Unit 1: Introduction to Web Technology

Presented By:
Bipin Maharjan

Learning Objectives

- Explore internet essentials, differentiating between Internet, Intranet, and World Wide Web (WWW).
- Distinguish between static and dynamic web pages, and comprehend the roles of web clients and servers.
- Grasp the fundamentals of client-server architecture, including single-tier, two-tier, and multi-tier models.
- Gain insight into HTTP, understanding request and response processes, and interpreting URLs.
- Differentiate client-side scripting from server-side scripting, and recognize their significance in web development.
- Trace the evolution of the web from Web 1.0 to Web 3.0, and understand the evolving nature of internet technologies.

Table of Contents

- Web Basics:
  - Internet, Intranet, WWW
  - Static and Dynamic Web Page
  - Web Clients
  - Web Servers
- Client Server Architecture:
  - Single Tier
  - Two-Tier
  - Multi-Tier;
- HTTP: HTTP Request and Response
- URL
- Client Side Scripting
- Server Side Scripting
- Web 1.0, Web 2.0 and Web 3.0

Web Basics

Internet

The Internet is a global system of interconnected computer networks that use the TCP/IP protocol suite to link devices worldwide.

It connects millions of private, public, academic, business, and government networks.

Key Characteristics:

- **Global connectivity:** Provides access to information and communication globally.
- **Standardized communication:** Uses TCP/IP for data transmission.
- **Decentralized structure:** No single owner or controller.
- **Resource sharing:** Enables sharing of files, data, software, and hardware (like printers).

Contd…

Common Internet Services:

- Email (SMTP, POP, IMAP) – for sending and receiving messages.
- World Wide Web (HTTP/HTTPS) – for accessing web pages.
- File Transfer Protocol (FTP) – for transferring files.
- VoIP (Voice over Internet Protocol) – for Internet-based calls.
- Instant messaging, online gaming, cloud storage, etc.

Example:

When you type *www.google.com* in your browser, your computer connects to a global network (the Internet) to access Google’s web server.

Intranet

An Intranet is a private network that uses Internet technologies (like HTTP, HTML, and email protocols) but is restricted within an organization.

Purpose:

- To enable internal communication, collaboration, and data sharing among employees.
- To protect sensitive data by limiting access to authorized users only.

Features:

- Uses web browsers and web servers, just like the Internet.
- Can host internal websites, databases, and file-sharing portals.
- Accessible only to members of the organization (secured through login authentication).

Contd…

Example:

A company may have an internal site like intranet.companyname.com for HR policies, announcements, and document sharing.

Advantages:

- Enhances communication and productivity.
- Centralized information access.
- Improved data security.

World Wide Web (WWW)

The World Wide Web (WWW) is a collection of interlinked documents (web pages) that are accessed via the Internet using web browsers.

Key Concepts:

- Web Page: A single document (written in HTML) that can display text, images, videos, etc.
- Website: A collection of related web pages (e.g., www.wikipedia.org).
- Hyperlink: A clickable element that links one web page to another.
- Browser: Software used to access and display web content (Chrome, Firefox, etc.).

Contd…

How the Web Works:

1. A user enters a URL (like https://www.example.com) in a browser.
2. The browser sends an HTTP request to the corresponding web server.
3. The web server processes the request and returns the requested web page.
4. The browser displays the page to the user.

Static Web Pages

A static web page is a fixed content page created using HTML and CSS, where the content does not change unless manually edited by a developer.

Characteristics:

- Content remains the same for every visitor.
- Does not use databases.
- Files (.html, .css) are stored directly on the web server.

Contd…

Advantages:

- Easy to create and host.
- Fast loading (no server-side processing).
- Suitable for small websites (like personal portfolios or brochures).

Disadvantages:

- Manual updates required.
- No interactivity or personalization.
- Not suitable for large or data-driven websites.

Static Web Pages: Example

```
<html>
  <head><title>About Us</title></head>
  <body>
    <h1>Welcome to Our Company</h1>
    <p>We are a leading provider of IT solutions.</p>
  </body>
</html>
```

Dynamic Web Pages

A dynamic web page is generated in real-time by the web server based on user input, preferences, or database information.

Characteristics:

- Content changes dynamically.
- Uses server-side scripting languages (like PHP, Python, ASP.NET, or JSP).
- Connected to databases (like MySQL, MongoDB).

Contd…

Advantages:

- Personalized user experience.
- Easier content management (via CMS like WordPress).
- Can display live data (weather, news, prices).

Disadvantages:

- Slower than static pages (server processing time).
- Requires more resources and maintenance.

Example:

When you log in to Facebook, your feed is generated dynamically from a database — it’s different for every user.

Web Clients

A web client is any device or software that requests resources from a web server and displays the results to the user.

Types of Web Clients:

- Web Browsers: Chrome, Edge, Safari, Firefox.
- Mobile Apps: Apps that access web APIs.
- IoT Devices: Smart TVs or voice assistants fetching web data.

Contd…

Functions of a Web Client:

- Send HTTP requests to web servers.
- Interpret HTML/CSS/JavaScript to render web pages.
- Store cookies, cache, and session data.
- Support plugins or extensions to enhance functionality.

Example:

When you open www.wikipedia.org in Chrome, Chrome (the web client) requests the page from the server and displays it.

Web Servers

A web server is a software or hardware system that stores, processes, and delivers web content to clients via the HTTP protocol.

Types of Web Servers:

- Apache HTTP Server – Most popular open-source web server.
- Nginx – Lightweight, high-performance server.
- Microsoft IIS (Internet Information Services) – Windows-based server.
- LiteSpeed – Commercial high-performance server.

Contd…

Functions:

- Accept client HTTP requests.
- Process and serve static (HTML/CSS) or dynamic (PHP, ASP.NET) content.
- Manage multiple simultaneous connections.
- Provide security, logging, and error handling.

How a Web Server Works:

1. The client sends an HTTP request (e.g., “GET /index.html”).
2. The server locates the requested file or generates it dynamically.
3. The server sends an HTTP response back to the client.
4. The client’s browser renders the page for the user.

Client-Server Architecture

Overview

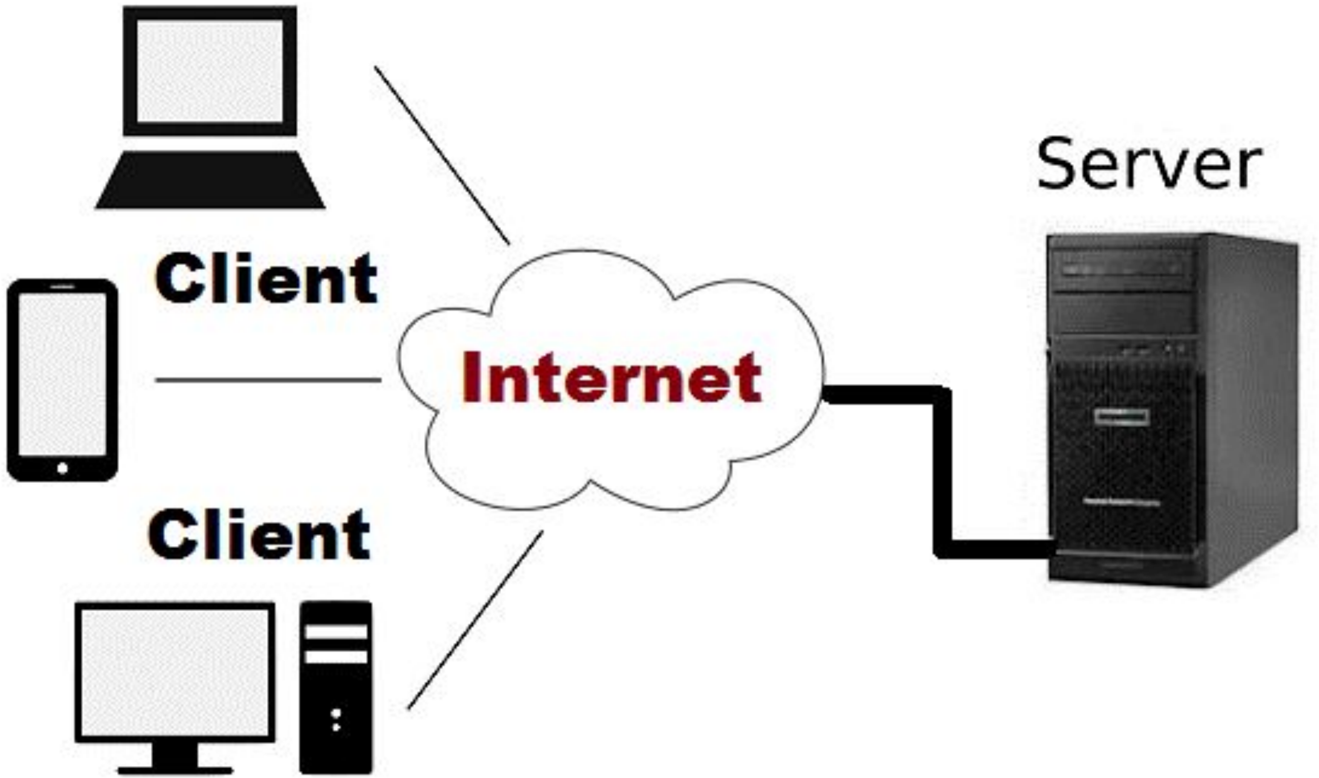
The Client–Server Architecture is the core design model of the web.

It defines how clients (users) and servers (providers of services or resources) communicate and share data over a network.

In simple terms:

> The client requests a service, and the server provides it.

This concept is fundamental for understanding how web applications, databases, and APIs function.


Client
Client
Internet
Server

Single-Tier Architecture

A Single-Tier Architecture (also known as Standalone Architecture) is a structure in which all components of an application — the user interface, business logic, and data storage — are contained within one system or layer.

There is no network communication between client and server because everything happens in the same environment.

Key Features

- Only one layer of processing.
- All tasks (input, processing, output) happen locally on a single machine.
- No need for a web server or network connection.

Contd…

Example

- A desktop calculator application:
- User interface, calculations (logic), and data (history) all exist on one computer.
- Microsoft Word (offline version).

Advantages

- Simple to design and develop.
- Fast execution (no data transfer over a network).
- No network dependency.

Disadvantages

- Not scalable — can’t handle multiple users.
- Hard to update and maintain.
- Limited resource sharing.

Two-Tier Architecture

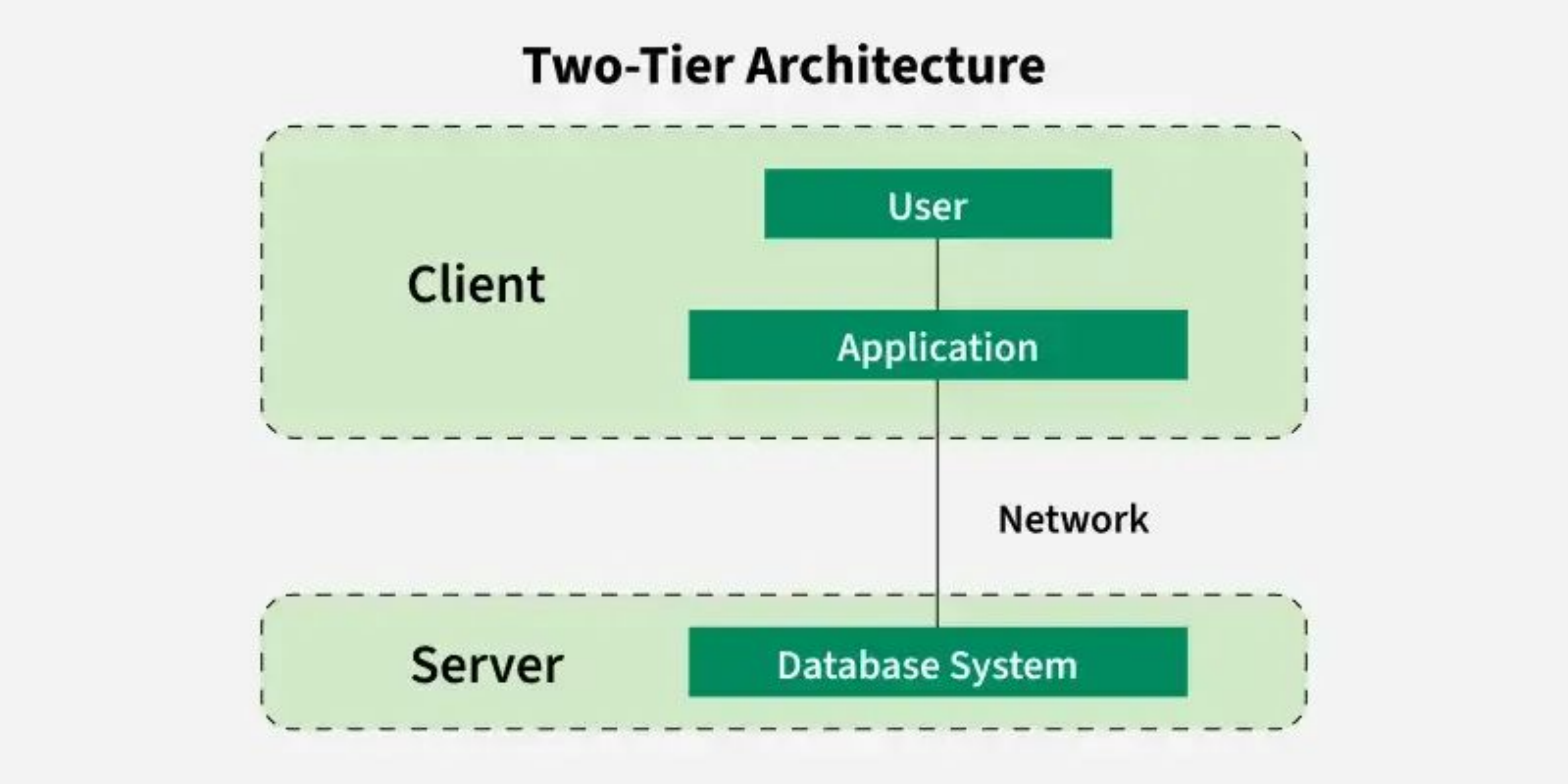
A Two-Tier Architecture splits an application into two layers:

1. Client (Front-end) – User interface and application logic.
2. Server (Back-end) – Database or data management system.

Here, the client communicates directly with the server using a protocol (often SQL or HTTP).

Key Features

- Client handles presentation and logic.
- Server stores and manages data.
- Direct communication between client and database.


Two-Tier Architecture
Client
User
Application
Network
Server
Database System

Contd…

Example

- Banking Software:
  - The client application (installed on employee computers) connects directly to the database server to retrieve account details.
- Library Management System with a front-end and a central database.

Advantages

- Easier to maintain and manage compared to single-tier.
- Provides data security (data stored on server).
- Faster than multi-tier for small systems.

Disadvantages

- Performance issues when many clients access the server simultaneously.
- Not easily scalable for web-scale systems.
- Tight coupling between client and server — changes on one side affect the other.

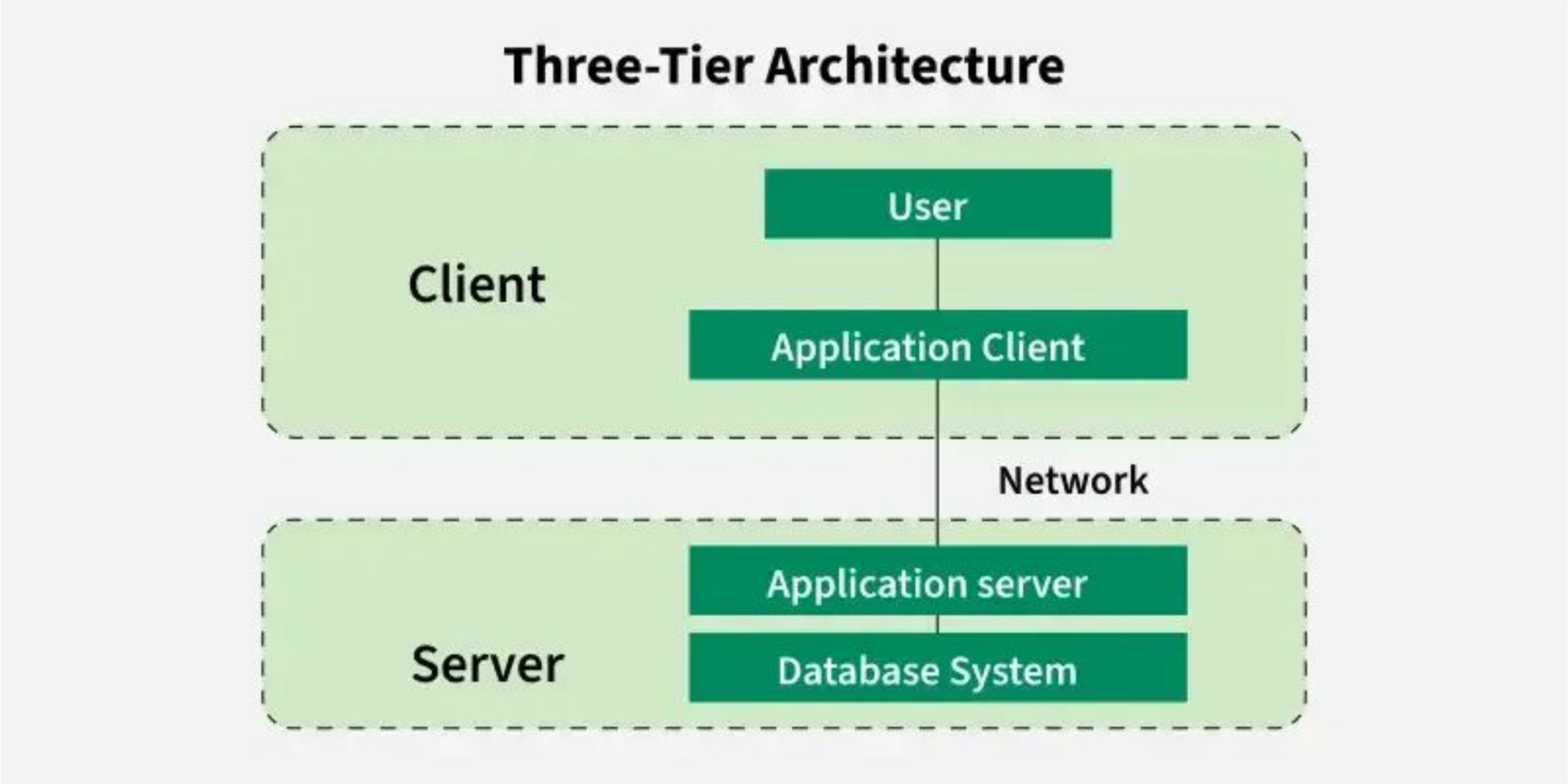
Multi-Tier Architecture (N-Tier Architecture)

- A Multi-Tier Architecture divides an application into three or more layers (tiers).
- Each tier performs a specific function, promoting modularity, scalability, and reusability.

The most common model is the Three-Tier Architecture.

Three Tiers Explained

1. Presentation Tier (Client Layer)
   a. The front-end of the application that users interact with.
   b. Technologies: HTML, CSS, JavaScript, React, Angular, etc.
2. Application Tier (Logic/Server Layer)
   a. Contains the business logic that processes client requests.
   b. Technologies: PHP, Java, Python (Django/Flask), Node.js, ASP.NET, etc.
3. Data Tier (Database Layer)
   a. Manages data storage and retrieval.
   b. Technologies: MySQL, MongoDB, PostgreSQL, Oracle.


Three-Tier Architecture
Client
User
Application Client
Network
Application server
Server
Database System

Contd…

E-commerce Website (like Amazon):

1. **Presentation Layer:** Product pages (HTML, CSS, JS).
2. **Application Layer:** Server-side scripts that handle login, cart, and payments.
3. **Data Layer:** Database storing product details, prices, and user accounts.

Advantages

- **Highly scalable:** easy to add more clients or servers.
- **Modular**: each layer can be updated independently.
- **Secure**: sensitive data is protected at the database level.
- **Better performance:** load can be distributed across servers.

Contd…

Disadvantages

- More complex to design and maintain.
- Requires more hardware resources.
- Slightly slower due to communication between multiple layers.

HTTP: HTTP Request and Response

What is HTTP?

HTTP (HyperText Transfer Protocol) is an application-layer protocol used for transmitting hypermedia documents (like HTML, images, and videos) over the Internet.

It works on top of TCP/IP and follows a request–response model between clients and servers.

Key Features:

- **Stateless Protocol:** Each request is independent; the server doesn’t retain client data between requests (unless cookies or sessions are used).
- **Connectionless:** A new connection is established for each request–response pair (in HTTP/1.0).
- **Extensible:** Supports various media types (HTML, JSON, XML, etc.).
- **Secure Version:** HTTPS (HTTP Secure) adds encryption via SSL/TLS for data privacy and security.

Introduction to HTTP

- HTTP is a communication standard governing the requests and responses that take place between the browser running on the end user’s computer and the web server.
- The server’s job is to accept a request from the client and attempt to reply to it in a meaningful way, usually by serving up a requested web page that’s why the term server is used.
- The natural counterpart to a server is a client, so that term is applied to the web browser and the computer on which it’s running.

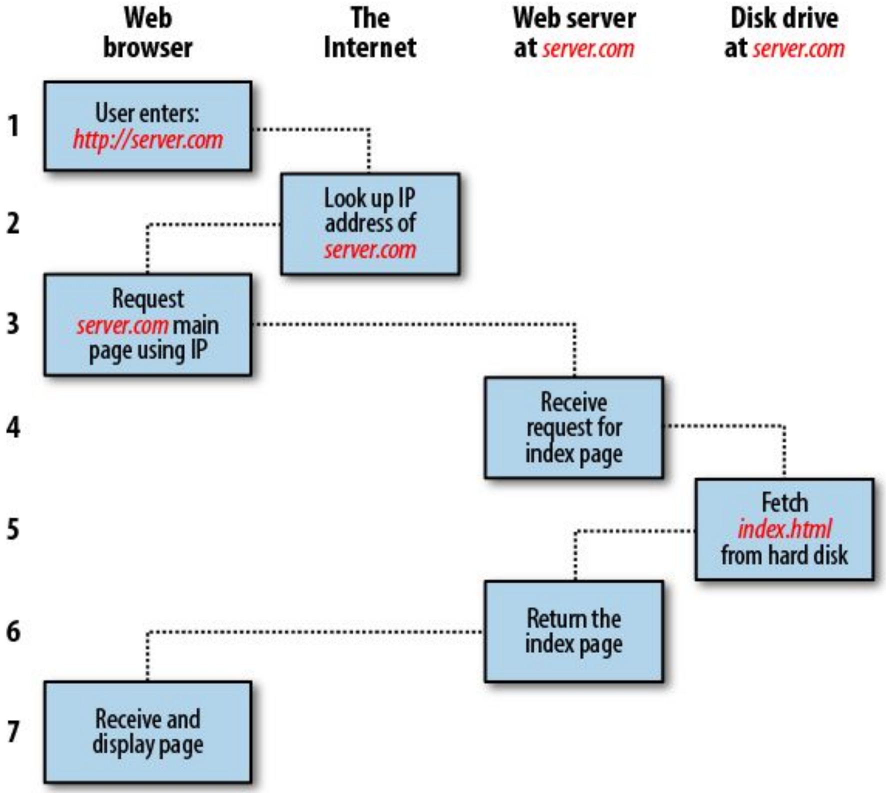
Introduction contd…

- Between the client and the server there can be several other devices, such as routers, proxies, gateways, and so on.
- They serve different roles in ensuring that the requests and responses are correctly transferred between the client and server.
- Typically, they use the Internet to send this information.
- A web server can usually handle multiple simultaneous connections and when not communicating with a client spends its time listening for an incoming connection.
- When one arrives, the server sends back a response to confirm its receipt.

The Request/Response Procedure

- At its most basic level, the request/response process consists of a web browser asking the web server to send it a web page and the server sending back the page.

- The browser then takes care of displaying the page.


Web browser
The Internet
Web server at server.com
Disk drive at server.com
1
User enters: http://server.com
2
Look up IP address of server.com
3
Request server.com main page using IP
4
Receive request for index page
5
Fetch index.html from hard disk
6
Return the index page
7
Receive and display page

Each step in the request and response sequence:

1. You enter **http://server.com** into your browser's address bar.
2. Your browser looks up the IP address for server.com.
3. Your browser issues a request for the home page at server.com.
4. The request crosses the Internet and arrives at the server.com web server.
5. The web server, having received the request, looks for the web page on its hard disk.
6. The web page is retrieved by the server and returned to the browser.
7. Your browser displays the web page.

URL

What is URL?

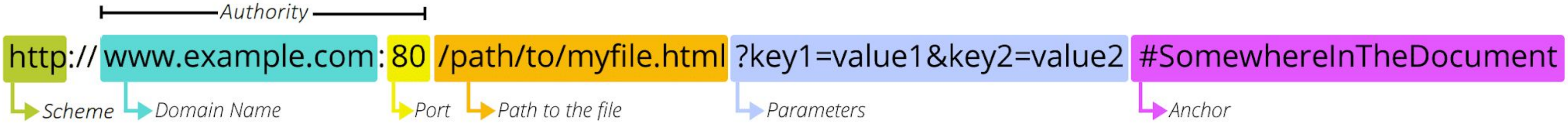
- A URL (Uniform Resource Locator) is the address of a resource (like a web page, image, or video) on the World Wide Web (WWW).
- “A Uniform Resource Locator (URL) is a compact string representation for identifying an abstract or physical resource on the Internet.” by W3C
- It tells the browser where to find a resource and how to retrieve it.
- In theory, each valid URL points to a unique resource.
- In practice, there are some exceptions, the most common being a URL pointing to a resource that no longer exists or that has moved.

Structure of a URL

A URL has several components, each serving a specific purpose.

The general format is:

protocol://domain:port/path?query#fragment


Authority
http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument
Scheme
Domain Name
Port
Path to the file
Parameters
Anchor

Example URL:

- https://www.example.com:443/products/item?id=101#reviews
- https://developer.mozilla.org
- https://developer.mozilla.org/en-US/docs/Learn_web_development/
- https://developer.mozilla.org/en-US/search?q=URL

Scheme

- Specifies the method or protocol used to access the resource.
- Common protocols include:
- **http** – HyperText Transfer Protocol
- **https** – Secure HTTP (with encryption)
- **ftp** – File Transfer Protocol
- **mailto** – Email link
- **file** – Access local files

Example:

https:// → tells the browser to use secure HTTP to communicate.

Domain Name (Host Name)

- The unique name of the server hosting the resource.
- It maps to an IP address via the DNS (Domain Name System).

Example: www.example.com

Part	Example	Description
Subdomain	`www`	Optional prefix, often used for web content
Second-level domain	`example`	Organization name
Top-level domain (TLD)	`.com`, `.org`, `.edu`	Type or category of domain

Port Number

- Indicates the network port on which the server is listening for requests.
- If omitted, browsers use default ports:
  - HTTP → Port 80
  - HTTPS → Port 443

Example:

:443 in https://www.example.com:443/

Path

- Specifies the location of the resource (file or page) on the web server.

Example:

/products/item → means “inside the ‘products’ directory, open the file or route ‘item’.”

Paths help organize website structure like folders in a computer.

Query String

- Contains data parameters sent to the server (after a ? symbol).
- Often used for searches, filters, or dynamic data.

Example:

?id=101 → tells the server to fetch the item with ID 101.

Multiple Parameters Example:

?category=shoes&brand=nike&color=black

Anchor

- Appears after #.
- Refers to a specific section within a web page.
- Used for navigation within the same page — the server doesn’t process this part.

Domain Name System (DNS) and URL Resolution

When you enter a URL like www.google.com:

1. The browser contacts a DNS server to translate the domain name into an IP address.
2. That IP address tells the browser which web server to communicate with.
3. The browser uses the HTTP/HTTPS protocol to send a request and retrieve data.

Example:

www.google.com → DNS → 142.250.190.78

Client-Side Scripting

Scripting Language

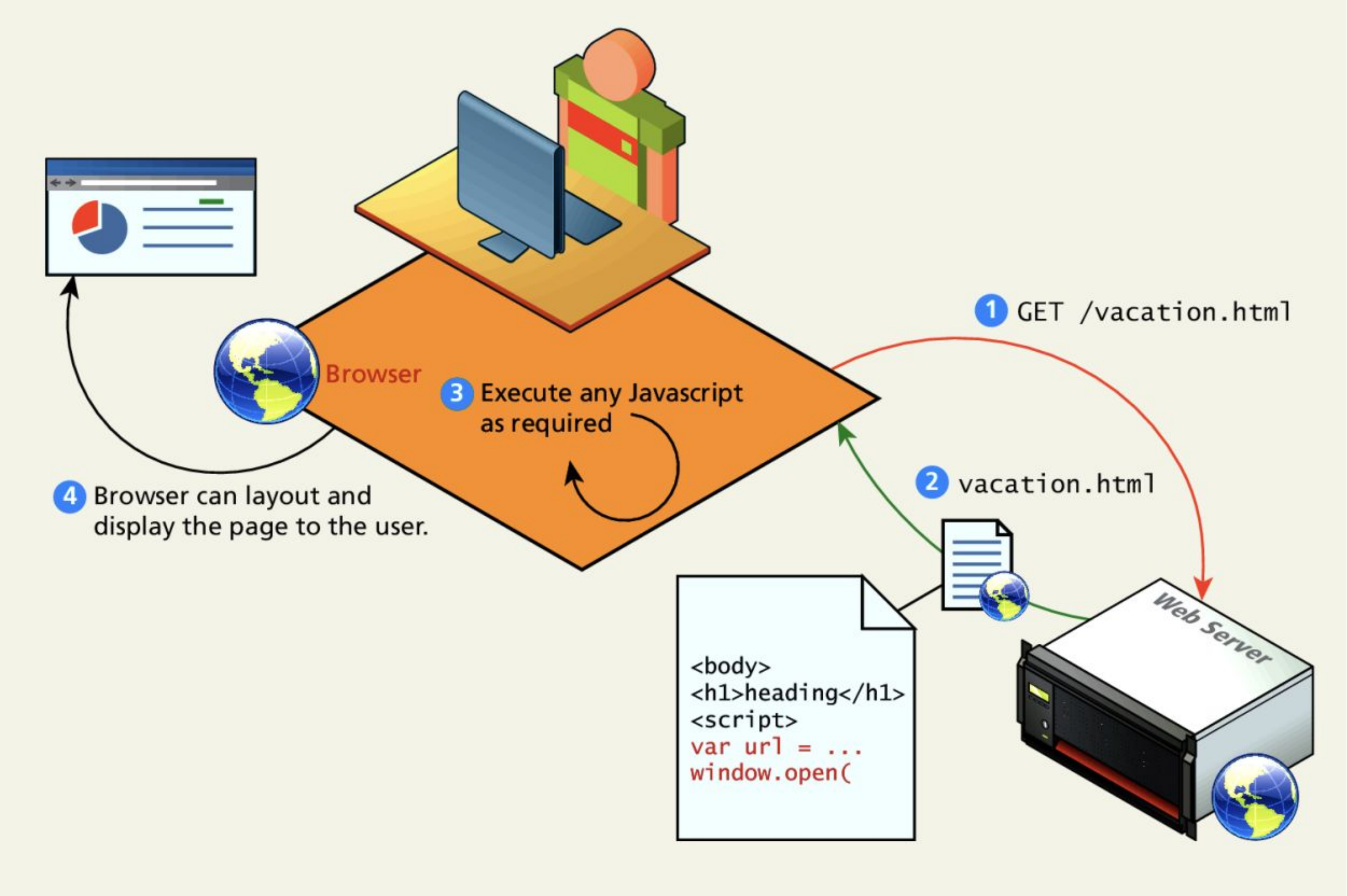
- A script is a program or sequence of instructions that is interpreted or carried out by another program rather than by the computer processor.
- A scripting language is a programming language.
- Supports scripts, programs written for a special run-time environment.
- It can interpret and automate the execution of tasks.
- It can alternatively execute one-by-one by a human operator.

Introduction to Client-Side Scripting

- Program that execute on client side, by the web browser
- Upon request, the necessary files are sent to the user’s computer by the web server on which they reside.

The Process is:

- A user requests a web page by entering a URL in the browser.
- The web server sends the page (HTML, CSS, and JavaScript) to the browser.
- The browser (client) interprets and executes the client-side scripts.
- The script interacts with the HTML Document Object Model (DOM) to modify page content, respond to events, or validate data.



1 GET /vacation.html

2 vacation.html

3 Execute any Javascript as required

4 Browser can layout and display the page to the user.

```
<body>
<h1>heading</h1>
<script>
var url = ...
window.open(
```

Client-Side Scripting Languages

- **JavaScript**: The most widely used scripting language for the web. Runs in all browsers.
- **VBScript**: Used in older Microsoft browsers (Internet Explorer). Now obsolete.
- **TypeScript**: Superset of JavaScript that adds type safety and compiles to JavaScript.

Key Components of Client-Side Scripting

- HTML (Structure)
  - Defines the content and layout of the web page.
- CSS (Style)
  - Controls the appearance and design of elements (colors, fonts, layout).
- JavaScript (Behavior)
  - Adds functionality and interactivity to make the page respond to user actions.

Advantages

- Processing can be offloaded from the server to client machines, thereby reducing the load on the server.
- The browser can respond more rapidly to user events than a request to a remote server ever could, which improves the user experience.
- JavaScript can interact with the downloaded HTML in a way that the server cannot, creating a user experience more like desktop software than simple HTML ever could.
- Offline Capability: Some actions can work without an active internet connection.

Disadvantages

- Security Risks: Code is visible and modifiable by users.
- Browser Compatibility Issues: Not all browsers handle scripts identically.
- Limited Access: Cannot access server-side files, databases, or system hardware.
- Disabled JavaScript: Some users turn off scripting for security reasons, causing features to break.

Server-Side Scripting

Introduction to server-side scripting

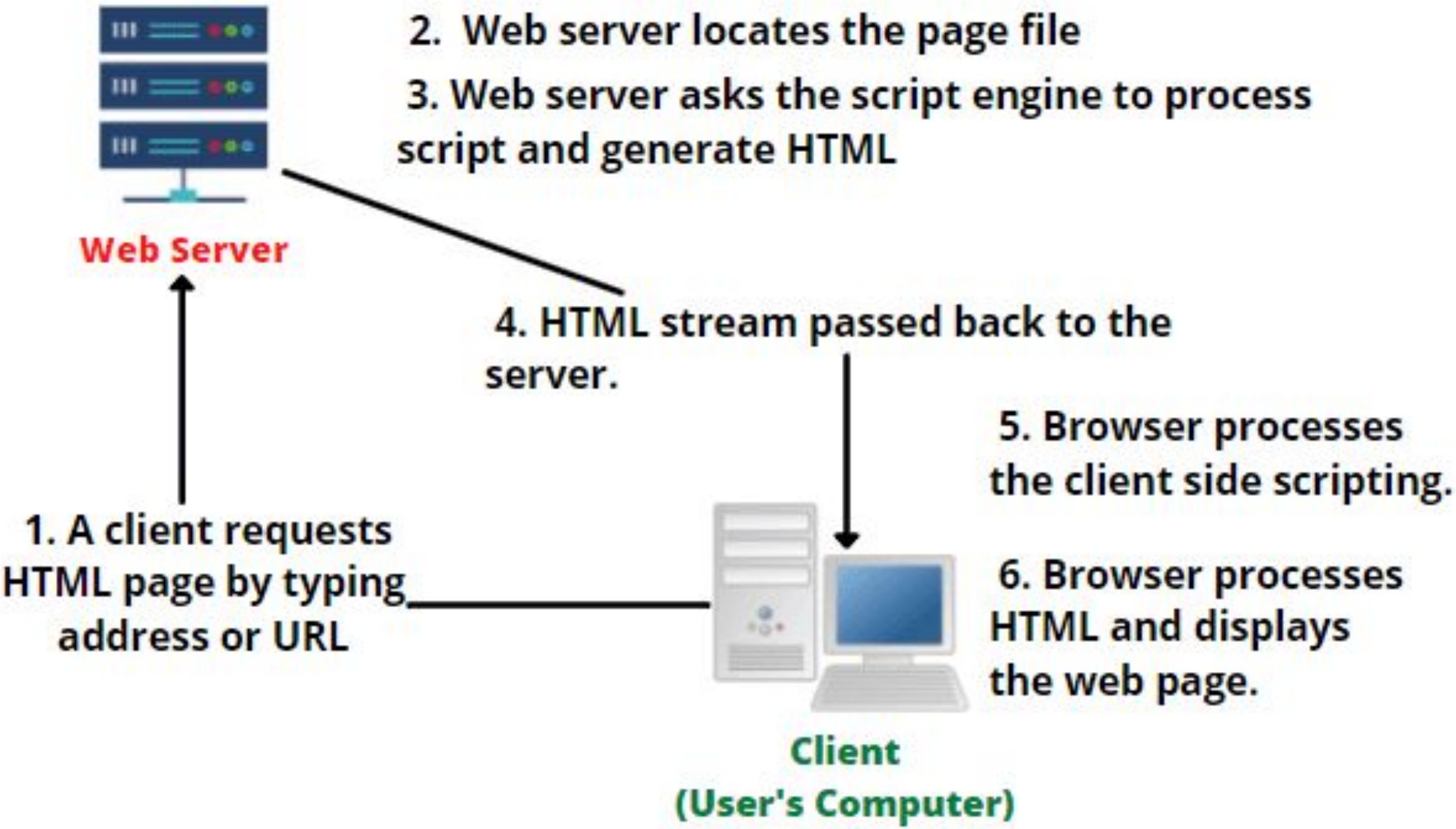
- The server is where the Web page and other content lives.
- The server sends pages to the user/client on request.

The Process is:

- The user requests a Web page from the server the script in the page is interpreted by the server creating or changing the page content to suit the user and the occasion and/or passing data.
- Around the page in its final form is sent to the user and then cannot be changed using server-side scripting.

Introduction to server-side scripting contd…

- Server-side scripting tends to be used for allowing users to have individual accounts and providing data from databases. It allows a level of privacy, personalization and provision of information that is very powerful.
- E-commerce and social networking sites all rely heavily on server-side scripting.
- Server-side scripts are never seen by the user.
- They run on the server and generate results which are sent to the user.



Web Server

2. Web server locates the page file

3. Web server asks the script engine to process script and generate HTML

4. HTML stream passed back to the server.

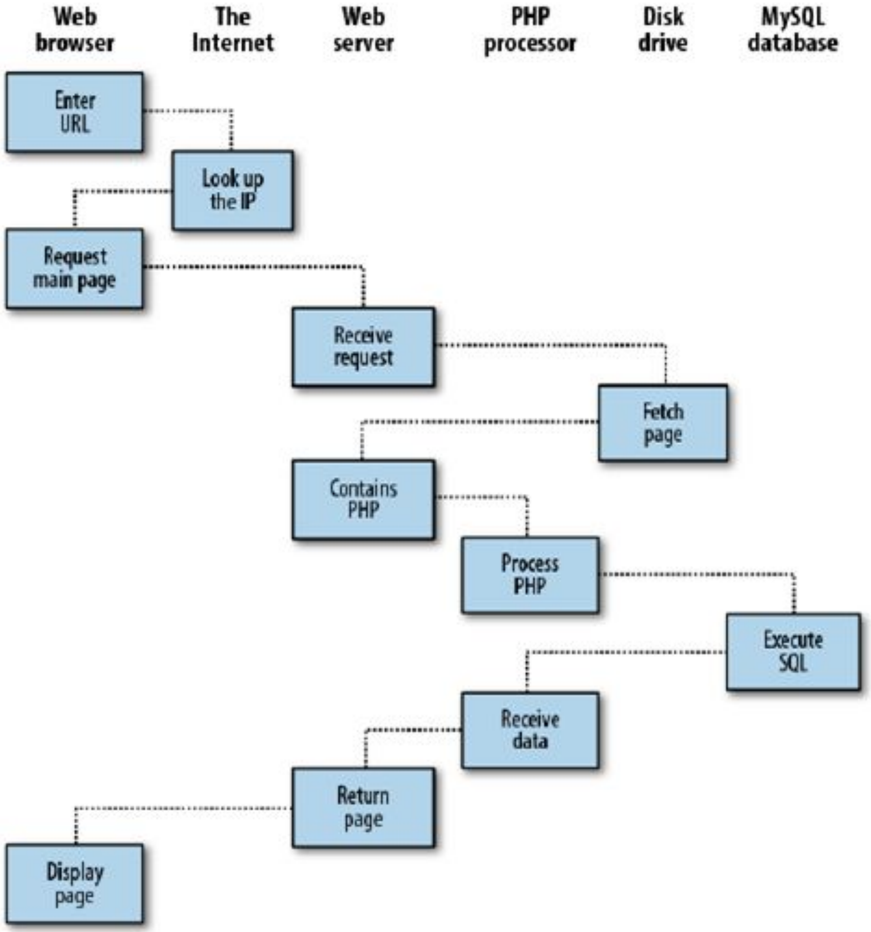
5. Browser processes the client side scripting.

6. Browser processes HTML and displays the web page.

1. A client requests HTML page by typing address or URL

Client (User's Computer)

Figure: Server-side and Client-side scripting


Web browser
The Internet
Web server
PHP processor
Disk drive
MySQL database
Enter URL
Look up the IP
Request main page
Receive request
Fetch page
Contains PHP
Process PHP
Execute SQL
Receive data
Return page
Display page

Examples of Server-side Scripting Language

- ASP (*.asp)
- ASP.NET (*.aspx)
- JavaServer Pages
- PHP (*.php)
- Python (*.py)
- Ruby (*.rb, *.rbw)

Advantages

- **Enhanced Security:** Code is hidden from users; better control over data access.
- **Database Integration:** Easy to connect to databases and handle large data sets.
- **Personalization:** Customizes content for users (like greetings or recommendations).
- **Complex Logic Handling:** Can process large computations or business logic.
- **Centralized Control:** Easy to manage and update at server level.

Disadvantages

- **Server Load:** All scripts run on the server, increasing workload.
- **Slower Response:** Requires communication with the server for each request.
- **No Offline Use:** Needs an active internet connection.
- **Scalability Issues:** Heavy traffic can slow down response time.

Web 1.0, Web 2.0 and Web 3.0

Introduction to WWW

The World Wide Web (WWW) has gone through several phases of development, each bringing new capabilities, technologies, and user experiences.

These phases are commonly known as:

- Web 1.0: The Static Web
- Web 2.0: The Social and Interactive Web
- Web 3.0: The Intelligent and Decentralized Web

Web 1.0 - The Static Web (1990s - Early 2000s)

- The first version of web (Web 1.0) also referred to as Syntactic web or read only web in the era (1990 - 2000).
- The role of a user is limited to reading information provided by the content producers.
- There is no option given for user or consumer to communicate back the information to the content producers. Example of Web 1.0 are static web sites and personal sites.
- Early websites like Yahoo (1996), Netscape, Geocities, and MSN.

Web 1.0 Features

- **Read-Only Web:** Users could only view information — no interaction.
- **Static Pages:** Content was fixed; no real-time updates.
- **Simple Design:** Text, images, and links; little or no multimedia.
- **Limited User Participation:** No comments, likes, or social sharing.
- **Technologies Used:** HTML, CSS, GIFs, basic JavaScript.
- **Web Servers:** Served the same content to all users.

Contd…

Advantages

- Simple to design and host.
- Fast loading (due to static content).
- Low server resource usage.

Limitations

- No user interactivity.
- Difficult to update content.
- No personalization or dynamic data.
- Limited functionality for e-commerce or social networking.

Web 2.0: The Social and Interactive Web (2004 – Present)

- The Web 2.0 also referred as Social Web or read-write web in the era (2004 - 2010 and continues even now)
- In this era every user can be a content producer and content is distributed and shared between sites.
- Some of the famous Web 2.0 applications are Facebook, Youtube, Flickr, Twitter etc.
- It uses web technologies like HTML5, CSS3 and JavaScript frameworks like ReactJs, AngularJs, VueJs etc.
- Examples:
    - Social Media: Facebook, Instagram, Twitter.
    - Collaborative Tools: Google Workspace, Trello, Slack.
    - E-commerce: Amazon, Flipkart.
    - Video & Streaming: YouTube, Netflix.

Web 2.0 Features

- **Dynamic Content:** Content updates in real-time.
- **User Interaction:** Comments, likes, posts, reviews, social sharing.
- **Rich User Interface (UI):** Responsive design and multimedia.
- **Social Networking:** Platforms like Facebook, YouTube, Instagram.
- **Collaborative Tools:** Wikipedia, Google Docs, Blogs, Forums.
- **Asynchronous Communication:** AJAX allows partial page updates.
- **APIs and Web Services:** Enable integration between apps.
- **Client & Server-Side Scripting:** JavaScript, PHP, Python, Node.js.

Contd…

Advantages

- Enhanced user engagement and interaction.
- Easy content creation (blogs, posts, videos).
- Real-time communication (chat, notifications).
- Cloud-based services and storage.
- Improved accessibility and mobile compatibility.

Limitations

- **Centralized control:** Data controlled by big tech companies.
- **Privacy concerns:** User data collected and monetized.
- **Security risks:** Identity theft, data breaches.
- **Dependence on servers:** Requires internet connectivity.

Web 3.0: The Semantic and Decentralized Web

- The Web 3.0 also referred as Semantic Web or read-write-execute in the era (2010 and above) which refers to the future of web.
- In this era computers can interpret information like humans via Artificial Intelligence and Machine Learning.
- Which help to intelligently generate and distribute useful content tailored to a particular need of a user.
- It aims to make the web smarter, more secure, and user-centric using technologies like AI, Blockchain, and Machine Learning.
- Examples:
    - Blockchain Platforms: Ethereum, Polkadot, Solana.
    - Decentralized Apps (DApps): Uniswap, OpenSea, Brave Browser.
    - AI-Based Systems: ChatGPT, Google Assistant, Siri.
    - Decentralized Storage: IPFS, Filecoin.

Web 3.0 Features

- **Decentralization:** Data stored on blockchain networks instead of centralized servers.
- **Semantic Understanding:** Machines can interpret and connect data contextually.
- **Artificial Intelligence (AI):** Personalized content and intelligent search.
- **Blockchain Technology:** Transparent, tamper-proof, and secure transactions.
- **Smart Contracts:** Automated agreements executed on the blockchain.
- **Cryptocurrencies & NFTs:** Digital assets owned and traded securely.
- **Interoperability:** Cross-platform data sharing and integration.
- **Edge and Cloud Computing:** Distributed storage and computation.

Contd…

Advantages:

- Greater privacy and data ownership for users.
- Reduced dependency on centralized platforms.
- Smarter search and recommendations using AI.
- Transparent and secure transactions (blockchain).
- Cross-platform connectivity and automation.

Limitations:

- Complex technology (AI, blockchain) — hard to implement.
- High computing and energy requirements.
- Lack of universal standards and regulation.
- Slow adoption and scalability issues.

Any Questions?