University of Tissemsilt
Faculty of Science & Technology
Departement of Math and Computer Science

Web Application Development

February 6, 2024

Lecturer

Dr. HAMDANI M

Speciality : Computer Science (ISIL)
Semester: S4

Plan

Introduction

Web Application Development refers to the process of creating and maintaining software applications that are accessed over the internet through web browsers.

Types of web developers

- **Frontend developers**: Frontend developers implement web page designs using HTML and CSS. They make sure the website looks pretty on different devices, and that the forms and buttons work.

Types of web developers

- **Frontend developers**: Frontend developers implement web page designs using HTML and CSS. They make sure the website looks pretty on different devices, and that the forms and buttons work.
- **Backend developers**: Backend developers create the backbone of the web application. They write code logic that handles a user's input (for example, what should happen when you click the signup button after filling in a form).

Types of web developers

- **Frontend developers**: Frontend developers implement web page designs using HTML and CSS. They make sure the website looks pretty on different devices, and that the forms and buttons work.
- **Backend developers**: Backend developers create the backbone of the web application. They write code logic that handles a user's input (for example, what should happen when you click the signup button after filling in a form).
- **Full stack developers**: Full stack developers do bits of both backend and frontend.

Internet vs. WWW

Most people use the two terms interchangeably but they are in fact different.

- **The Internet** is a vast, international network, made up of computers and the physical connections (wires, routers, etc.) allowing them to communicate.
- **The World Wide Web** (WWW or just the Web) is a collection of software that spans the Internet and enables the interlinking of documents and resources.
  Provides a way of accessing information on the Internet.

Web Apps Compared to Desktop Apps

Advantages of web apps :

- Accessible from any internet-enabled computer.
- Usable with different operating systems and browser platforms.
- Easier to roll out program updates since only need to update software on server and not on every desktop in organization.
- Centralized storage on the server means fewer concerns about local storage (which is important for sensitive information such as health care data).

Web Apps Compared to Desktop Apps

Disadvantages of web apps :

- Internet is not always available everywhere at all time).
- Security concerns about sensitive private data being transmitted over the internet.
- Concerns over the storage, licensing and use of uploaded data.
- Problems with certain websites on certain browsers not looking quite right.
- Limited access to the operating system can prevent software and hardware from being installed or accessed (like Adobe Flash on iOS).

Internet

- Global public network.
- Accessible by anyone with an internet connection.
- Emphasis on external security.
- Open for public information sharing.
- Users: Worldwide public.

Intranet

- Private network within an organization.
- Access limited to organization members.
- Emphasis on internal security.
- Primarily for internal collaboration.
- Users: Employees or members.

Extranet

- Extends to external parties.
- Controlled access for trusted stakeholders.
- Emphasis on both internal and external security.
- Facilitates external collaboration.
- Users: External parties (e.g., clients, partners).

Static Websites

- Fixed Content
- HTML, CSS, and Limited JavaScript
- Manual Updates
- Limited Interactivity
- Examples: Personal Blogs, Brochure Sites

Dynamic Web Sites

- Dynamic Content
- Server-Side Scripting (PHP, Python, etc.)
- Easy Updates (Content Management Systems)
- High Interactivity (User Logins, E-commerce)
- Examples: Social Media, E-commerce, Web Applications

Server-side and client-side

- Server-side processes are executed on the web server.
- Client-side processes are executed on the user's device.

Server-Side

- **Execution Location**: Code runs on the web server.
- **Languages**: Commonly uses server-side languages like PHP, Python, Ruby, Node.js, etc.
- **Responsibilities**: Handles server operations, database interactions, and business logic.
- **Data Processing**: Data processing occurs on the server.
- **Security**: Secure for sensitive data and logic.
- **Examples**: Content management systems (CMS), e-commerce platforms, web applications.

Client-Side

- **Execution Location**: Code runs in the user's web browser.
- **Languages**: Primarily JavaScript.
- **Responsibilities**: Enhances user interface, interactivity, and user experience.
- **Data Processing**: Limited data processing; relies on server for critical operations.
- **Security**: Limited security for sensitive data and logic.
- **Examples**: Interactive websites, single-page applications (SPAs), browser games.

Server-side and client-side : Key take-aways

- Server-side and client-side refer to the location where certain tasks or processes are carried out in a web application.
- Server-side processes are executed on the web server before the web application is delivered to the user's device.
- Client-side processes are executed on the user's device after the web application is delivered.
- Server-side processes have more access to resources and are more secure, while client-side processes have less access to resources and are potentially less secure.

Internet

- Internet is a world-wide global system of interconnected computer networks.
- It uses the standard Internet Protocol (TCP/IP).
- Every computer is identified by a unique IP address. IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer location.
- DNS (Domain Name Server) is used to give name to the IP Address so that user can locate a computer by a name.

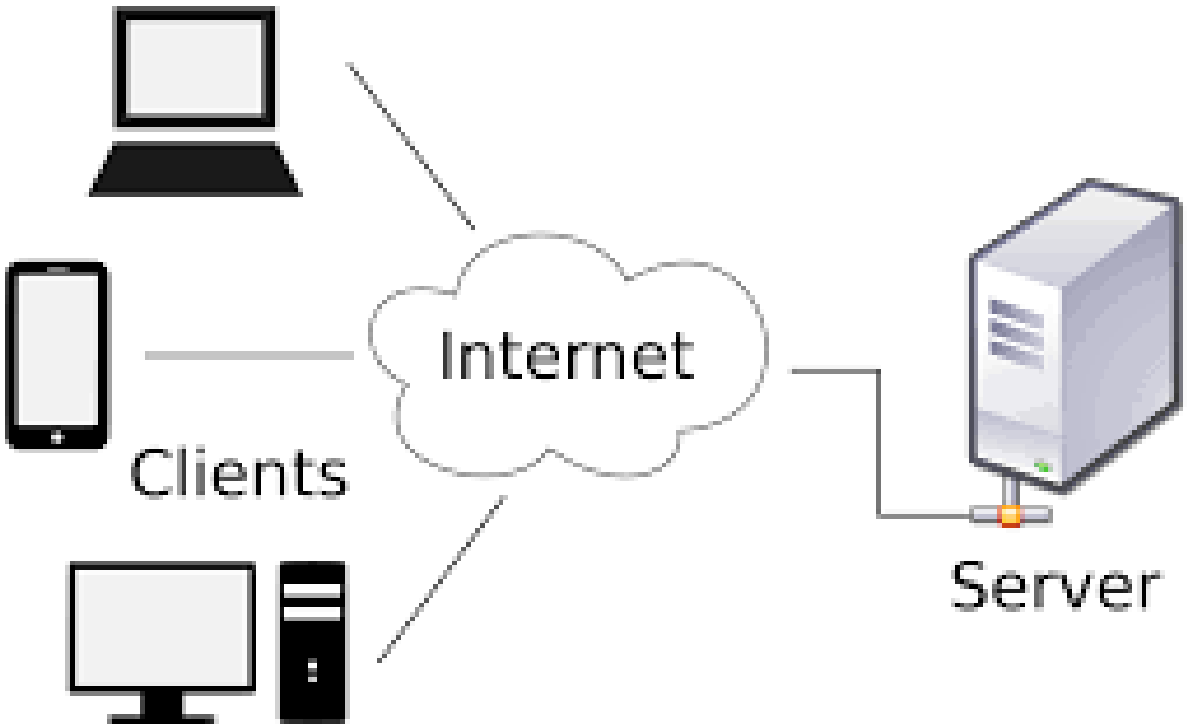
History

- Network of networks, a networking infrastructure
- Created in 1969 as ARPANET by the United States Department of Defense
- On October 29, 1969, the first internet message was sent
- The first message to be distributed was "LO", which was an attempt at "LOGIN" by Charley S. Kline to log into the computer from UCLA (University of California, Los Angeles)
- In the 1980s this networking infrastructure was made available to the general public

World Wide Web

- Tim Berners-Lee introduced WWW in 1989 and became available for everyone August of 1991.
- A distributed document delivery system.
- The documents are formatted in a markup language called HTML (HyperText Markup Language).
- Web browsers make it easy to access the World Wide Web.
- The World Wide Web uses a client-server model.

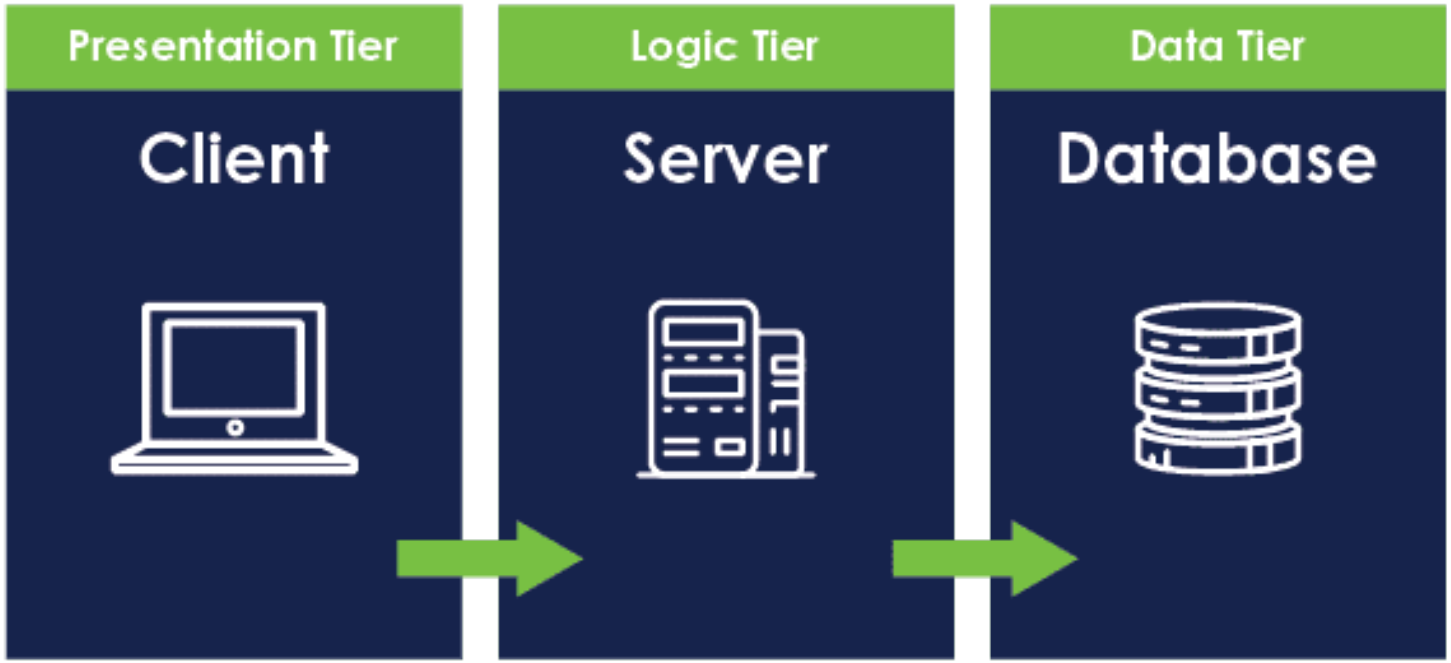
Client-Server Architecture



Three-Tier Client/Server Architecture

- To overcome the limitations of two-tier architecture, a middle tier server (also called gateway) is added between the user machine (typically a thin client) and the backend servers (database servers).
- The middle tier is where the application/business logic of the system resides and it performs a number of different functions like mapping different database queries, integrating results from different data sources and interfacing with backend legacy applications

Three-Tier Architecture



Protocol

In diplomatic circles, a protocol is the set of rules governing a conversation between people

Client and server carry on a machine-to-machine conversation

A network protocol is the set of rules governing a conversation between a client and a server

There are many protocols, HTTP is just one

URI

- A Uniform Resource Identifier (URI) is a string of characters that provides a compact and standardized way to identify or locate resources on the internet or in other contexts.
- can be further categorized into two main types:
  - Uniform Resource Locators (URLs)
  - Uniform Resource Names (URNs)

URL

Address used to locate resources on the internet.

https://www.example.com:8080/path/to/resource?param1=value1&param2=value2#section2

- Protocol: "https://"
- Domain: "www.example.com"
- Port: ":8080" (optional, specifying a custom port)
- Path: "/path/to/resource"
- Query: "?param1=value1&param2=value2"
- Fragment: "#section2"

URN

- URNs are another type of URI that serves as a persistent and location-independent identifier for resources.
- Unlike URLs, URNs do not specify how to access the resource but rather provide a unique name or identifier for it.
- URNs are intended to remain stable and unchanged over time, even if the resource's location or access methods change.
- Examples of URNs
  - "urn:isbn:0451450523" for identifying books by ISBN and
  - "urn:doi:10.1007/978-3-642-04898-2_1" for identifying digital objects by DOI.

HTTP (**H**yper**T**ext **T**ransfer **P**rotocol)

- Set of rules governing the format and content of the conversation between a Web client and server
- Uses TCP as its underlying transport protocol
- Uses port 80
- It's a text-based communication protocol
- Stateless Protocol (doesn't require a server to retain the information of a session or the status of every communicating partner in multiple requests)

Type of data transfered

Protocol for transfer of various data formats between server and client

- Plaintext
- Hypertext
- Images
- Video
- Sound

Meta-information also can be transferred

Message structure

HTTP requests and responses consist of headers and an optional message body.

Headers contain metadata about the message, such as content type and length.

Message Structure

HTTP attaches a header, which contains information such as:

- Name and location of the page being requested,
- Name and IP address of the remote server that contains the Web page,
- IP address of the local client,
- HTTP version number,
- URL of the referring page.

Example Request

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
```

Example Response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1234

<html>
  <body>
  ...
  </body>
</html>
```

Static Pages

- Content does not change based on user interactions or other factors.
- Simple to create and maintain, as they don't require server-side scripting or database interactions.
- Static pages Load quickly because their content is pre-generated and stored on a server, meaning they don't need to process complex requests or retrieve real-time data.
- Often used to display general information, such as company about pages, contact pages, or simple homepages.
- Typically written in HTML and can include images, text, and links.

Dynamic Pages (1)

- Real-time Updates: Dynamic pages can update their content without requiring the user to reload the entire page.
- User Interactions: They can respond to user actions, such as form submissions, button clicks, or menu selections, by processing user input and displaying relevant information or performing specific actions.
- Data Retrieval: Dynamic pages often involve interactions with databases or external APIs to retrieve and display data in real-time. (ex. product listings, search results, user profiles, ..)

Dynamic Pages (2)

- Personalization: *DP* can provide personalized content to users based on their preferences, behavior, or account information. For example, an e-commerce website may display personalized product recommendations.
- Server-side Processing: *DP* are typically generated on the server-side using programming languages like PHP, Python, Ruby, or JavaScript (Node.js).

What is SEO

- SEO stands for Search Engine Optimization.
- It's the practice of optimizing web content to rank higher in search engine results pages (SERPs).
- The process of improving the visibility of a website or a web page in search engines via the "natural," or un-paid ("organic" or "algorithmic"), search results

How do organic search listings work?

- A ***spider*** or ***crawler*** which is a component of a SE gathers listings by automatically "crawling" the web
- The spider follows links to web pages, makes copies of the pages and stores them in the SE's index
- Based on this data, the SE then indexes the pages and ranks the websites
- Major SEs that index pages using spiders: Google, Yahoo, AltaVista, MSN, AOL, Lycos

Why SEO Matters ?

- **Increased Visibility**: Higher search rankings lead to more visibility for your website.
- **Organic Traffic**: SEO can drive organic (non-paid) traffic to your site.
- **User Trust**: High-ranking sites are often perceived as more trustworthy.
- **Competitive Advantage**: SEO can give you an edge over competitors
- Accessible websites are more SEO-friendly.

SEO Elements in Web Development

- **Content**: High-quality, relevant content is essential for SEO.
- **Keywords**: Research and use keywords strategically in your content.
- **HTML Tags**: Proper use of headings, meta tags, and alt attributes.
- **Site Speed**: Fast-loading websites rank better.
- **Mobile Optimization**: Mobile-friendly sites are favored by search engines.
- **Backlinks**: Quality inbound links from other sites improve SEO.

Popular SEO Tools (1)

- Google Analytics: A comprehensive web analytics tool that tracks website traffic, user behavior, and more.
- Google Search Console: Offers insights into how Googlebot views your website, monitors search performance, and helps fix issues.
- Keyword Research Tools: Tools like Google Keyword Planner, SEMrush, and Ahrefs assist in finding relevant keywords.
- On-Page SEO Tools: Tools like Moz and Yoast SEO provide on-page optimization recommendations.

Popular SEO Tools (2)

- Backlink Analysis Tools: Tools like Majestic and Moz help analyze and manage backlinks.
- SEO Auditing Tools: Tools like Screaming Frog and Site-bulb perform in-depth site audits.
- Rank Tracking Tools: Monitor your website's search engine rankings using tools like Rank Tracker.

Questions ?