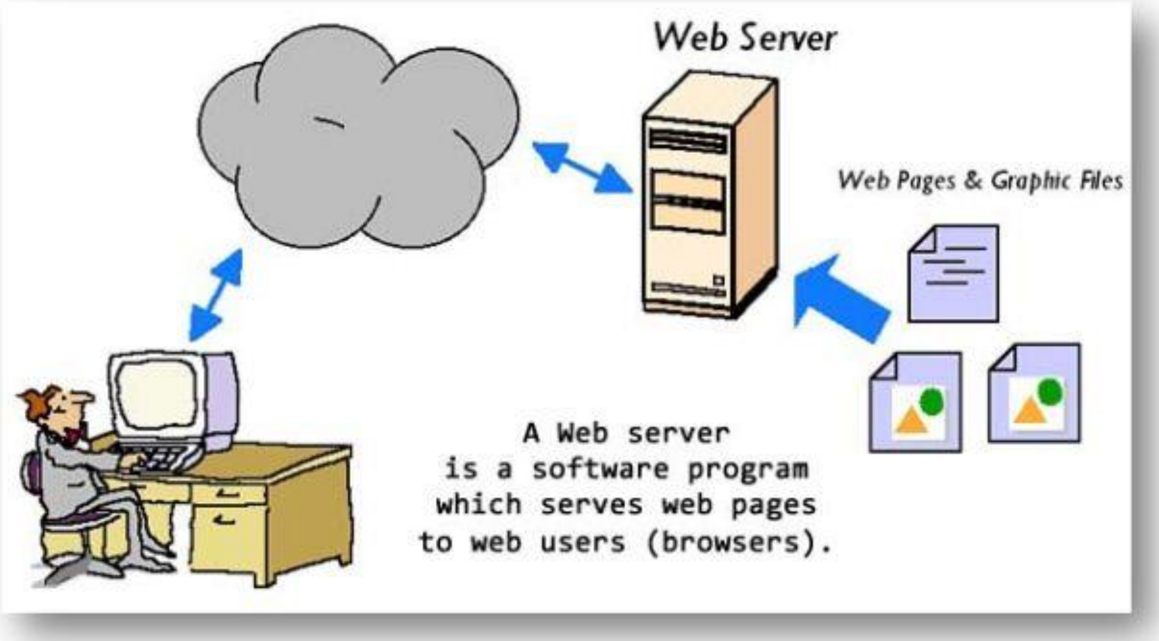
ওয়েব হল একটি কম্পিউটার সিস্টেম যা নথি এবং ছবিগুলিকে একটি ডাটাবেসের সাথে লিঙ্ক করে যা বিশ্বের বিভিন্ন অংশে কম্পিউটারে সংরক্ষিত থাকে এবং যেটি সর্বত্র মানুষ ব্যবহার করতে পারে।

What is web server ?

A web server is a specialized type of computer system that delivers web pages and other content to users over the internet.

একটি ওয়েব সার্ভার হল একটি বিশেষ ধরনের কম্পিউটার সিস্টেম যা ইন্টারনেটের মাধ্যমে ব্যবহারকারীদের কাছে ওয়েব পেজ এবং অন্যান্য সামগ্রী সরবরাহ করে।

When you request a web page by typing a URL into your browser, your browser sends a request to the web server that hosts that page. The web server then processes the request, retrieves the requested content (such as HTML, CSS, images, or videos), and sends it back to your browser for display.



What is a web application ?

A web application (web app) is an application program that is stored on a remote server.

How does a web application work?

1. User Interaction

- **Client-Side**: The user interacts with the web app through a web browser on their device. This involves sending requests by clicking buttons, filling out forms, or performing other actions.

2. Request Handling

- **Client-Side**: The browser sends an HTTP request to the web server. This request might be for a specific page or to perform an action like submitting a form.
- **Server-Side**: The web server receives the request. Depending on the web app's design, it may directly serve static content (like HTML, CSS, and JavaScript files) or pass the request to an application server.

3. Server-Side Processing

- **Application Server**: The application server processes the request. This involves running server-side code written in languages like Python, Ruby, PHP, Java, or JavaScript (Node.js). The server-side code may interact with a database or perform other logic to handle the request.
- **Database Interaction**: If needed, the application server communicates with a database to retrieve, store, or modify data. This is common in web apps that involve user accounts, content management, or other dynamic data.

4. Response Generation

- **Server-Side**: Once the application server has processed the request and possibly interacted with a database, it generates a response. This response is typically in the form of HTML, JSON, or XML.

5. Response Delivery

- **Client-Side**: The web server sends the response back to the web browser. For web apps, this often involves delivering HTML content, CSS for styling, and JavaScript for interactivity.

6. Rendering and Interaction

- **Client-Side**: The browser renders the HTML, applies CSS for styling, and executes JavaScript to provide interactive features. JavaScript can also handle asynchronous operations (e.g., using AJAX or Fetch API) to update parts of the web app without reloading the entire page.

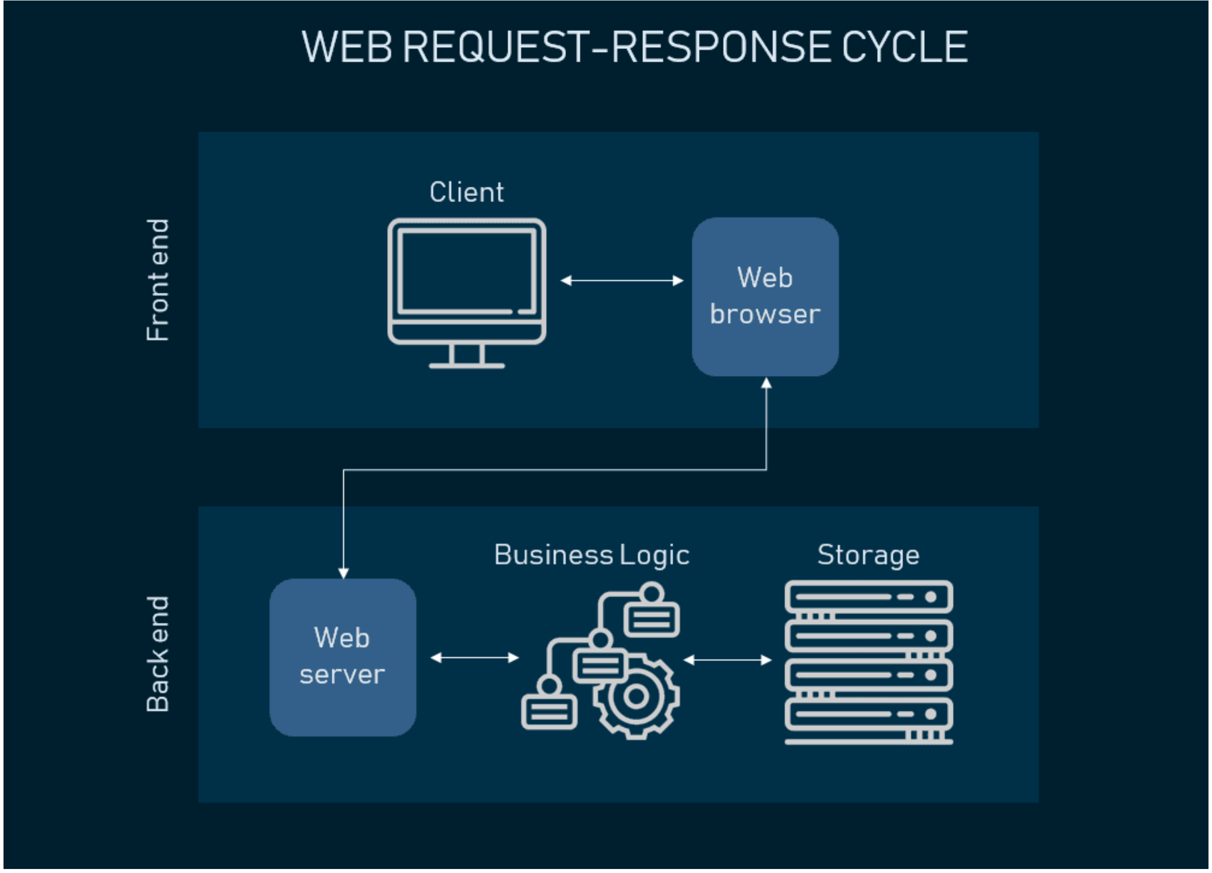
7. User Feedback

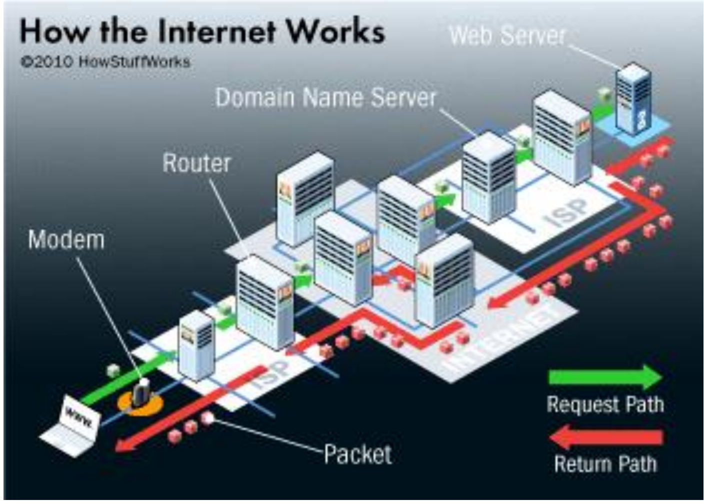
- The user sees the updated content or interface and can continue interacting with the web app. This process of sending requests and receiving responses continues as the user interacts with the app.

Summary

- **Front-End**: This is the part of the web app that runs in the browser, including HTML, CSS, and JavaScript.
- **Back-End**: This is the server-side part, which handles business logic, database interactions, and request processing.
- **Database**: Stores persistent data and interacts with the back-end to retrieve or save information.

Web apps are designed to be dynamic and interactive, allowing for rich user experiences and real-time updates.





How does internet works ?

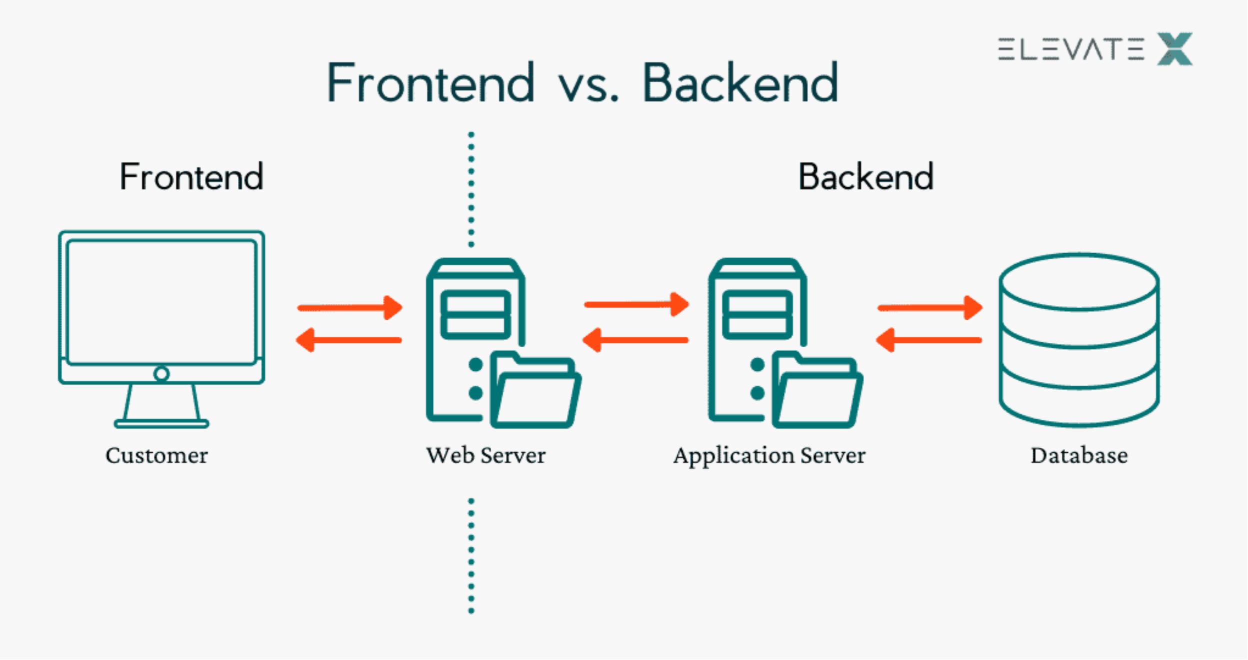
The web browser connects to the web server and sends an HTTP request (via the protocol stack) for the desired web page.

The web server receives the request and checks for the desired page. If the page exists, the web server sends it. If the server cannot find the requested page, it will send an HTTP 404 error message.

Difference between frontend and backend

Frontend : The frontend is what your users see and includes visual elements like buttons, checkboxes, graphics, and text messages. It allows your users to interact with your application.

Backend : The backend is the data and infrastructure that make your application work. It stores and processes application data for your users.

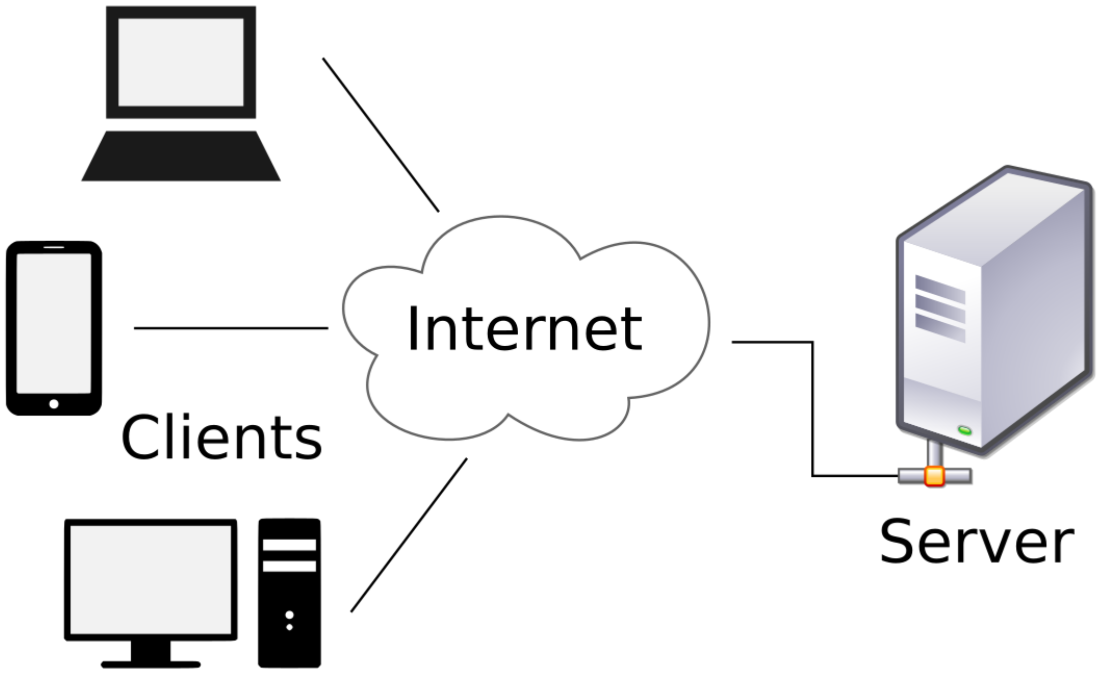


What is database ?

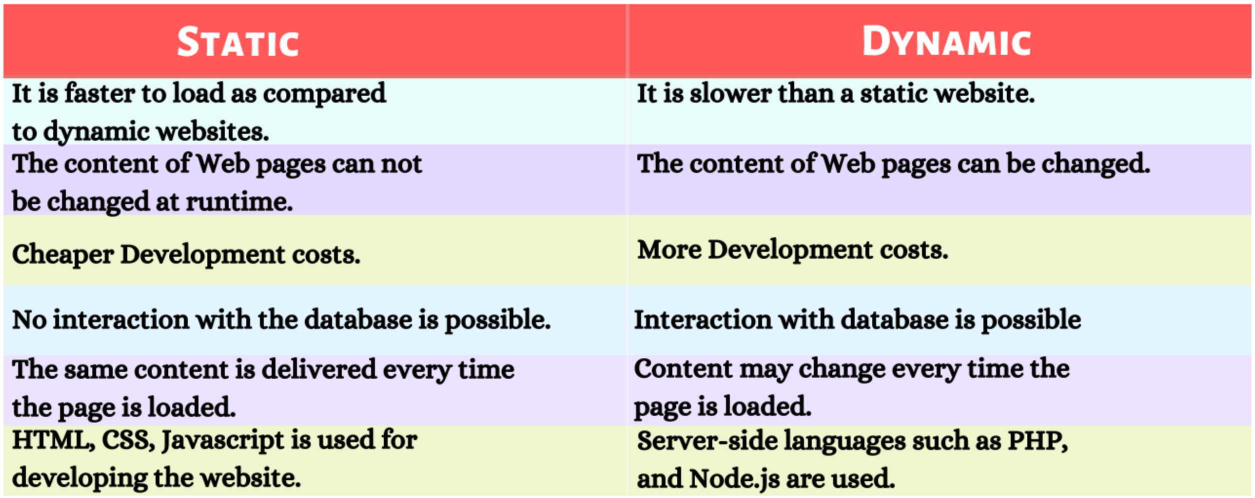
A database is **an organized collection of data or a type of data store based on the use of a database management system (DBMS)**

Client and server communication

In Client client-server communication System one process or machine acts as a client requesting a service or data, and Another machine or process acts like a server for providing those Services or Data to the client machine.



Static and Dynamic Website: Differences



Static	Dynamic
It is faster to load as compared to dynamic websites.	It is slower than a static website.
The content of Web pages can not be changed at runtime.	The content of Web pages can be changed.
Cheaper Development costs.	More Development costs.
No interaction with the database is possible.	Interaction with database is possible
The same content is delivered every time the page is loaded.	Content may change every time the page is loaded.
HTML, CSS, Javascript is used for developing the website.	Server-side languages such as PHP, and Node.js are used.

UI = User Interface

UI **refers all the elements a user interacts with, such as colors, typography, buttons, and icons**.

UX =: User Experience

User experience (UX) design is **the process of building products that are useful, easy, and enjoyable for people to use.**

What is full stack web development?

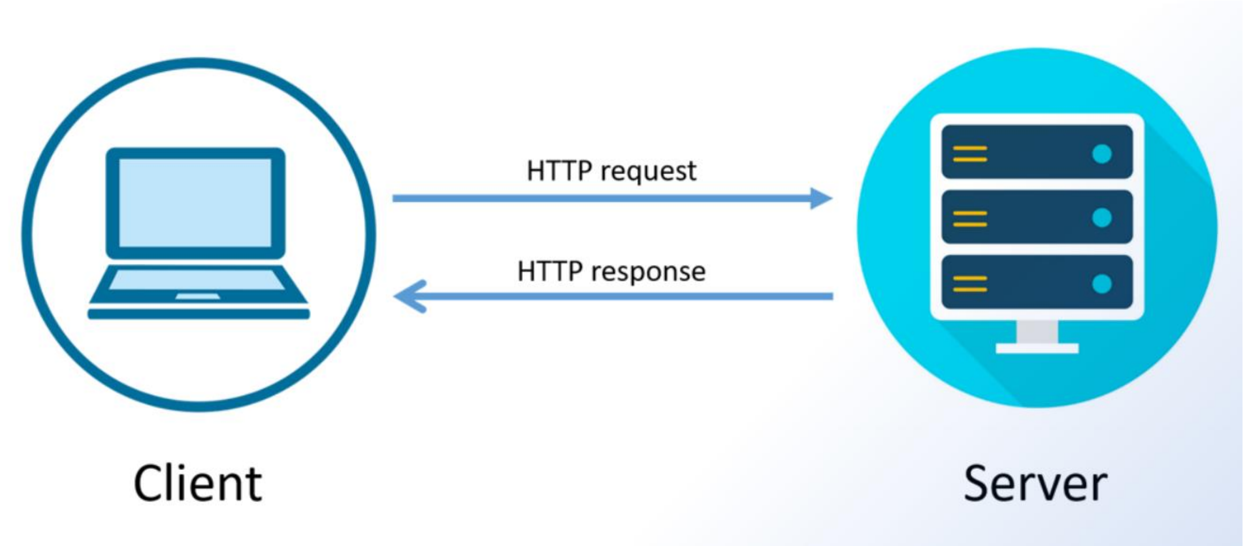
Full stack development is **the process of developing both the frontend and backend of applications**.

What is a HTTP Request/Response model ?

The request-response model defines a communication pattern where a client sends a request to a server, and the server responds with a corresponding response.

How Does a Request/Response model works?

There are various number of steps performed between a request sent by a client and response received. Let's have a general understanding of that flow…



1. **Client Sends a request:**

- The client (usually a web browser or an application) initiates the communication by sending an HTTP request to a specific URL or endpoint on the server.
- The request includes the HTTP method (such as GET, POST, PUT, DELETE) that indicates the desired action to be performed on the server, along with additional headers and sometimes a message body containing data.

2. **Server Parses the request:**

- When the server receives a request from client, It needs to parse the request to understand the desired action needs to be performed. The request parsing step has many task to do (like extracting the request method, request Headers, determining start and end of a request, etc.)

3. **Server processes the request:**

- After parsing the request, Server knows the required action to perform, That action may performing various operations, such as retrieving data from a database, processing the data, or generating a response dynamically.

4. **Server sends the response:**

- After processing the request, the server formulates an appropriate response and sends it back to the client. The response includes the necessary information, such as status codes, headers, and a response body containing data or resources.

5. **Client parses the response and uses it:**

- Upon receiving the response, the client parses and processes it based on the provided status codes, headers, and message body. The client then utilizes the response to display results on a web page or perform further actions as needed.
- **Where Request/Response Model is used ?**
- The request-response model finds application in various network protocols and systems where clients and servers interact to exchange information. It is commonly used in APIs, network communications (e.g., SMTP), SSH, and other communication protocols.

What is web server ?

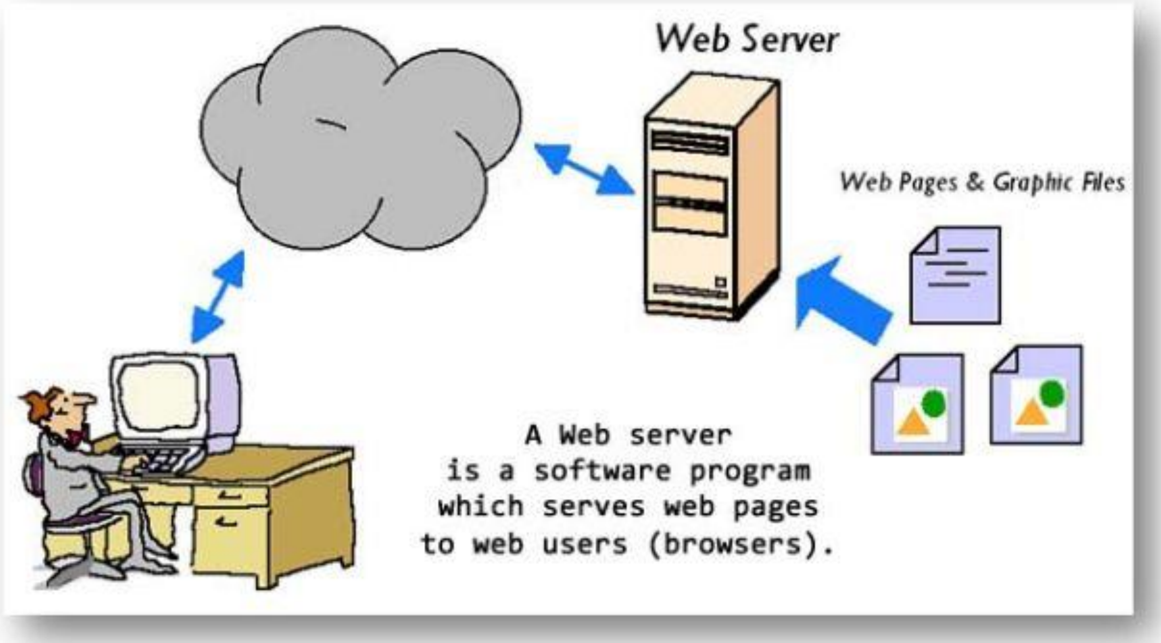
A web server is software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The main job of a web server is to display website content through storing, processing and delivering webpages to users. Besides HTTP, web servers also support SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol), used for email, file transfer and storage.

Web server hardware is connected to the internet and allows data to be exchanged with other connected devices, while web server software controls how a user accesses hosted files. The web server process is an example of the client/server model. All computers that host websites must have web server software.

Web servers are used in web hosting, or the hosting of data for websites and web-based applications -- or web applications.

How do web servers work?

When a web browser, like Google Chrome or Firefox, needs a file that's hosted on a web server, the browser will request the file by HTTP. When the request is received by the web server, the HTTP server will accept the request, find the content and send it back to the browser through HTTP.



A Web server is a software program which serves web pages to web users (browsers).