Republic of Iraq
The Ministry Of Higher Education & Scientific Research
Department

University Of Technology

Computer Science

WEB INFORMATIN SYSTEM

ا.م.د.سهير محمد زكي عبد الصمد

2024 -2023

Web Information System

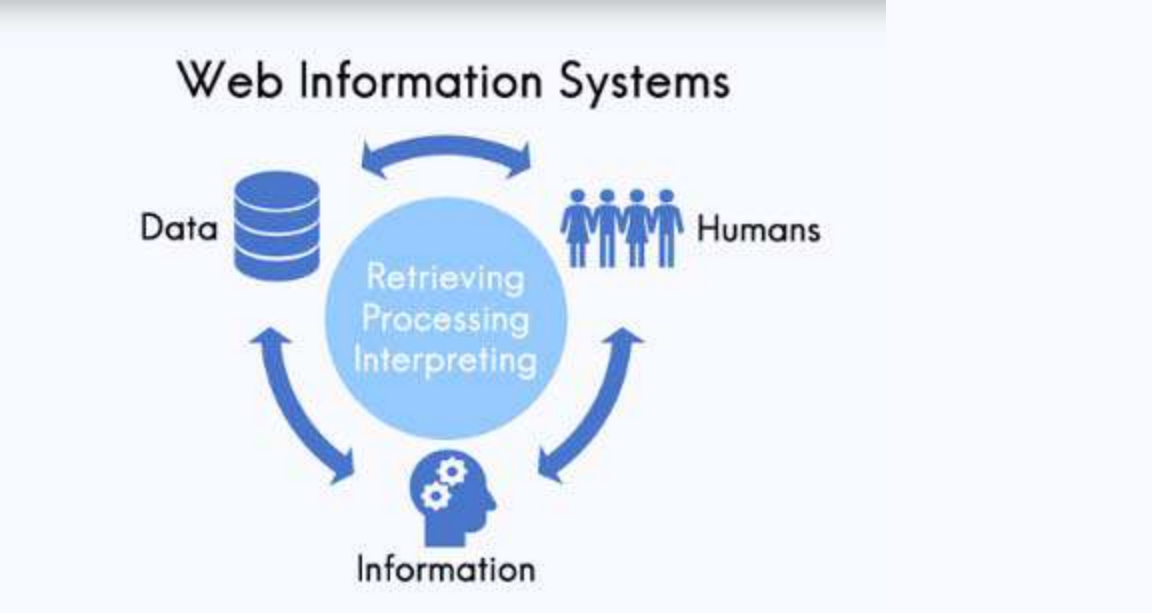
➢ General Aspects For Web Information Systems (WIS)

- **WIS**
- **General terms**
  - ✓ **Data**
  - ✓ **Metadata**
  - ✓ **Information**
  - ✓ **Knowledge**
  - ✓ **ontology**

Web Information Systems (WIS) group focuses on research & education in data-based information systems on the Web and making them more effective in retrieving, processing and interpreting data generated by humans and machines.

overall mission of the WIS research group is:

- **to develop a deep understanding of the construction and use of web-based information systems,**
- **to develop novel methods, techniques, and tools that advance the way in which web-based information systems are constructed and used, and**
- **To offer students education that prepares them for a leading role in web-based information systems science and technology.**



What is a Web information system?

A web-based information system is an information system that uses a software system and shares different types of internet protocol and plays an essential role in publishing and analyzing data by hypertext principle and showing cast information and services at a glance

What are the benefits of web-based systems?

Benefits get from using web-based systems and applications.

- **1 Enhanced Collaboration. ...**
- **2 Improved Flexibility and Accessibility. ...**
- **3 Improved Data Security. ...**
- **4 Loss Prevention. ...**
- **5 Automatic Software Updates. ...**
- **6 Data-Driven Decision Making. ...**
- **7 Cost Savings. ...**
- **8 Competitive Edge**

Characteristics of Online Information Retrieval System:

1. **Direct Access: Users can directly access online storage to gather desired information to use computer input/output devices and communication channels.**
2. **Conventional Mode: It is used to the following media channels i.e. computer, software, network, internet, and so on.**
3. **Two-way Communication: A terminal and a server network use to interconnect each other.**
4. **Centralized Storage: Whole resources find on the querying method on a centralized database.**
5. **Centralized Control: All the process of tasks is controlled by a centralized server.**
6. **Rapid Response: This system response in a convenient way when the system receives a new query request from the server.**
7. **Real-Time Communication: to respond immediately when a new search query found.**
8. **Modern Process: This system follows a modern way to retrieve information from online databases.**
9. **Effective Communication: It is an effective way of communication between man and machine.**

What are Data, Information, and Knowledge

Data, information, and knowledge are often used interchangeably. However, these terms represent different stages of value creation from data to decision-making.

Data are the raw alphanumeric values obtained through different acquisition methods. Data in their simplest form consist of **raw alphanumeric values**.

Information is created when data are processed, organized, or structured to provide context and meaning. Information is essentially **processed data**.

Knowledge is what we know. Knowledge is unique to each individual and is the accumulation of past experience and insight that shapes the lens by which we interpret, and assign meaning to, information. For knowledge to result in action, an individual must have the authority and capacity to make and implement a decision. Knowledge (and authority) are needed to produce **actionable information** that can lead to impact.

Metadata is defined as the data providing information about one or more aspects of the data; it is used to summarize basic information about data that can make...

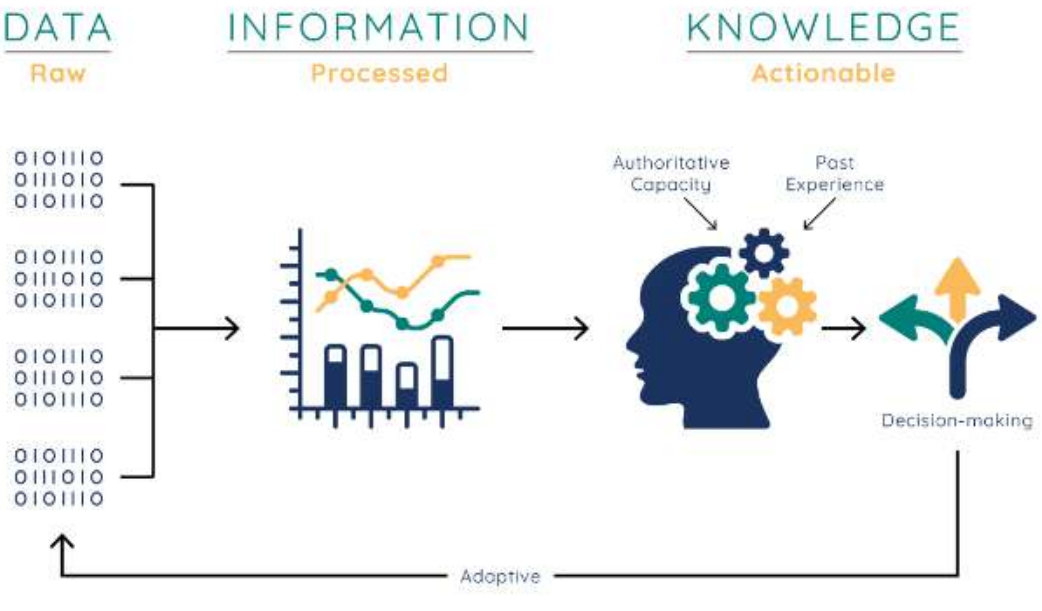
Metadata standard · Geospatial metadata · Metadata modeling · Metadata repositor

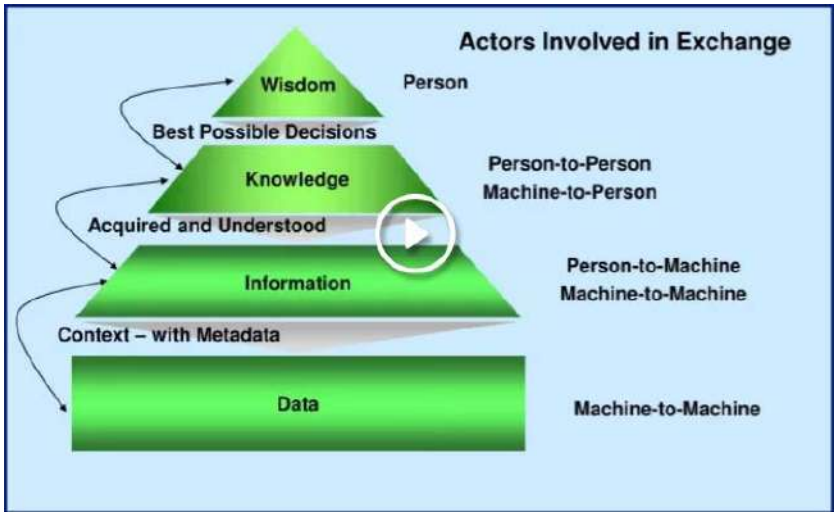
What is metadata ontology?

Meta-ontology is the study of the field of inquiry known as Ontology. The goal of meta-ontology is to clarify what ontology is about and how to interpret the meaning of ontological claims.

What is metadata knowledge?

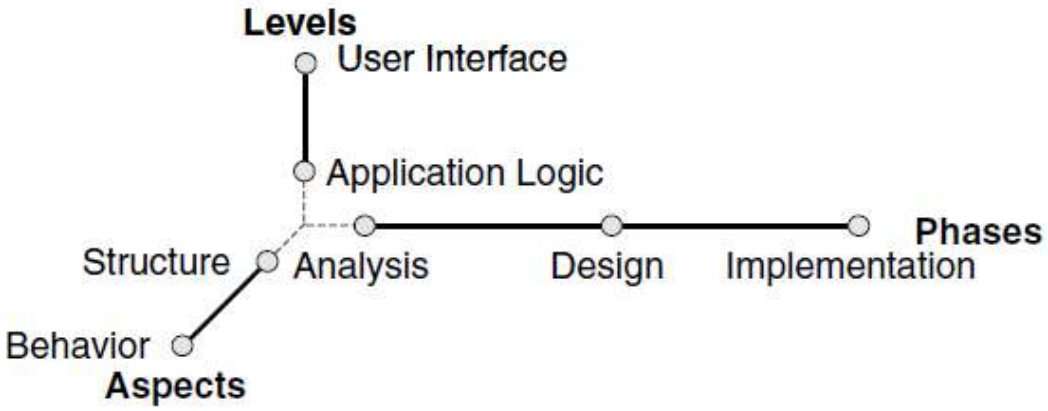
Metadata summarizes basic information about data, making finding & working with particular instances of data easier. Metadata can be created manually to be more accurate, or automatically and contain more basic information.




Actors Involved in Exchange
Wisdom
Person
Best Possible Decisions
Knowledge
Person-to-Person
Machine-to-Person
Acquired and Understood
Information
Person-to-Machine
Machine-to-Machine
Context – with Metadata
Data
Machine-to-Machine

Modeling Web Information Systems

Engineering disciplines have successfully used models to reduce complexity, document design
Decisions, and facilitate communication within project teams. Modeling is aimed at providing
A specification of a system to be built in a degree of detail sufficient for that system's
Implementation. The result of a modeling process is models representing the relevant aspects
Of the system in a simplified and – ideally – comprehensible manner



Modeling Specifics in Web Engineering

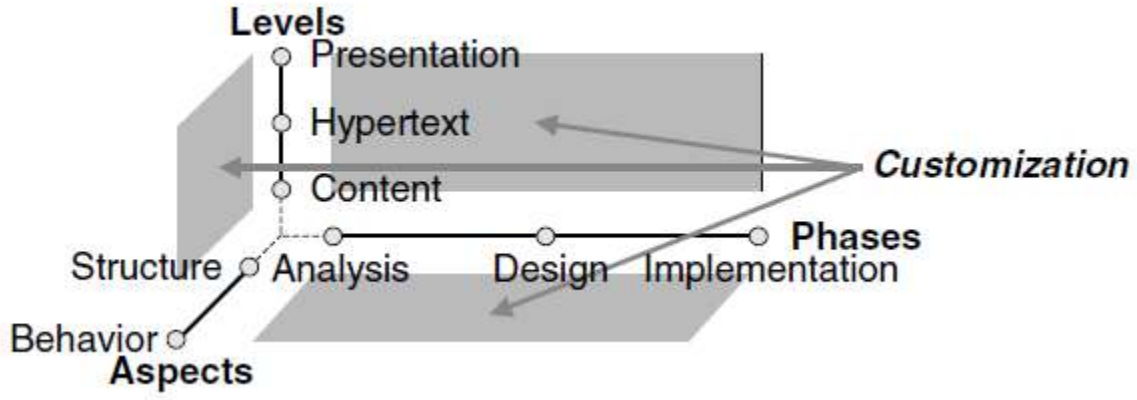
The tools of the trade in Web application modeling are basically not new, however, methods
To model traditional applications are not expressive enough for specific characteristics of Web
Applications For example, traditional modeling languages
Do not provide appropriate concepts for the specification of hyperlinks. This was the reason why

Special modeling approaches for Web applications have been developed during the past few
Years, which allow addressing a Web application in the three dimensions introduced above

Levels, aspects, and phases.

Levels

To model Web applications, the document-like character of its content as well as it's non-linear
Hypertext navigation has to be taken into account. This is the reason why we distinguish three
Levels when modeling Web applications, , in contrast to the two levels
Used in the modeling methods for traditional applications. The three levels are *content*, i.e., the
Information and application logics underneath the Web application, *hypertext*, i.e., the structuring
Of the content into nodes and links between these nodes, and the *presentation*, i.e., the user
Interface or page layout. Most methods which are used to model Web applications follow this
Separation into three levels



Despite a separation of concerns and the different objectives at the three levels, we would like to
Map the levels to one another. To achieve this mapping between levels, level inter-dependencies

Have to be captured explicitly. For example, different personalized hypertext access paths could
Be mapped onto one single content model. A comprehensive model of a Web application includes
All three levels discussed here, however, the emphasis can vary depending on the type of Web
Application. Web applications that provide a purely hypertext-oriented user interface to a large
Data set will probably require the modeling focus to be on content and hypertext structure.
In contrast, presentation-oriented Web applications, e.g., corporate portals or online shopping
Malls will most likely have larger demands on presentation modeling

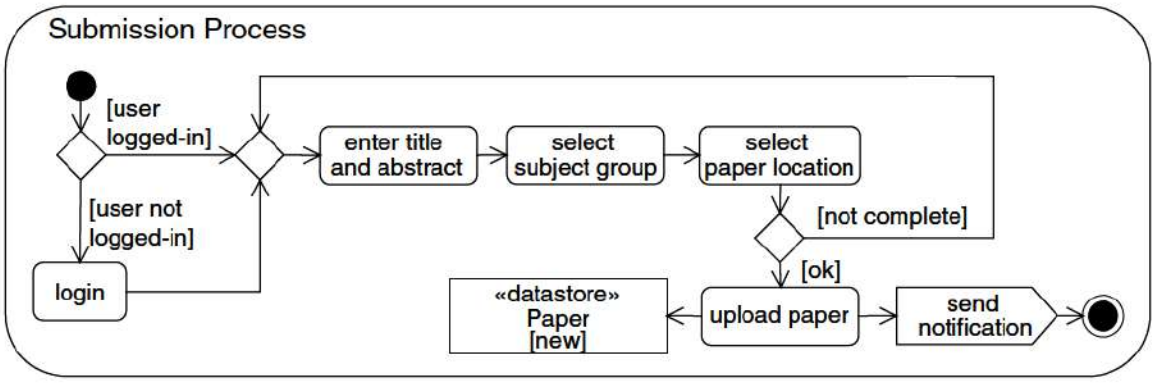
Aspects

Following the object-oriented principles, structure and behavior are modeled at each of the three
Levels, i.e. at content, hypertext and presentation. The relevance of the structure and behavior
Models depend on the type of Web application to be implemented. Web applications which
Make mainly static information available require less behavior modeling compared with highly
Interactive Web applications, such as for example e-commerce applications which provide
Search engines, purchase order functions, etc. With respect to mapping the different levels, it
Is recommended to use a uniform modeling formalism for structure and behavior, which might
Allow relying on one single CASE tool

Phases

There is no consensus in literature about a general modeling approach for the development
of Web applications (see also Chapter 10). In any case, the sequence of steps to model the
levels should be decided by the modeler. Depending on the type of Web application, it should

Be possible to pursue an information-driven approach, i.e., starting with content modeling, or
A presentation-driven approach, i.e., starting with modeling of the application's presentation
Aspects. Model-based development in Web engineering contradicts somewhat the often found
Practices in Web projects comprising, e.g., short-lived development cycles and the desire for
''agile methods''



Content Modeling

The information provided by a Web application is one of the most important factors for
The success of that application, not least due to the origins of the Web as information
Medium Modeling the content in the sense of pure data modeling is
Normally sufficient for static Web applications. Complex Web applications (according to the
Categorization additionally requires the modeling of behavioral aspects.
This means that content modeling includes the creation of the problem domain model, consisting
of static and dynamic aspects, as known from traditional Software Engineering. In addition the
Following Web application characteristics have to be taken into account:
• *Document-centric character and multimedia*: It is necessary to take all kinds of different
Media formats into account when modeling the content, including the structures the
Information is based on.

• *Integration of existing data and software*: Many Web applications build on existing
Data repositories and software components, which were not created for Web applications
Originally. Content modeling has to satisfy two potentially contradicting objectives, i.e., it
Should cover the content requirements of the Web application to the best possible extent,
And it should include existing data structures and software components.

Objectives

Content modeling is aimed at transferring the information and functional requirements determined
By requirements engineering to a model. The hypertext character of a Web application and the
Requirements of its presentation will not be considered in this effort.
Content modeling produces a model that comprises both the structural aspects of the content,
in the form of a class diagram, and, depending on the type of Web application, the behavioral
Aspects, e.g., in the form of state and interaction diagrams.

Hypertext Modeling

The non-linearity of hypertext is one of the most important properties to be taken into account
When modeling Web applications. Thus the hypertext structure has to be designed carefully. This
Can be achieved by using suitable access structures, i.e., navigation options, to avoid the risk of
Users getting lost and putting them under excessive cognitive stress

Hypertext Structure Modeling Concepts

Structural links connect elements of the same node, e.g., from a review summary to the
Review details.

• Perspective links put various views of a node in relation to each other, e.g., the PostScript
And the PDF versions of a paper.

Application links put different nodes in relation to each other, depending on the application,
e.g., a link pointing to ‘‘best paper’’.
Other classifications are based on the possible transport of information during navigation.
For example, the Web ML (Web Modeling Language) method (Ceri et al. 2003) specifies the
Following types of links:
• Contextual links carry context information, e.g., the unique number of a reviewer, to
Navigate from one reviewer to the reviews he or she created.
• Non-contextual links have no associated context information, e.g., links pointing from a
Single review to the list of all reviews.

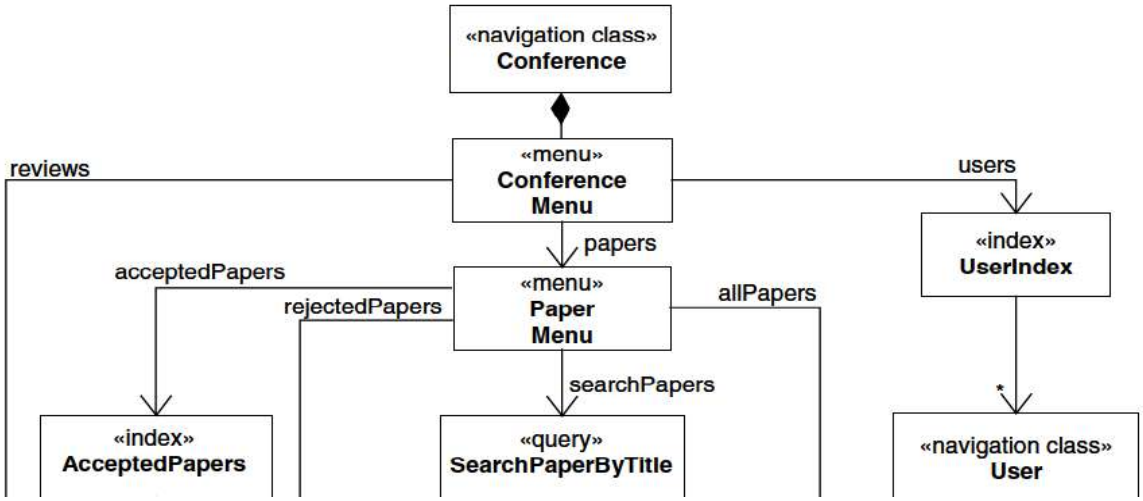
Web ML specifies additionally the following types of links:

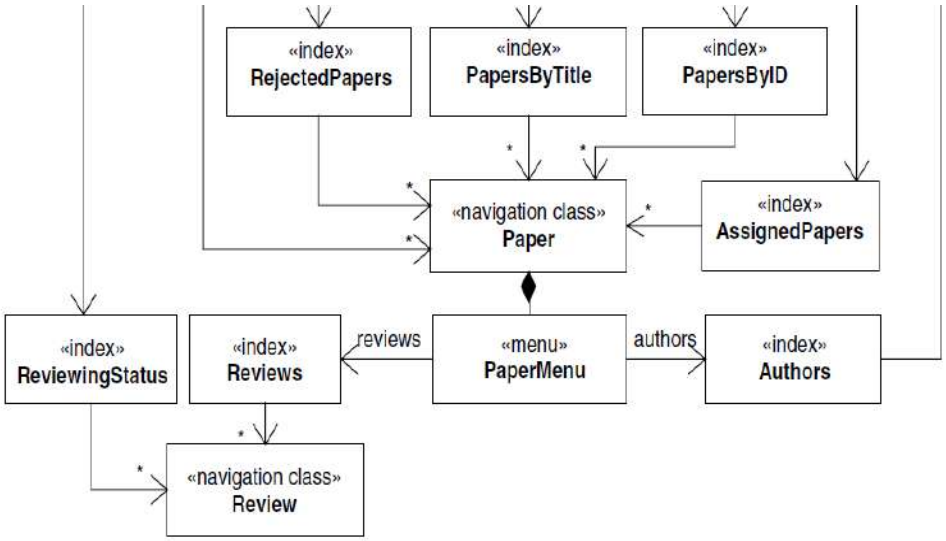
• *Intra-page links* are used when the source and the destination of a link belong to the same
Page, e.g., when a link allows the user to directly navigate to the summary of a paper,
Which is displayed further down on the page?
• *Inter-page links* are used when the source and the destination are on different pages, e.g.
When detailed information about the authors and their papers are on different pages

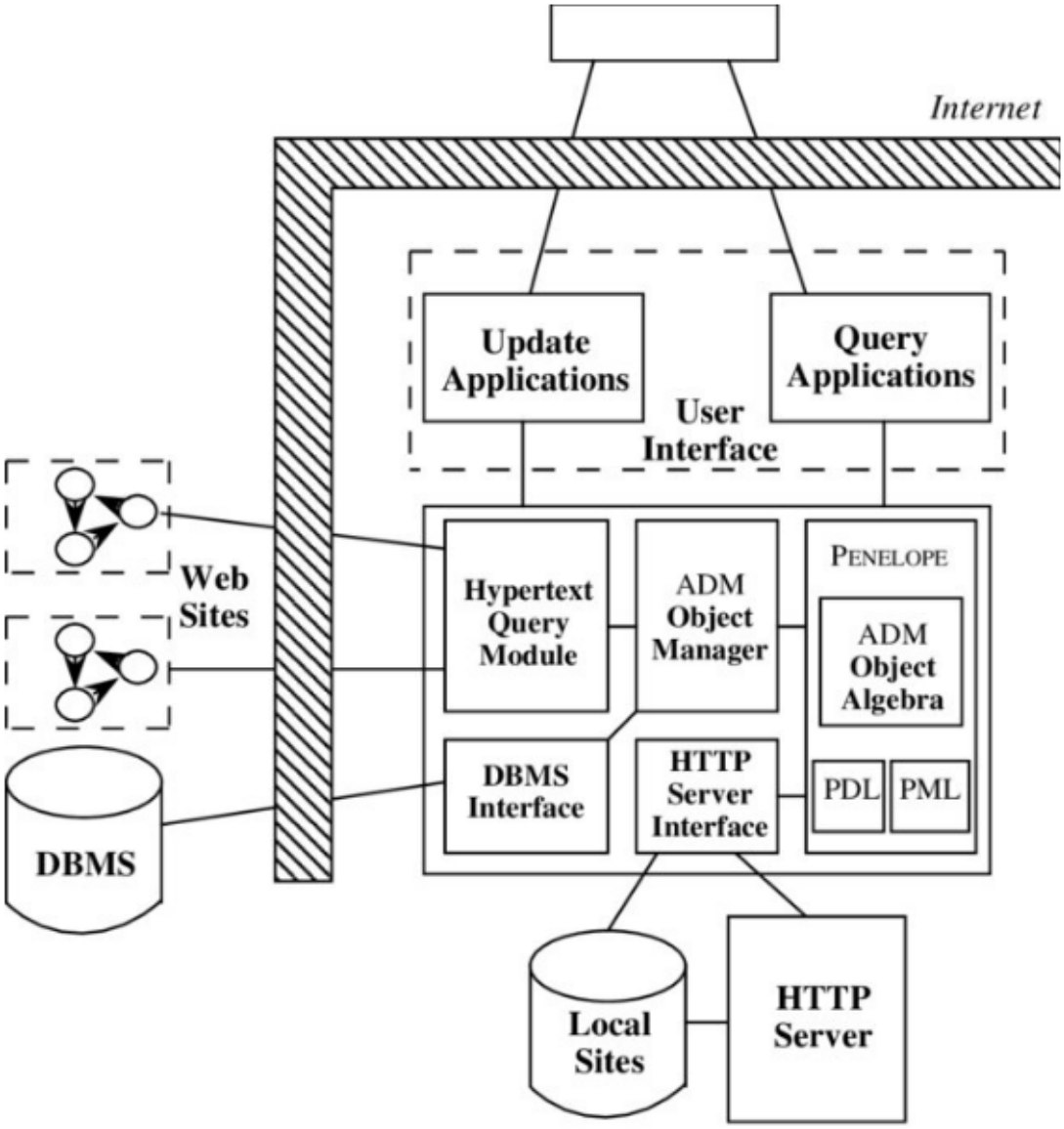
Based on the functional requirements of Web applications, the UWE

Modeling method defines the following types of links:

• Navigation links are used to navigate between nodes, e.g., links between papers and their
Authors.
• Process links point to the start node of a process, e.g., to the beginning of the review Submission.







WEB INFORMATION SYSTEM

Categories of Web Information Systems
The Characteristics of WIS
Abstraction Layers in WIS Modeling and Design

What are the categories of information systems?

It is useful (and common practice) to break them down into categories such as the following: Data-processing systems. Real-time systems. Decision support systems. Knowledge-based systems.

An Introduction to Information Systems

Information system refers to various information technology systems like computers, software, database, communication systems, the internet, devices, and others used by an organization to collect, transfer, organize, and store data. Bursting with changes, the current business milieu has helped companies implement a varied set of advanced technologies into different processes. These IT applications have introduced automation, efficiency, and timeliness in various business activities. The introduction of information systems into the business has evoked a chain reaction among different interrelated processes that have only benefited the companies by increasing profits and reducing costs and lead time, among other things. Therefore, it is imperative to understand the growing importance of information systems in companies.

What Is Information Systems And its Importance?

An information system is a group of data sets that ensures that business operates smoothly, embracing change, and helping companies achieve their goal. The dictionary defines an information system as a computer system or set of components for collecting, creating, storing, processing and distributing information. The information system is incomplete without the support of information technology (IT) **systems.**

An information system is not primarily associated with technology or IT system. Instead, it is related to how technology is used to fulfil the needs of- individuals, groups or organizations. In the digital era that we are in, the importance of

information systems is increasing because it standardizes the process of passing, collecting, storing, and accessing information or data for individuals or businesses.

There are different types of information systems that help individuals and companies to use the information to their benefit. In the succeeding part of the article, we will discuss various types of information systems in detail.

Six Types Of Information Systems And Their Application

Although many information systems offer various benefits, typically, businesses use these five applications in their company. Whichever information system you plan to implement in your business, here are the benefits it will offer:

- **It will induce innovation in business activities through its research and development.**
- **It will enable automation, reducing steps undertaken to complete a task.**
- **It helps keep the hardware, software, data storage, and networking system safe and up to date.**

Now that you know the changes information systems can bring about in an organization, let's look at the application that yields the power to change the business process- types of information systems.

What Are The Types Of Information Systems?

1. Knowledge Work System

There are different knowledge management systems that an organization implements to ensure a continuous flow of new and updated knowledge into the company and its processes. A knowledge work system (KWS) is one of the knowledge management systems that ease the integration of new information or knowledge into the business process.

Furthermore, KWS also offers support and resources to various knowledge creation techniques, artificial intelligence applications, and group collaboration systems for knowledge sharing, among others. It also uses graphics, visuals, etc., to disseminate new information. Below are some of the applications that work on the core fundamentals of KWS.

- **Designers often use computer-aided design systems (CAD) to automate their design process.**
- **Financial workstations are used to analyze huge amounts of financial data with the help of new technologies.**

- **Virtual reality systems are found in the scientific, education, and business fields for using graphics and different systems to present data.**

2. Management Information System

The management information system provides aid to managers by automating different processes that were initially done manually. Business activities like business performance tracking and analysis, making business decisions, making a business plan, and defining workflow. It also provides feedback to the managers by analyzing the roles and responsibilities.

A management information system is considered a significant application that helps managers immensely. Here are some of the advantages of the information system:

- **It enhances the efficiency and productivity of the company**
- **It provides a clear picture of the organization's performance**
- **It adds value to the existing products, introduces innovation and improves product development**
- **It assists in communication and planning for business processes**
- **It helps the organization provide a competitive advantage**

3. Decision Support System

A decision support system is an information system that analyses business data and other information related to the enterprise to offer automation in decision-making or problem-solving. A manager uses it in times of adversities arising during the operation of the business. Generally, the decision support system is used to collect information regarding revenue, sales figures or inventory. It is used across different industries, and the decision support system is a popular information system.

4. Office Automation System

An office automation system is an information system that automates different administrative processes like documenting, recording data, and office transactions, among others. The office automation system is divided into managerial and clerical activities. Here are some of the business activities that are done under this type of information system:

- **Email**
- **Voice mail**
- **Word processing**

5. Transaction Processing System

The transaction processing system automates the transaction collection, modification, and retrieval process. The peculiar characteristic of this type of

information system is that it increases the performance, reliability and consistency of business transactions. It helps businesses perform daily operations smoothly without hassle.
Once you are well-versed with different types of information systems, understanding the application of these systems becomes easy to comprehend. Therefore, in the last part of the article, we will look into applying information systems.

6. Executive Support System

an Executive Support System or ESS helps top-level executives to plan and control workflow and make business decisions. It is very similar to Management Information System or MIS.
Here are some of the unique characteristics of ESS:

- **It provides great telecommunication, better computing capabilities, and effective display options to executives.**
- **It enables them with information through static reports, graphs, and textual information on demand.**
- **It helps monitor performances, track competitors' strategies, and forecast future trends, among others.**

How To Apply Information Systems in Business?

Here are some of the business activities that require the intervention of an information system.

- **Enterprise resource planning (ERP)**

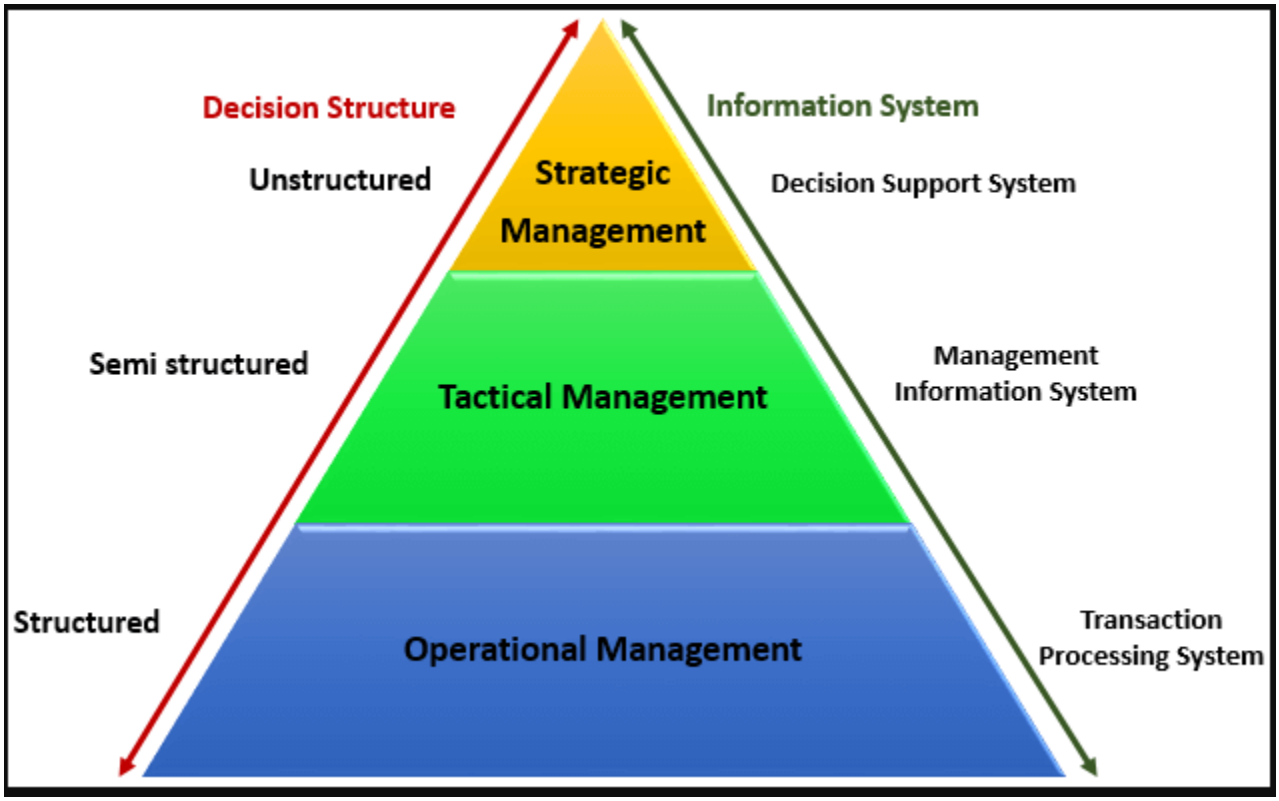
Applying information systems to enterprise resource planning helps automate business administration and planning functions.

- **Supply chain management (SCM)**

Information systems provide a common forum to connect with different parties in supply chain management. Moreover, it makes communication between parties easy and resourceful.

- **Customer relationship management (CRM)**

Many information systems help in realizing customer requirements. Furthermore, other information applications help companies interact with their audience easily and hassle-free.



The 6 most important characteristics of a good website include:

- **A clear purpose.**
- **Quality content.**
- **Outstanding UX and UI.**
- **Great performance.**
- **Multi-device optimization.**
- **High security.**

Characteristics of A Good Website

A website is a public application that usually targets any number of users. If you're building a website, you want to make sure you build something that is likeable and usable. In this article, I talk about top 10 characteristics a good website must possess.

Functional

A website is designed to serve a purpose, usually to solve a problem. For example, a job board has a purpose where employers can post jobs and job seekers can find and apply for jobs. Once applied, there should be a way for candidates and employers to communicate and keep up to date with a job application. If you build a job board and it only lets you post jobs, that's not enough.

Easiness

A website should be easy to use and navigate. When users visit your website, do they find the information they are looking for, without struggling? Does your website provide easy navigation to jump from one page to another and go back?

Relevant Content

You do not want to put content on a website that is not relevant to the users. You do not want to put car details on a job board. If your website is a job board, it should have content and tips related to being a good candidate such as how to create a professional resume and how to behave in an interview. You do not want to post about cars or sports on a job board.

Modern

You want to make sure your website is using current trends and technologies. You do not want to build a website that does not support today's needs. For example, responsiveness and fluid web designs are the key requirements for modern websites.

You do not want to create fixed page layouts for your website that doesn't support mobile readability.

Optimized

your website and its content should be optimized for different devices, browsers, data speed, search engines, and users. If your website is not optimized for mobile data users and their download speed, users may leave your website. If your website does not support common modern browsers, you may be missing out on some users. If your website is not optimized for popular smartphones and their sizes, users may not want to visit your website.

Responsive

Responsive web design is a modern need. A responsive website changes its layout and options to fit the device and browser size. For example, a website may provide more options on a large PC monitor compared to on a mobile phone.

Performance and Speed

Does your website load fast enough to show visitors what they are looking for? Your website can't be slow when presenting content to its visitors. If your website takes more than two seconds to load, it is probably too slow.

Reliable

Is your website reliable? If I apply for a job, does your website send me on-time notifications and messages to keep me posted with the updates? Can I access your website whenever and wherever? Is your website up and running 24/7?

Scalable

Do you have sufficient infrastructure to support your website and its visitors? A

website should be able to scale to a number of potential visitors. Not only should the website's user interface be scalable but the back-end database, APIs, and services too should be able to scale.

Secure

Security is a major concern to today's web visitors. You need to make sure that your website follows industry standards and guidelines. For example, if you have user registration feature, you want to make sure that user passwords are encrypted and not displayed in plain text. You also want to ensure that the website is secure and uses SSL encryption. If you've forgotten a password feature, you do not want to provide a reset password feature without asking the security questions to the user.

WIS Modeling and Design

- **Strategic Layer**
- **Business Layer**

What is the definition of strategic web design?

Strategic design is the fusion of your organizational goals with every aspect of your design process. You aren't simply designing a user interface that looks good and is usable and accessible.

Steps to Create a Website Strategy

What is a website strategy?

A website strategy is essentially a long-term business plan that's built around your website. Your website strategy goes beyond the design you choose for your website—it helps inform the content you create, your marketing efforts, and more.

Typically, a website strategy includes several parts, including your website design, branding strategy, keyword research and implementation, and analytics. All of these individual parts work together to help you design and develop an effective website, making it easier to optimize your website in a way that's aligned with your goals. Your website strategy can also help with the basics of branding, including your brand's personality, appearance, and purpose.

How to develop your website strategy in 10 steps

With a little time and effort, you can create a website strategy to help you build the best website for your brand. A content strategy guide can even help you understand how to use content to optimize your website. In the meantime, here's how you can start creating a website strategy.

1. Set your goals

No good plan is complete without goals, so take some time to set a handful of goals for your website. Perhaps you want to sell more products or services, or maybe you want to increase the sales of a particular product. You might also set a goal to get a specific number of unique visitors each month or to convert a certain percentage of visitors to leads. The goals you set will determine what you're working toward with your site strategy, so setting clear goals is important.

2. Find a story to tell

Every brand has a story to tell, so find your story and figure out how you want to tell it. If you want to tell a story for branding purposes, find a good, honest narrative and make sure you keep it simple. The stories you find in Mailchimp's Courier magazine are a great example of effective brand storytelling.

3. Take an inventory of your content

Before you launch your website, figure out what kind of content you have to offer. You've probably got a basic story about your brand, but do you have any videos, testimonies, or blog posts that you're ready to share? Content is essential if you want to drive more traffic to your website and convert website visitors into customers. If you're using content that was written or created previously but not published, you can always add some relevant keywords to help increase your search engine rankings.

4. Identify your target audience

One thing you need to do is decide on the right target audience for your website. Every business targets a different specific demographic, so it's smart to figure out who uses your products or services. The easiest way to do this is to look at any existing data you may have about your customers, including their age, location, and other information that allows you to segment your target audience. If you don't have any existing data to pull from, you can send a survey to your customers to determine your target audience. Keep in mind that you might need to offer an incentive to collect enough data to understand your target demographic.

5. Plan your website

Once you've figured out the basics, take some time to plan out your website. There are a lot of decisions to make when you're building a website, including choosing between a static and dynamic website,

deciding on a responsive or adaptive design, and deciding on a budget for your project.
During this planning process, you can also work with a web designer to start mocking up a rough draft of your website. You don't have to keep your website design the same from start to finish, but having a plan to build off can be a big help. This also gives you a chance to tailor a website strategy around your website, your customers, and your needs

What is strategic level?

At a strategic level, the management of performance offers the opportunity to ensure that the contributions of each of the various elements effectively support and are aligned with the organisation's overall aims and objectives.

What is Web Application Architecture?

So, what is a web application and how is it different from a website?
The basic definition of a web application is a program that runs on a browser. It's not a website, but the line between the two is fuzzy. To differentiate a web application from a website, remember these three formal characteristics. A web application:

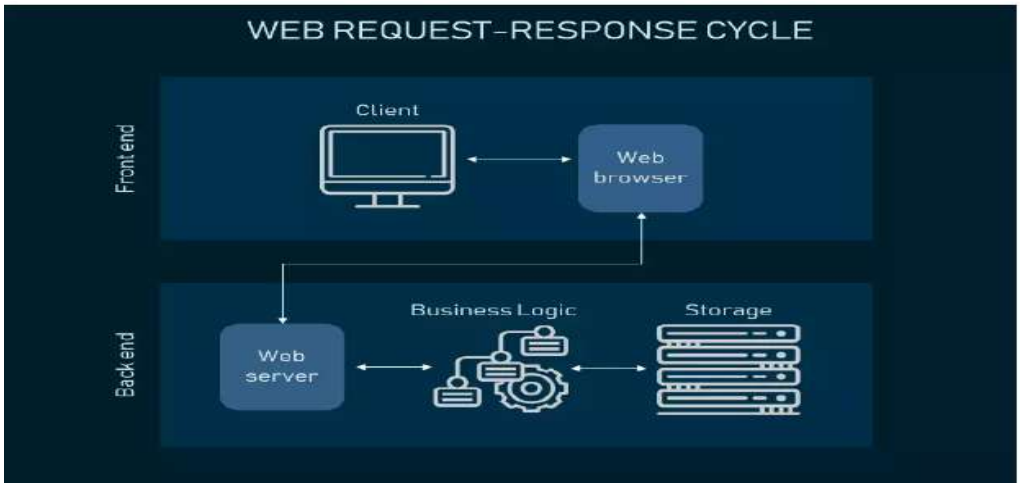
- **addresses a particular problem, even if it's simply finding some information**
- **is as interactive as a desktop application**
- **has a Content Management System**

A website is traditionally understood to simply be a combination of static pages. But today, most websites consist of both static and dynamic pages, which makes almost all modern websites – you guessed it! – web applications. In this article, we will use the terms interchangeably.
Your computer, or smartphone, or any other device you're browsing with is called a client. The other half of the web equation is called a server because it serves you the data you request. Their communication is called a client-server model, whose main concern is receiving your request and delivering the response back.

Web application architecture is a mechanism that determines how application components communicate with each other. Or, in other words, the way the client and the server are connected is established by web application architecture.
Web applications of different sizes and complexity levels all follow the same architectural principle, but details may differ. We will further explain how a basic request-response process works and what components comprise the architecture.

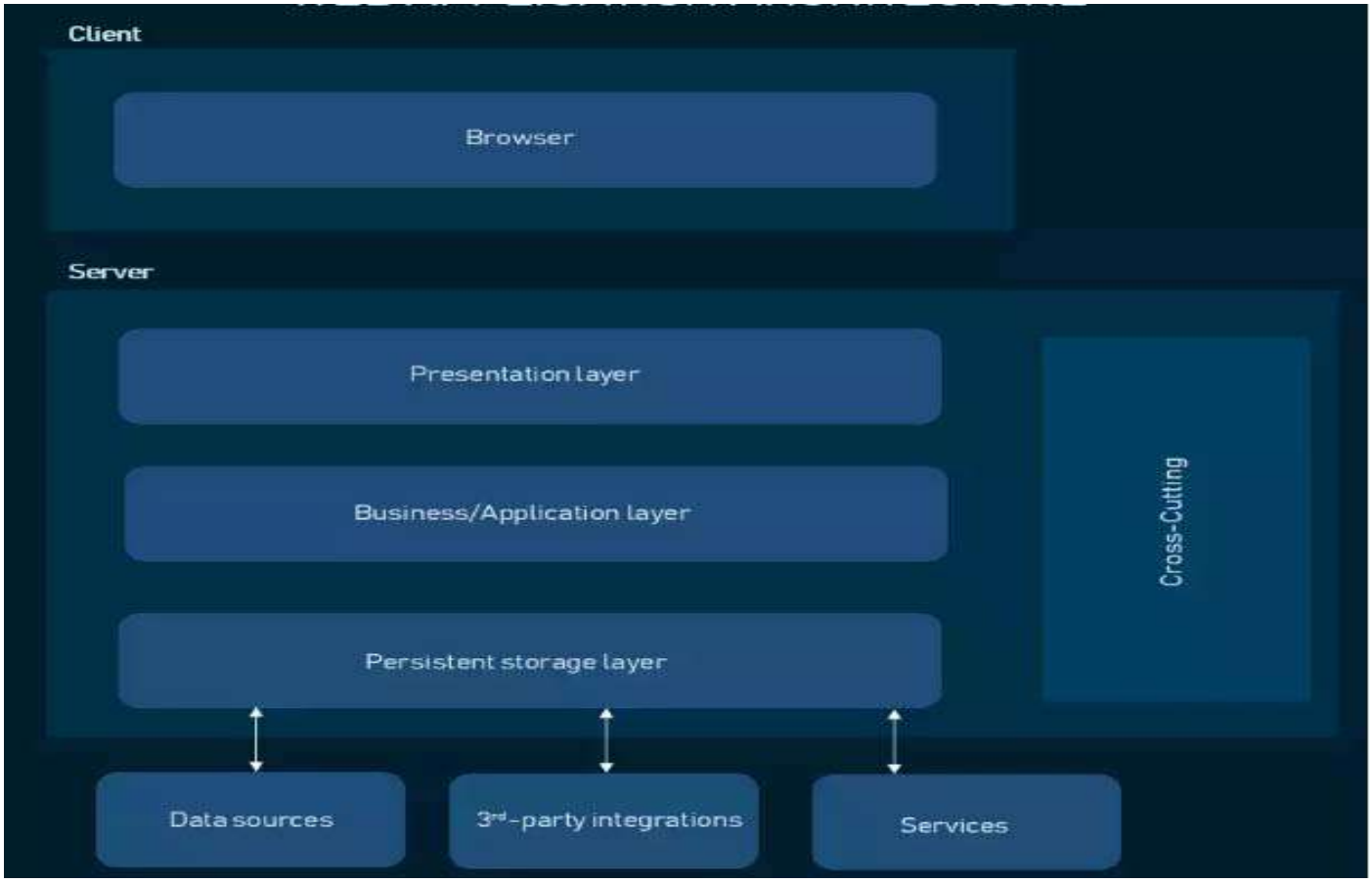
How does the web request work?

To understand the components of web application architecture, we need to understand how they are used in performing the most basic action – receiving and responding to a web request.



Web application architecture components and Three-Tier Architecture

Most web applications are developed by separating its main functions into layers, or tiers. This allows you to easily replace and upgrade each layer independently. This architectural pattern is called Multi- or Three-Tier Architecture.



Presentation layer

The presentation layer is accessible to users via a browser and consists of user interface components and UI process components that support interaction with the system. It's developed using three core technologies: HTML, CSS, and JavaScript. While HTML is the code that determines what your website will contain, CSS controls how it will look. JavaScript and its frameworks make your website interactive – responsive to a user's actions. Developers use JavaScript frameworks such as Angular and React to make the content on the page dynamic.

Business layer

This layer, also called Business Logic or Domain Logic or Application Layer, accepts user requests from the browser, processes them, and determines the routes through which the data will be accessed. The workflows by which the data and requests travel through the back end are encoded in a business layer.

For example, if your application is a hotel booking website, business logic will be responsible for the sequence of events a traveler will go through when booking a room.
Although *business rules* can be a manifestation of the business logic, they are not the same. Sometimes business rules are extracted and managed separately, using a Business Rules Management System, as we discussed in our article on back office systems.

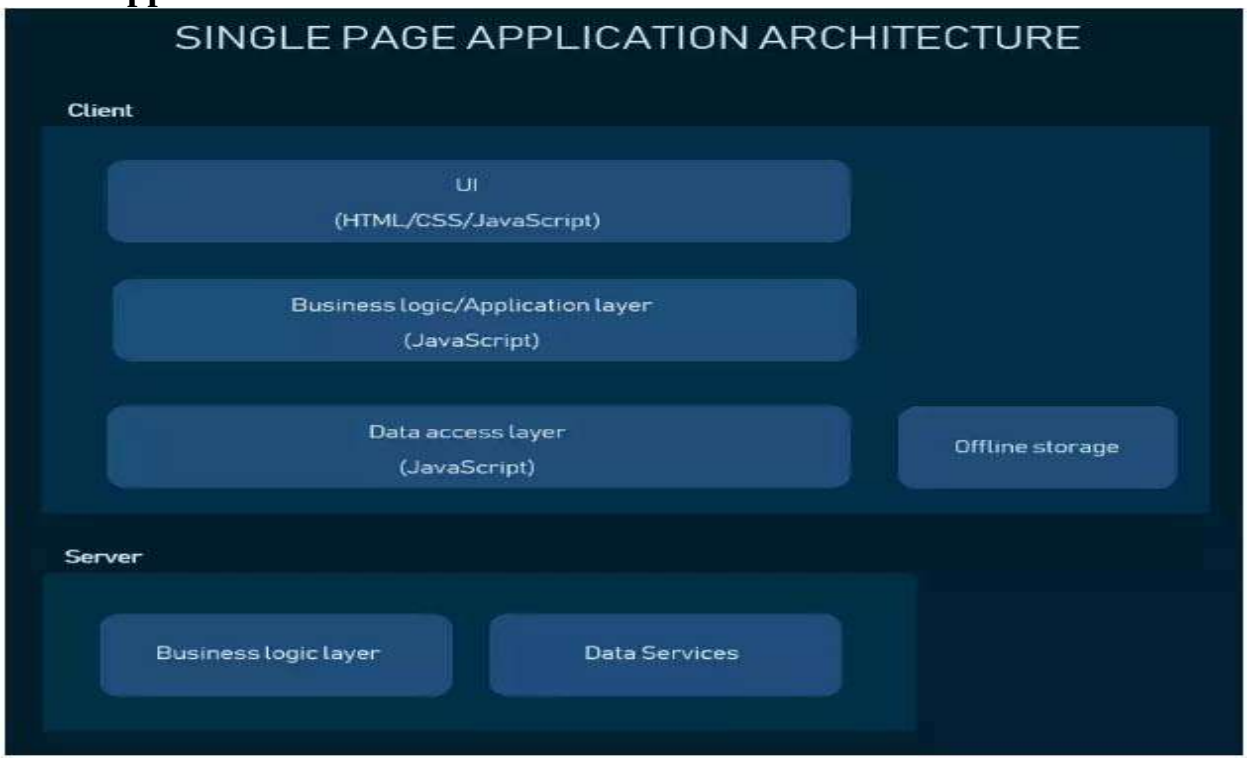
Persistence layer

Also called the storage or data access layer, the persistance layer is a centralized location that receives all data calls and provides access to the persistent storage of an application. The persistence layer is closely connected to the business layer, so the logic knows which database to talk to and the data retrieving process is more optimized.
The data storage infrastructure includes a server and a Database Management System, software to communicate with the database itself, applications, and user interfaces to obtain data and parse it. Typically you can store your data either in owned hardware servers or in the cloud – meaning, that you purchase data center management and maintenance services while accessing your storage virtually. Using the services of cloud technology providers such as Amazon, Google, or Microsoft, you can utilize Infrastructure-as-a-Service, Platform-as-a-Service, or serverless approaches to cloud management.
There are also components that usually exist in all web applications but are separated from the main layers:
Cross-cutting code. **This component handles other application concerns such as communications, operational management, and security. It affects all parts of the system but should never mix with them.**

Single Page Applications

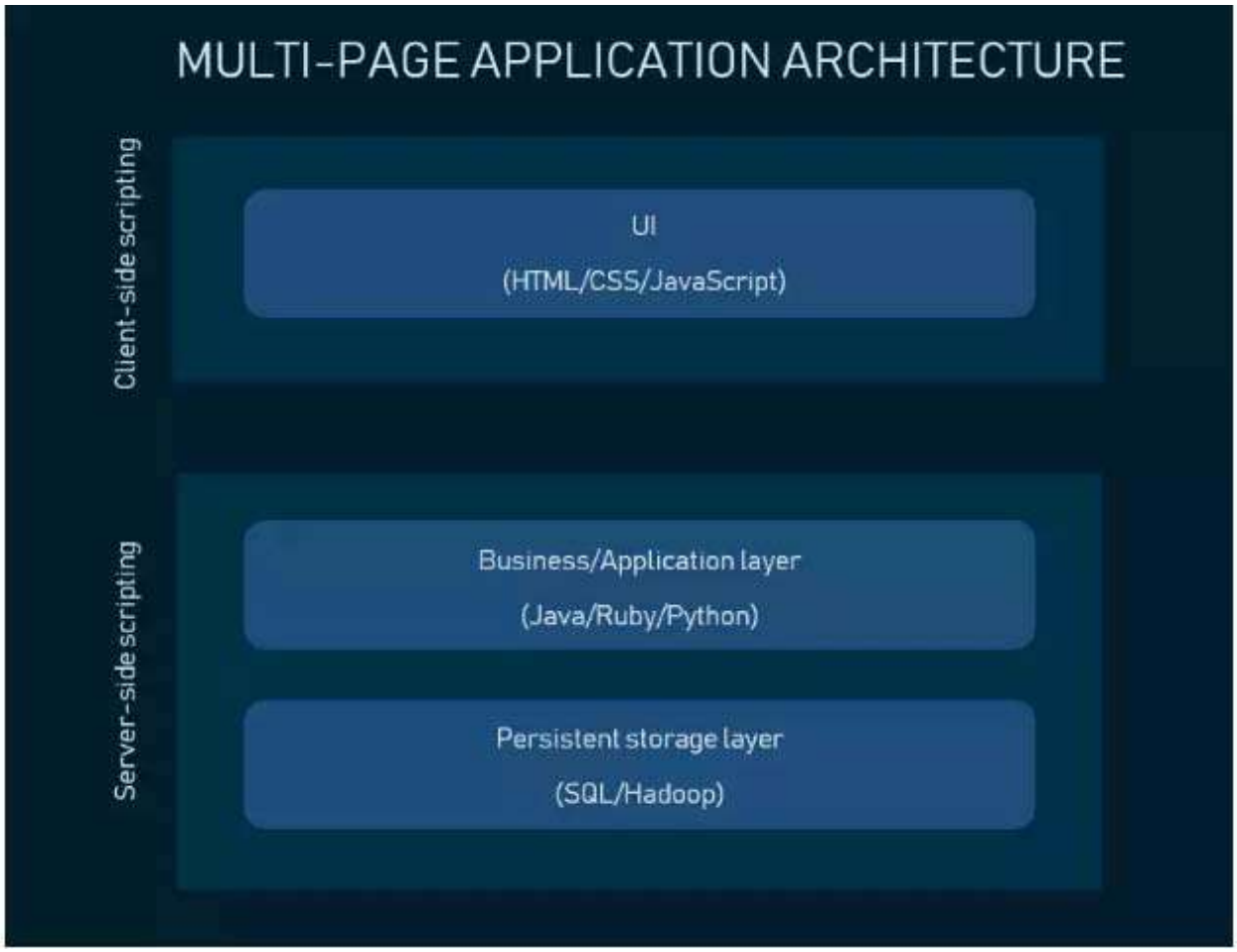
The main purpose of SPAs is the ability to access all information from a single HTML page. Having moved the application logic to the client-side and using server-side only as data storage, developers can make the website run faster and ease the load off the server. The front end, aside from HTML and CSS, is written on a single framework, which dynamically generates content and transmits it to a user (think of a Facebook feed or your Gmail). Dependencies between components are tight. This means that making changes to one of the UX or UI elements necessitates rewriting the whole front end code.

Since SPAs move the logic to the client-side, they have to be written using client-side scripting. If you're using *client-side scripting technologies*, you're basically building templates, so when a user requests content, a server simply transmits this data back to the browser, which renders it according to the templates. This significantly reduces the server load, as opposed to server-side scripting. The core technology of client-side scripting is JavaScript. Along with its many frameworks, this language allows creation of both small and robust applications.



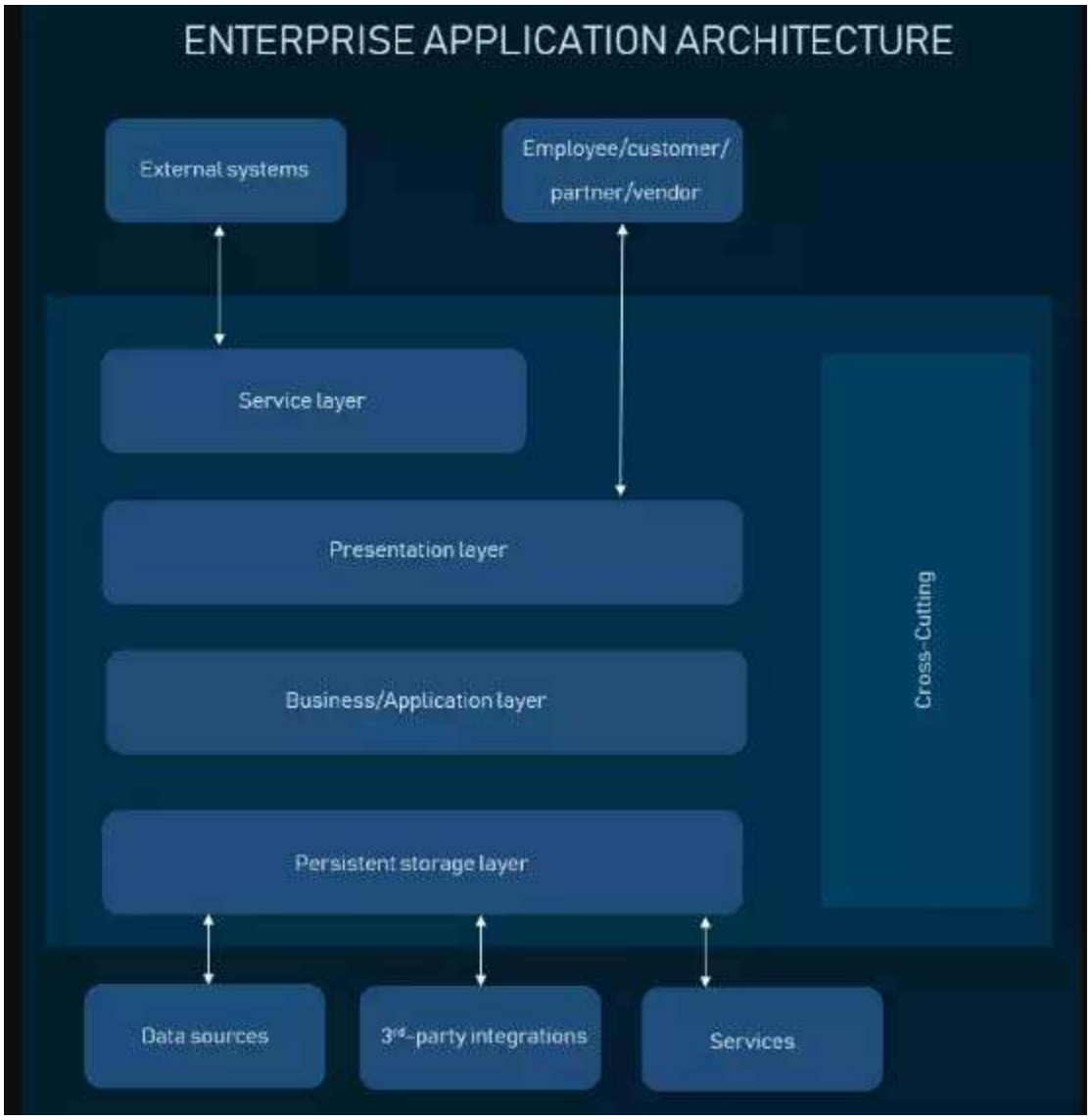
Multi-Page Applications

In Multi-Page Applications, some content requests need a whole new web page to be retrieved from the server. These are massive applications with multi-layered UI. AJAX technology solves the difficulties of complex applications transferring a huge amount of data between server and browser, refreshing only selective elements of the application. At the same time, the given approach brings more complexity to the table being more difficult to develop as compared to that of the SPA.



The two main distinctions enterprise application architecture has from a regular web application is the addition of another layer to the classic pattern – the service layer.

The service layer is another abstraction between Presentation and Business Logic. It's an integration gateway that allows other software to access your business logic and resources without interacting with those resources directly. It works by passing messages through a separate interface and works.

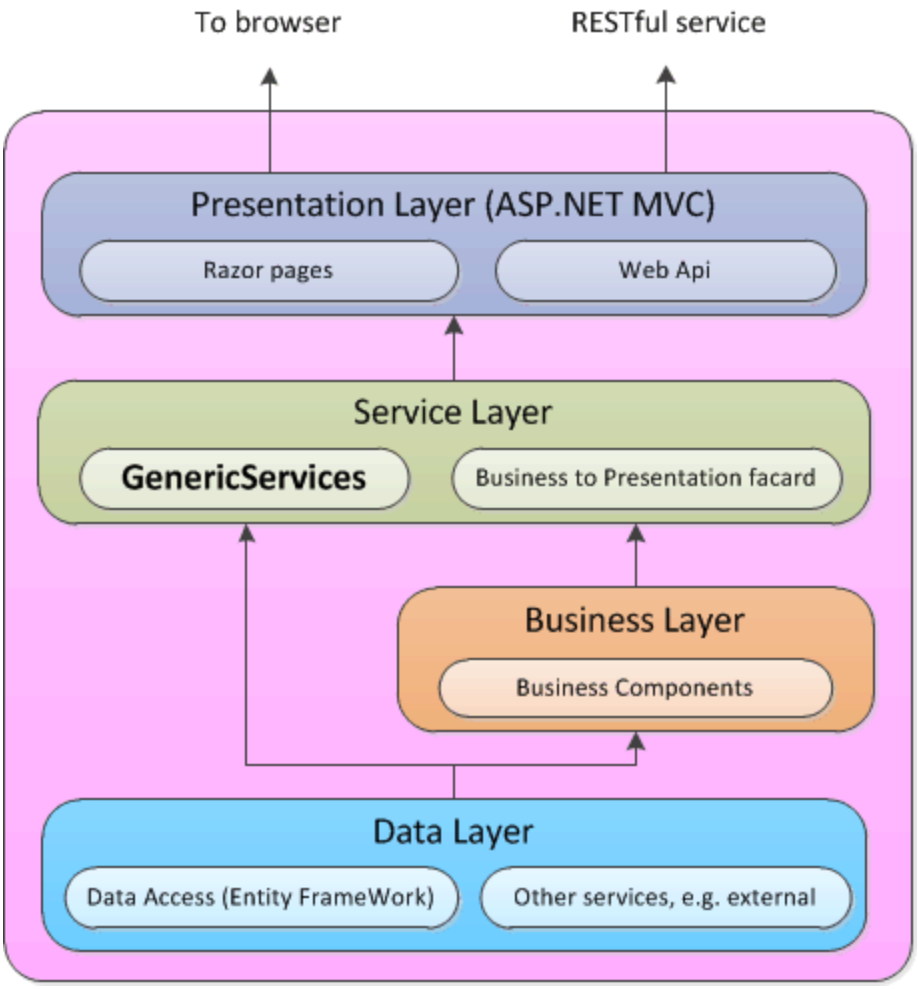

ENTERPRISE APPLICATION ARCHITECTURE
External systems
Employee/customer/
partner/vendor
Service layer
Presentation layer
Business/Application layer
Persistent storage layer
Cross-Cutting
Data sources
3rd-party integrations
Services

What is the business layer?

The business logic layer contains objects that execute the business functions. The Command pattern should be considered to implement these objects. With the Command pattern, each use case in the requirements document is implemented as a separate command or set of commands executed in the business logic layer.

What is business layer example?

Example: In an application, when the user access it or write queries in it with the help of a presentation or user interface layer the business logic layer helps the user to get a response to the asked queries by transferring it to the Data Access layer which further processes the query and give the suitable result to ...



Architecture of Business Layer working with Entity Framework

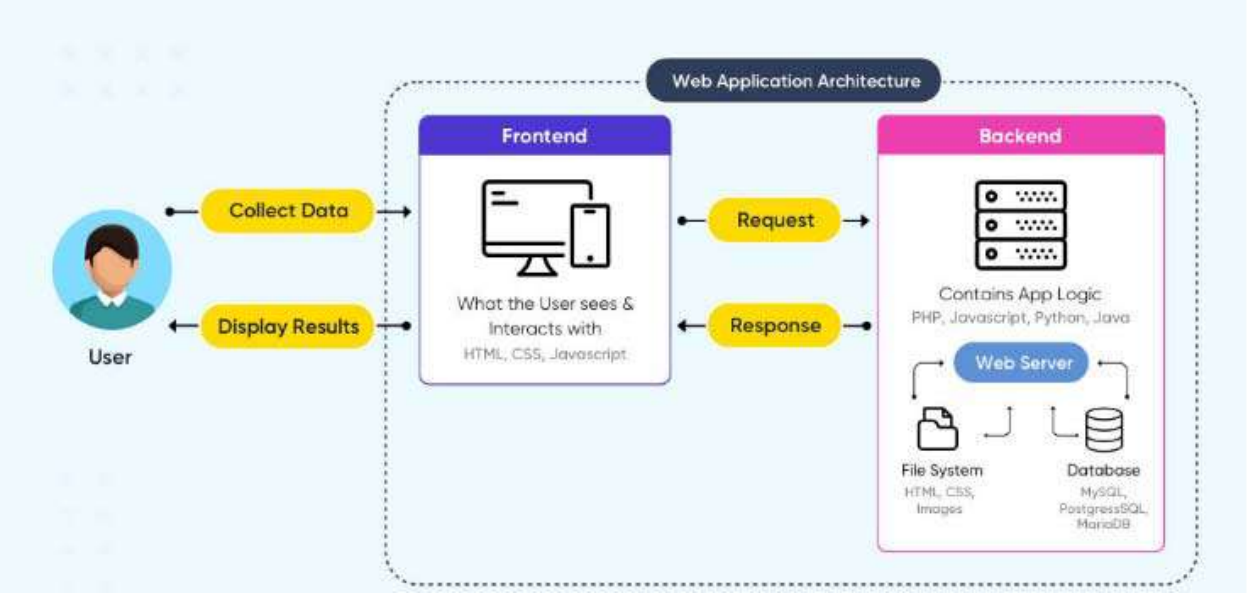
- **The Business Layer does not have to handle database validation errors, which can occur when 'SaveChanges' is called. The Service Layer does that, which has more information available to it on what is going on.**
- **I know that all the data will be saved in one transaction. It is either there or its not.**
- **If I need to chain multiple business methods I can use a transaction scope to rollback changes if one of the later methods fails.**
- **The Service Layer can check other things to decide whether the data should be saved. I use this to allow users to save/not save on warnings.**

- **It makes the Unit Testing of the Business Logic much easier as the database has not been changed. Most of my Business Logic has a property holding the data structures about to be written, so I can just check that.**

Conceptual Layer
Presentation Layer
Implementation Layer

Web Application Architecture

What is Web Application Architecture



Web application architecture defines the high-level components of a web system and the interactions between those components, decisions which impact how the web app performs and scales.

At the most basic level, the web application architecture defines the front-end (what the user sees) and the back-end (what powers the app on the server side) and how all those interactions take place (note the arrows for the flow of information):

Why Does Web Architecture Matter

Today's consumers bring high expectations to their online and mobile experiences, influenced by how useful, usable, desirable, accessible, credible, findable and valuable they find the interactions, as expressed by the user experience honeycomb. While users cannot see the web app architecture, decisions made at the architecture level can impact the ultimate experience the user receive.

The web app architecture is made up of several application components (or layers) and their interactions, choices that have a direct impact on the efficiency of the overall web app as well as its security, user experience, and ability to scale.

Web application defines how the app will ultimately function, including:

- **Authentication**
- **Security**
- **Speed and reliability**
- **Scaling**
- **Automation**
- **Error logging**
- **Analytics and testing**

- **User experience**

Further, questions at the web architecture level will define how much the web app will cost to develop and maintain, the availability of IT skills, and the overall time-to-market.

How Web Application Architecture Works

In a web application, you can think of two separate programs (buckets of code) that run in parallel as follows:

- **Client-side code**
  The code in the front-end (HTML, CSS, JavaScript) that responds to user inputs and creates the visual representation (user interface or UI)./li>
- **Back-end code**
  The back-end or server-side code has two jobs: one, to respond to HTTP requests and return responses to the client and two, to store different types of data. Server-side code can be one of a myriad of languages including C#, Java, Python, Ruby, Node.js, PHP, etc.

In web application architecture, the developers (or architects) must decide what the code on the front-end and back-end should do in relation to each other. However, these "code" decisions are very high-level (what happens and where). Web application architecture design is not about making code level decisions – that is a part of software design.

Web Application Architecture Components

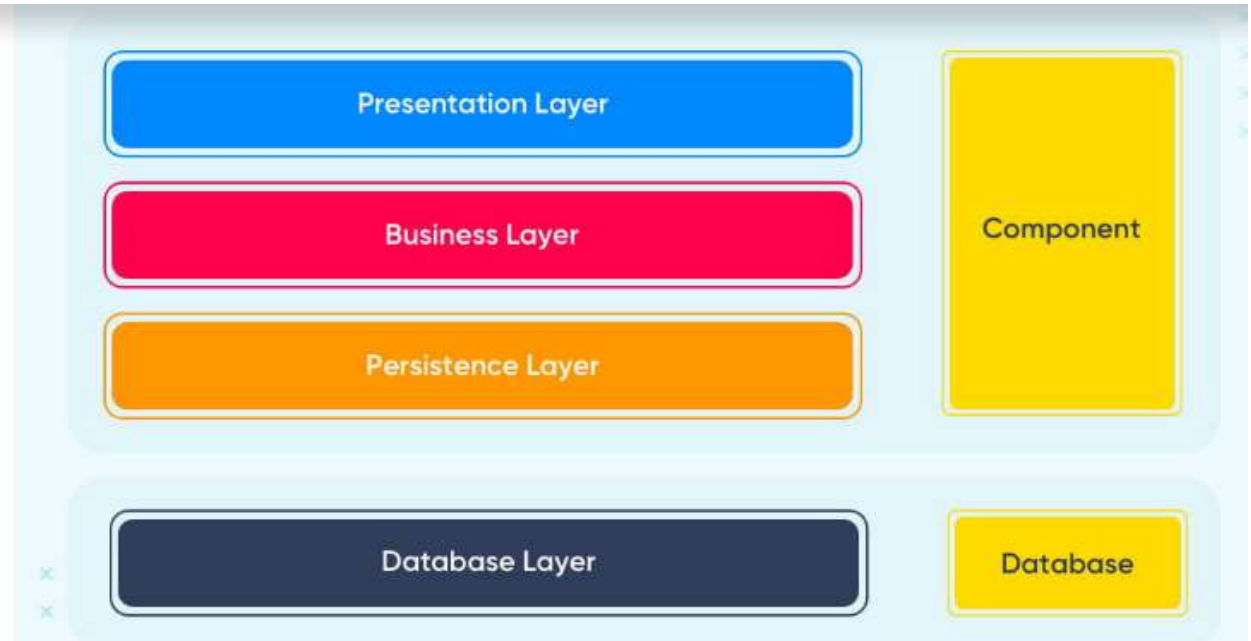
Web application architecture is can be simplified into two broad categories of components:

- **User Interface (UI) Components**
  These application components are any that impact the UI or the user

experience (UX) including dashboards, notifications, activity logs, settings, layouts, and design.

- **Structural Web Components**
  The structural components are hidden from the end-user and include the server, logic or workings of the web app, and the database.

Layers of Web App Architecture



Web application architecture is made up of four specific layers that each contain the components of the web app, each layer performing a specific function. The four layers of web architecture are:

- **Presentation Layer**
  The presentation layer is user-centric: what the user sees and interacts with. Also known as the client-side, the presentation layer includes the specific user interface (UI) needed for the display at hand and the logic

responsible for browser communication and any code to support user interaction.

- **Business layer**
  The business layer represents the logic of the app, getting data from the persistence layer and executing the request back to the presentation layer following the rules of the business operations – for this reason, this data exchange layer is sometimes also called the "logic layer" or the "application layer."
- **Persistence layer**
  The persistence layer is another logic level, but here the logic is concerned only with reading, writing, or manipulating the database. Like other layers, the persistence layer can go by a more obvious name: a data access layer.
- **Database layer**
  The database layer is where all the data lives in the database, e.g. SQL Server or MongoDB.

In a layered architecture, changes that happen to one layer don't impact other layers, reducing interdependencies and making for a more resilient app. On the flip side, for a simple app, the presence of layers can increase the code requirements and lead to a more monolithic application.

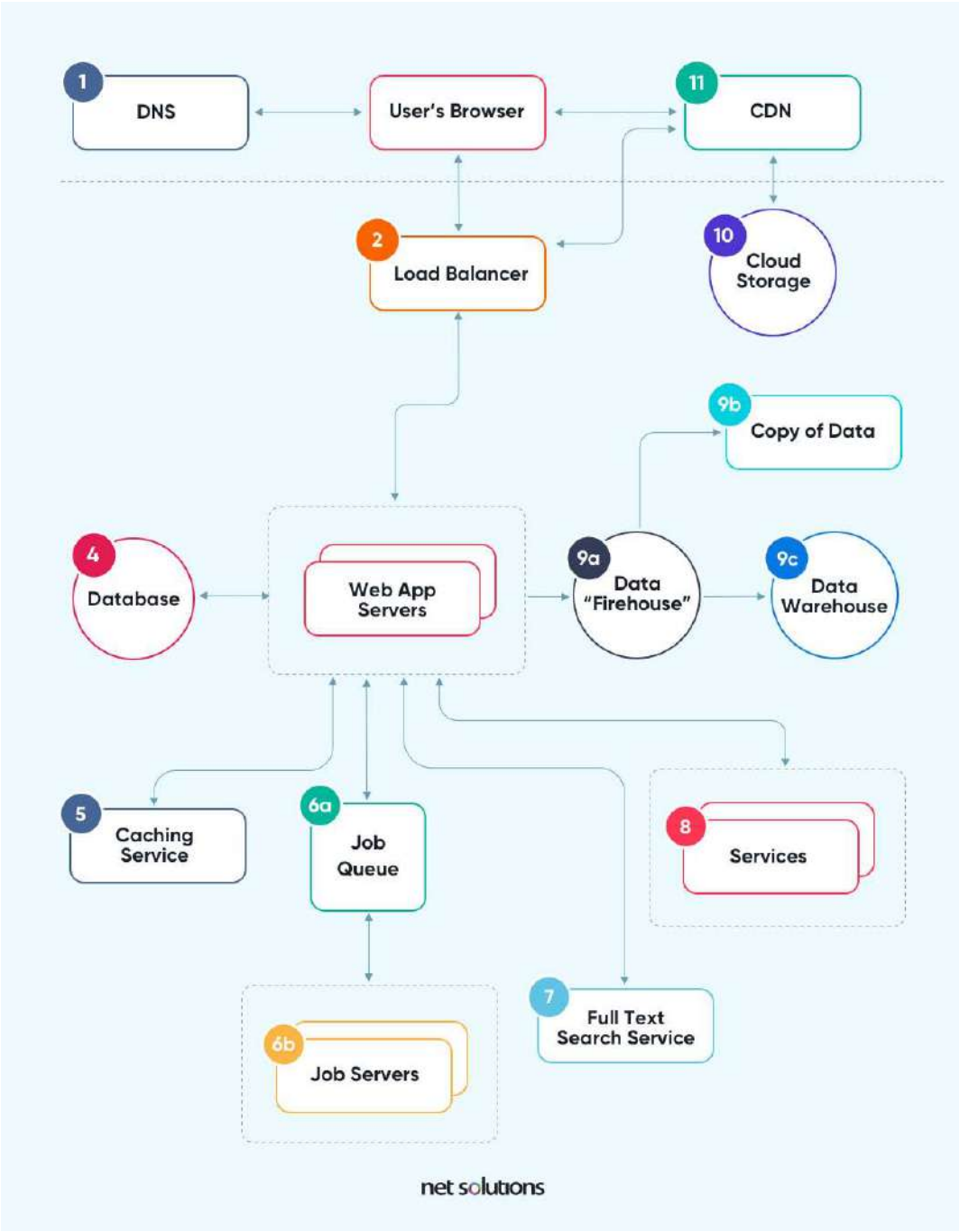
Best Practices for Web Application Architecture

While this guide has outlined the various components and layers of web application architecture, there are in fact many different approaches one can take. After all, one can say that a house has four walls, a floor and a roof, but that does not answer questions about how many rooms it has, their shape or their size. Web application architecture draws up the blueprint that helps define the overall shape of the web app.

To answer the question of "what is the best architecture for web applications?," consider the following best practices:

- **Is as simple as possible and easy to understand**
- **Scales easily (both overall and individual components)**
- **Supports flexibility**
- **Minimizes crashes (supports stability) and logs errors**
- **Supports automated or continuous deployment**
- **Supports strong analytics and testing**
- **Leverages strong security standards**
- **Leverages well-structured, understandable code**
- **Can be updated easily and independently**
- **Take advantage of new technology that offers clear and tangible benefits (isn't just a fad), such as single-page applications and progressive web apps.**

Web Application Architecture Diagram



Earlier, we demonstrated a simplified version of a web app architecture diagram explaining how the client and server communicate. This can further be broken down into all the components of web application architecture as follows:

Models of Web Application Components

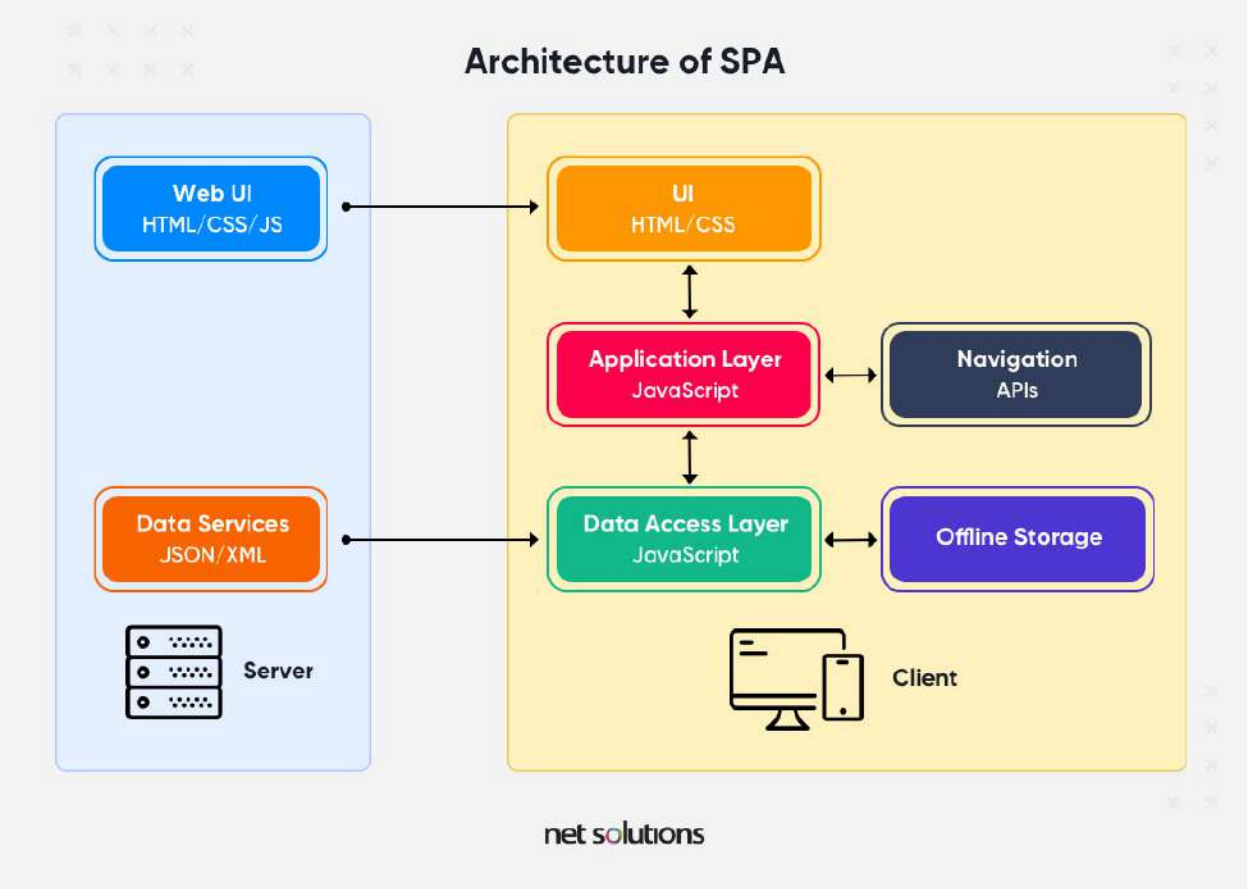
Regardless of the overall architecture, every web app will have at least one server and one database. However, there are reasons when an app may need more of one or both of them, a process of further defining the components of the web app. The most common approaches include:

- **One Web Server, One Database**
  In this approach, there is just one server and one database. If the server fails, the entire web app goes down; if the database fails, you lose all your data. This approach is suitable only for a very small scale or personal web app.
- **Multiple Web Servers, One Database**
  In this approach, the app is more risk-proof against a server crash, allowing one server to take the place of another if it fails or to scale as needed (add more load), but data loss remains a risk and could eventually become a bottleneck to scaling.
- **Multiple Web Server, Multiple Databases**
  Redundancy is key here, with at least 2 servers and at least 2 databases, providing a backup for all systems and allowing work to be distributed across different databases. However, when it comes to multiple databases, new risks emerge about data integrity and restore integrity as well as added complexity around access.
- **Application Services**
  This approach is known as "serverless," taking advantage of distributed SaaS solutions that take care of all the questions around servers and

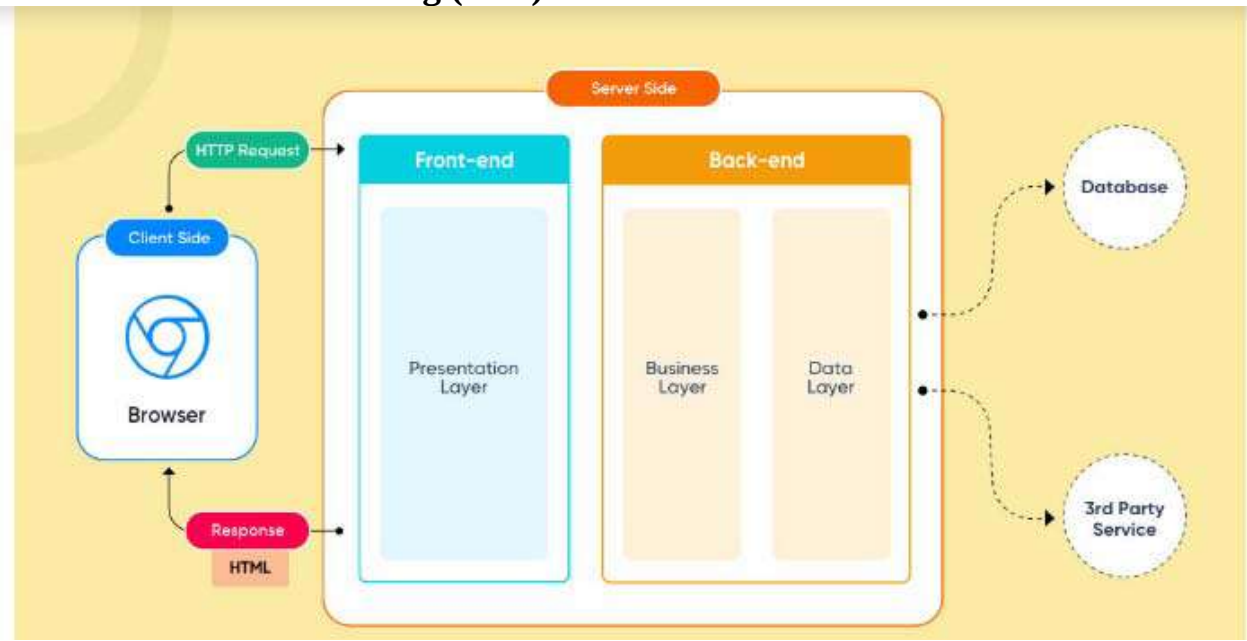
databases – all the configurations, tasks, maintenance, or scaling questions as they arise.

Types of Web Application Architecture

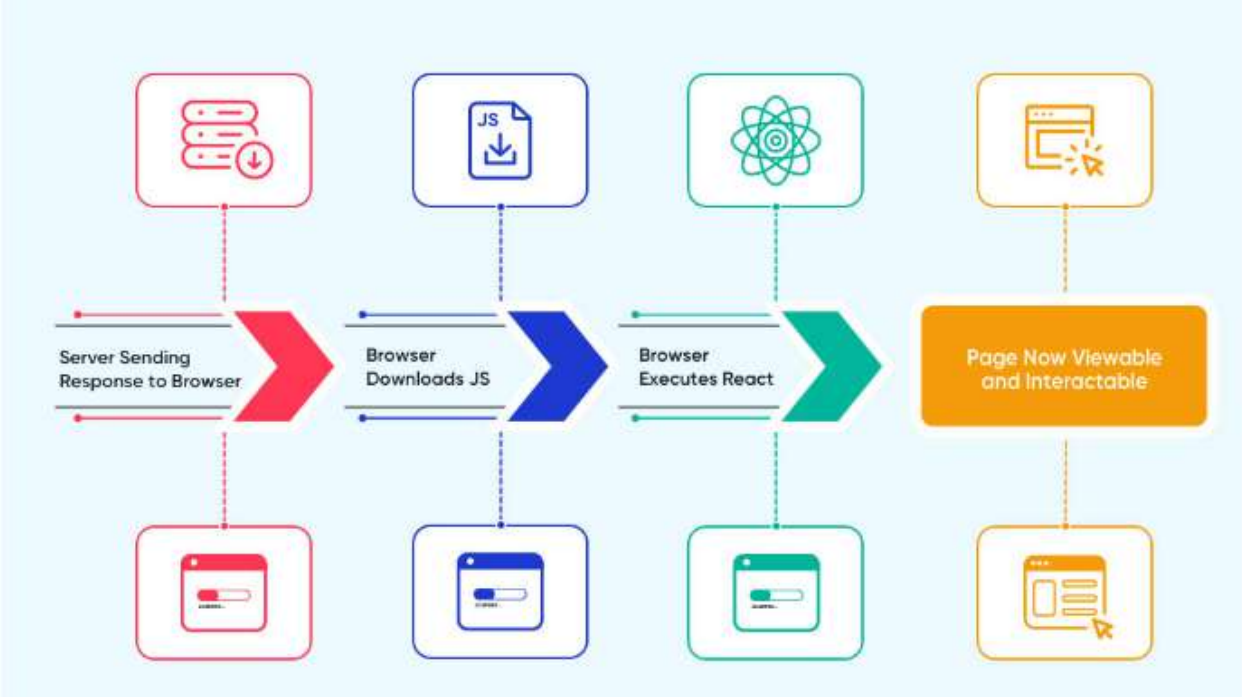
There are different approaches to web application architecture based on where the logic (operation) occurs or is shared between the client (front-end) and the server (back-end). Common examples include:



- **A Single Page Application (SPA) is a single page that dynamically updates as users interact with it, with famous examples being Facebook or Gmail. An SPA is in contrast to a typical website that loads each new page separately as it is requested.**
- **Server Side Rendering (SSR)**

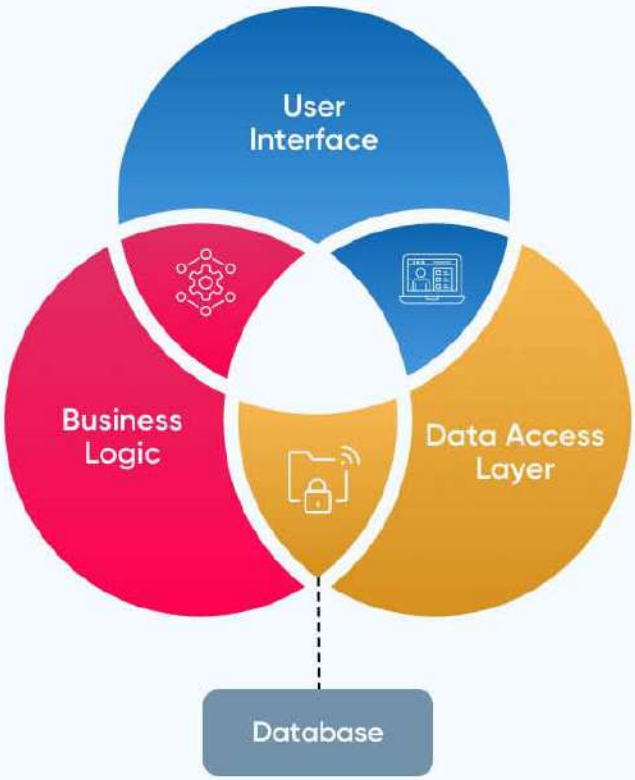


- **With SSR, the server compiles the data and delivers a full-populated HTML page to the client. While this was abandoned due to load delays, SSR is making a comeback by leveraging a framework for dynamic routing to deliver only the necessary data for subsequent pages – an approach popularized with React.**
- **Client Side Rendering (CSR)**

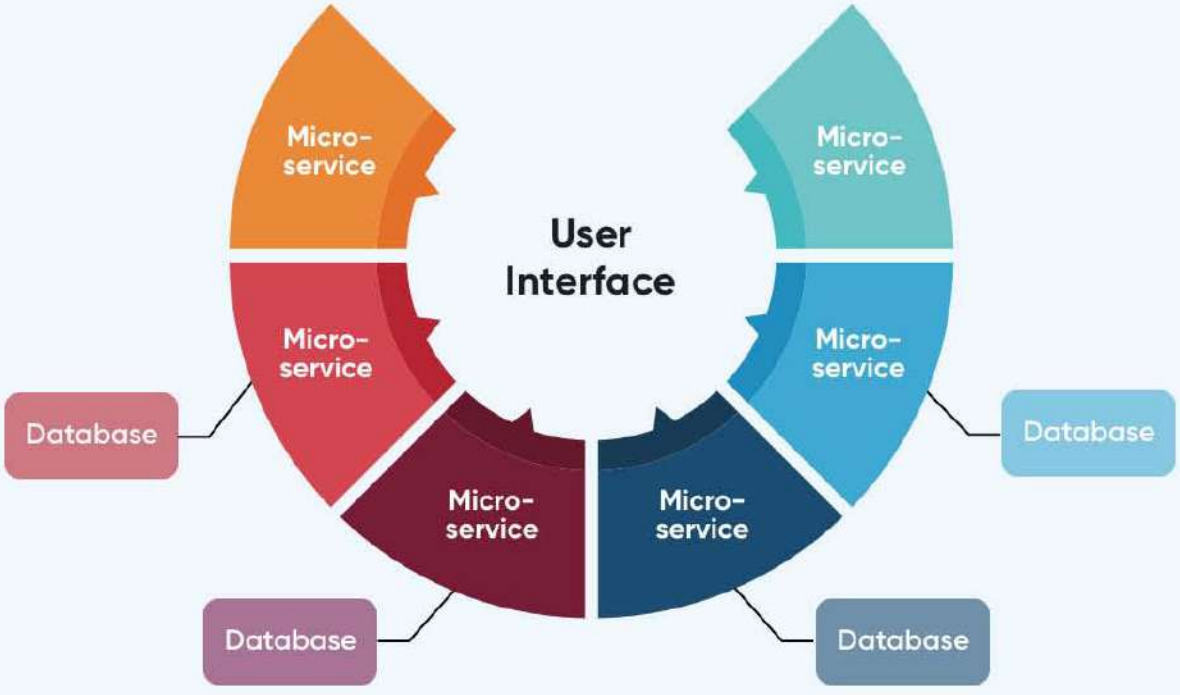


- **With CSR, the browser compiles and renders the web app directly in the browser with JavaScript. CSR can be difficult to keep fast for mobile delivery of web apps.**
- **Monolith Architecture**

Monolithic Architecture

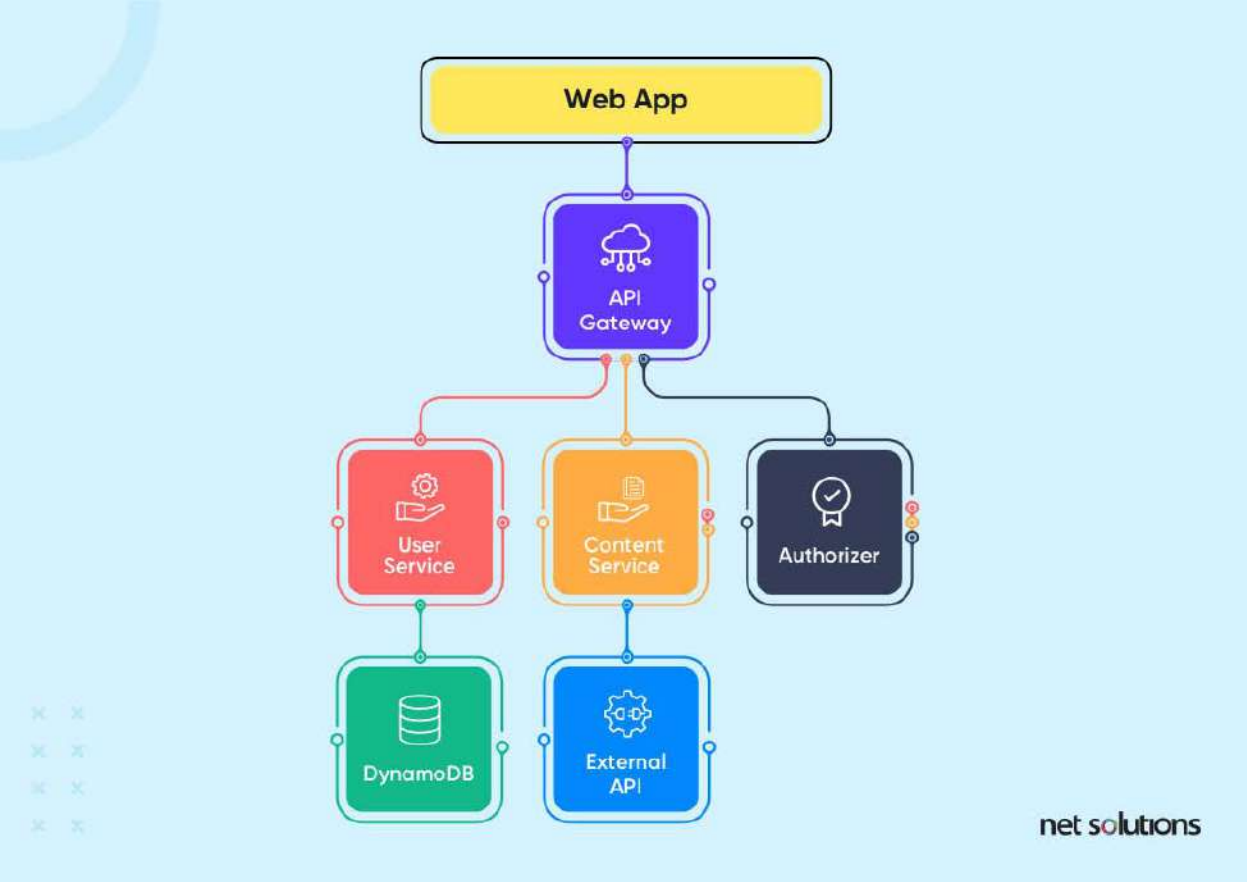


Microservice Architecture

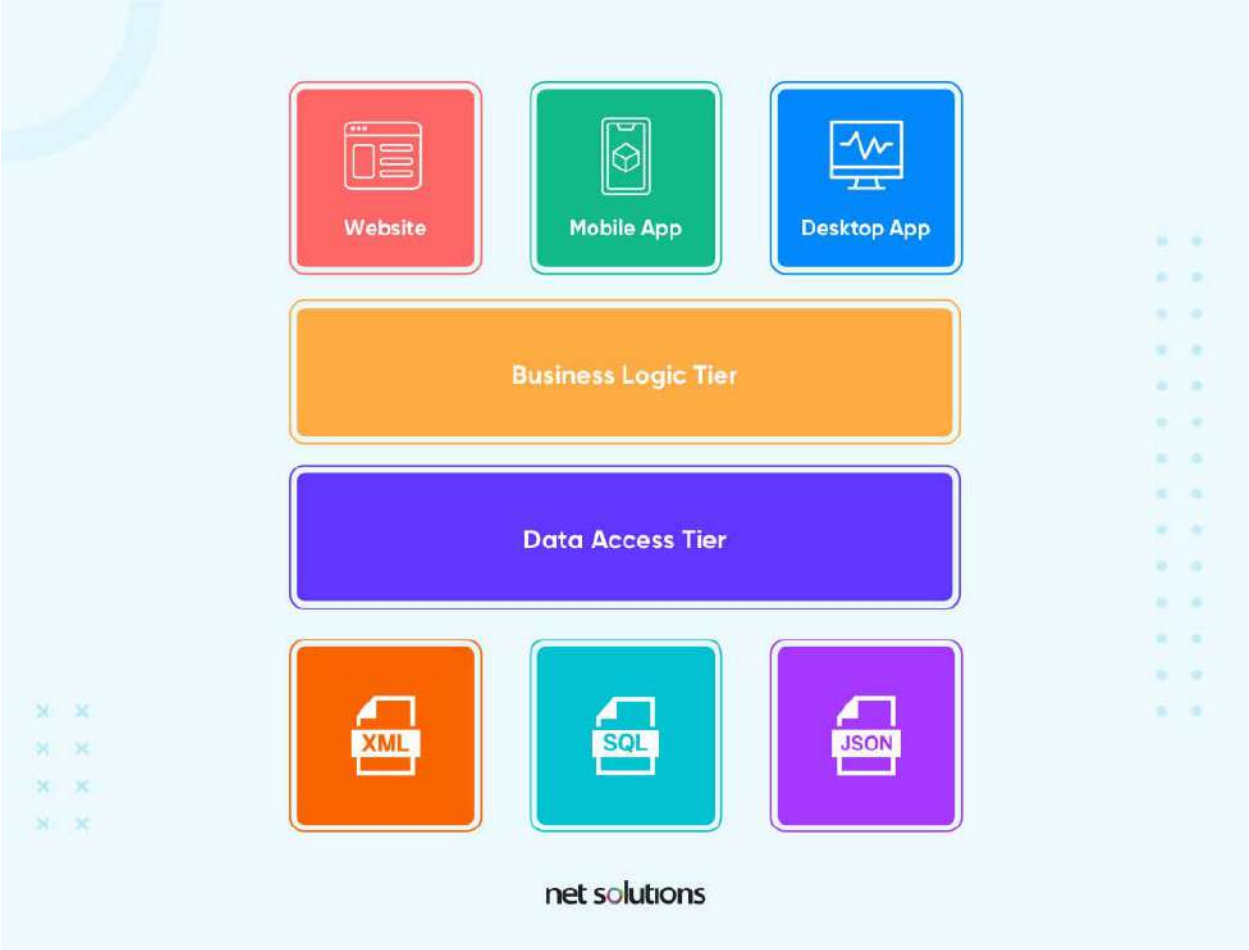


- **Monolith architecture (monolithic or traditional architecture) is a single application built up of three parts: the front-end (UI, presentation layer, client-side), the back-end (server-side) and the database. Monolithic apps are simple to develop, but over time tend to become large, difficult to manage, and complex to update.**
- **Microservices Architecture**

  Microservices architecture decouples the front and back-ends of architecture, linking various independent services in the back-end (microservices) to the front-end via API. The microservices approach supports flexibility to choose and scale best of breed services as needed.
- **Serverless Architecture**



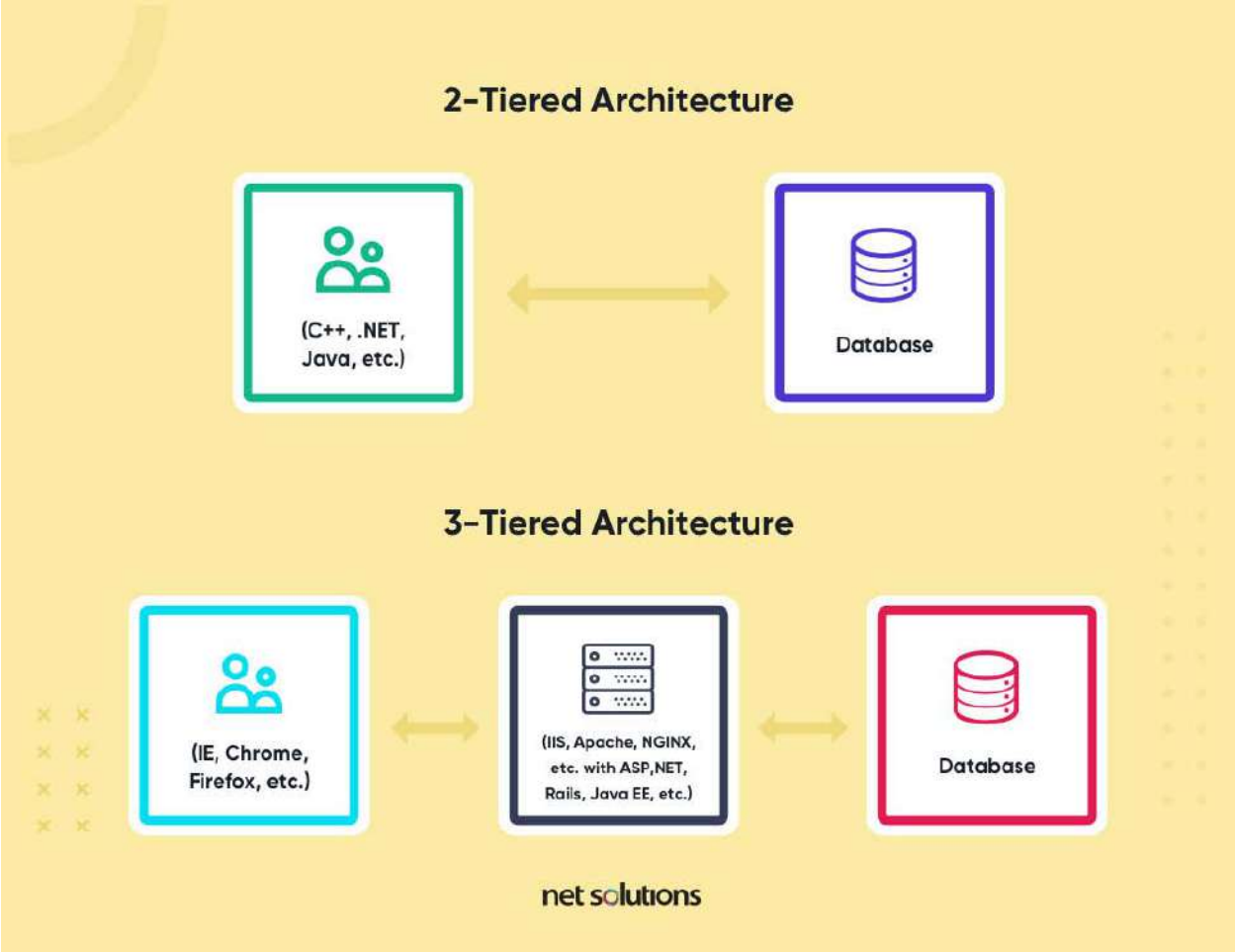
- **Serverless architecture leverages managed infrastructure such as AWS or Google App Engine to support building scalable web apps without having to provision or maintain servers that run the apps or databases.**
- **Multilayer Architecture (n-tier Architecture)**



- **Multilayer or n-tier architecture uses layers as well as tiers (where "n" is any greater than 1" in the application to physically separate (on-premise or cloud) and manage each tier independently. Layers separate responsibilities, with higher layers accessing services in lower layers (but not vice versa). This architecture introduces resiliency and allows each service to scale and perform optimally. In an n-tier architecture, there may be multiple tiers for user interfaces for desktop, mobile, etc.**
- **Three-Tier Architecture**
  A multilayer architecture that has three formalized layers: presentation /

UI, application logic / functionality, and a database tier. Each tier can be independently deployed and managed.

- **Two-Tier Architecture**



A multilayer architecture with only two layers: application (which handles UI and application logic) and database, with no logic layer acting as intermediary between the two.

Web search engines Web Interaction Types

How do search engines interact with websites?

Search engines work by crawling billions of pages using web crawlers. Also known as spiders or bots, crawlers navigate the web and follow links to find new pages. These pages are then added to an index that search engines pull results from.

How do people interact with search engines?

They create an inquiry on a search engine. They click on a result that seems to best match their inquiry. If the page doesn't do what they wanted, they return to the search results and try again. If that doesn't work, they change the wording of their inquiry, hoping for a better result.

8 Different Types of Web Browsers (And Their Purposes)

Web browsers allow you to open websites to get information, order food and check your messages. In a nutshell, web browsing has become an integral part of our daily life, and web browsers make it possible.

In this article, we are going to discuss the most popular web browsers and the purpose they serve, so you can choose one that suits you best.
But before we dive into the features and functions of web browsers, let's understand what web browsers are, and how they work.

What is a Web Browser?

Back in History: The first web browser, developed in 1990, was named WorldWideWeb, and it had very basic features for web surfing compared to modern browsers.
A web browser is a software application that fetches and interprets the code written in HTML (Hypertext Markup Language), CSS, and Javascript to display a webpage that you request. Some web browser examples are Google Chrome, Mozilla Firefox, and Safari.
Web browsers let users locate, view, and use websites. Not to be confused with search engines that rank websites and are accessible via web browsers.
For example, Google is a search engine that has a database of websites that it maintains and ranks. In contrast, Google Chrome is a web browser that uses search engines such as Google to find information, which it displays to the user after interpreting the code.

How Do Web Browsers Work?

When a user types a URL into the address bar and clicks on the link to access a particular website, the browser sends a request to the web servers of the website web host.
The web server responds to the request by sending the requested data back to the web browser. The requested data is sent in computer code. The code contains HTML, CSS, JavaScript, and any other files required to display the requested web page.
The web browser renders the page by applying CSS and Javascript rules to display the web page to the user. It renders the entire web page, including text, images, animations, and other media files embedded in HTML.
A web browser is one of the most important applications on your computer, tablet, or phone. It displays websites and web pages on your device. But displaying a website is not the only function of a web browser. Other than displaying web pages, web browsers can also execute scripts, run plug-ins, and interact with other web technologies, such as cookies

and cache for web storage. Most web browsers offer features like bookmarks, history, and tabbed browsing. Different types of web browsers allow users to manage their browsing experience and access website pages later quickly (pocket).

Different Types of Web Browsers

Here is a breakdown of the different types of web browsers that you can choose from:

1. Google Chrome

Google Chrome, launched in 2008, has become the most popular browser mainly because of its fast speed. It can load several pages instantaneously with just a single click. The web browser is also constantly updated with the latest security features to ensure safe web browsing. It is the default web browser for all google devices.

It is also compatible with other devices, operating systems, and web technologies. Google's streamlined design focuses on a minimalistic user interface that is clutter-free. You can also customize your browsing experience with extensions. Its incognito mode will not save the browsing history and site cookies.

2. Safari

Apple Safari, or simply Safari, is a popular browser invented by Apple and is therefore preinstalled and set as the default web browser on all Apple iOS devices.

Safari has cross-site tracking prevention, which protects you from third-party web trackers when you surf the web. It is a free browser for all iOS devices, and you can sync and access your browsing history on any of your Apple devices.

3. Microsoft Edge

Microsoft Edge is an open-source web browser built by software giant Microsoft in 2015. It comes as a default browser on Windows 10. It is also available on macOS, iOS, and Android. Microsoft Edge has replaced Internet Explorer, its older and slower ancestor.

This web browser is designed to be compatible with all web standards and is capable of rendering modern websites and applications. Unlike its predecessor, Internet Explorer, it is faster and more secure.

4. Internet Explorer

Internet Explorer, also known as Microsoft Internet Explorer, was a web browser developed by Microsoft and launched on the World Wide Web in 1995. Since then, Microsoft has rolled out 11 versions of Internet Explorer.

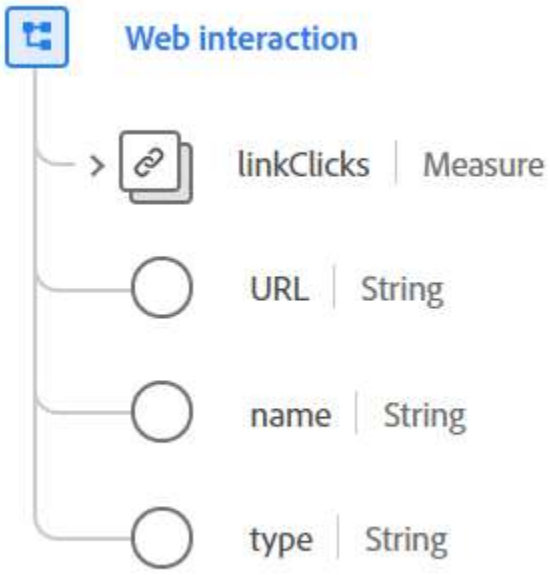
Support for Internet Explorer 11 ended on June 15, 2022. It has now been replaced with Microsoft Edge.

Some problems that led to the demise of Internet Explorer include:

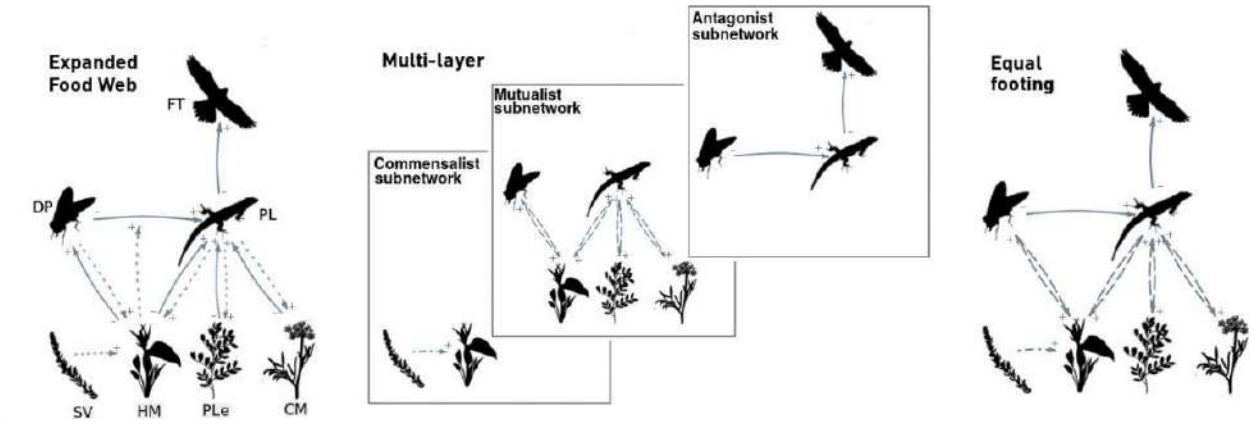
- **Slow browsing**
- **Lack of active updates**
- **Limited Customization**
- **Lack of essential browser features**

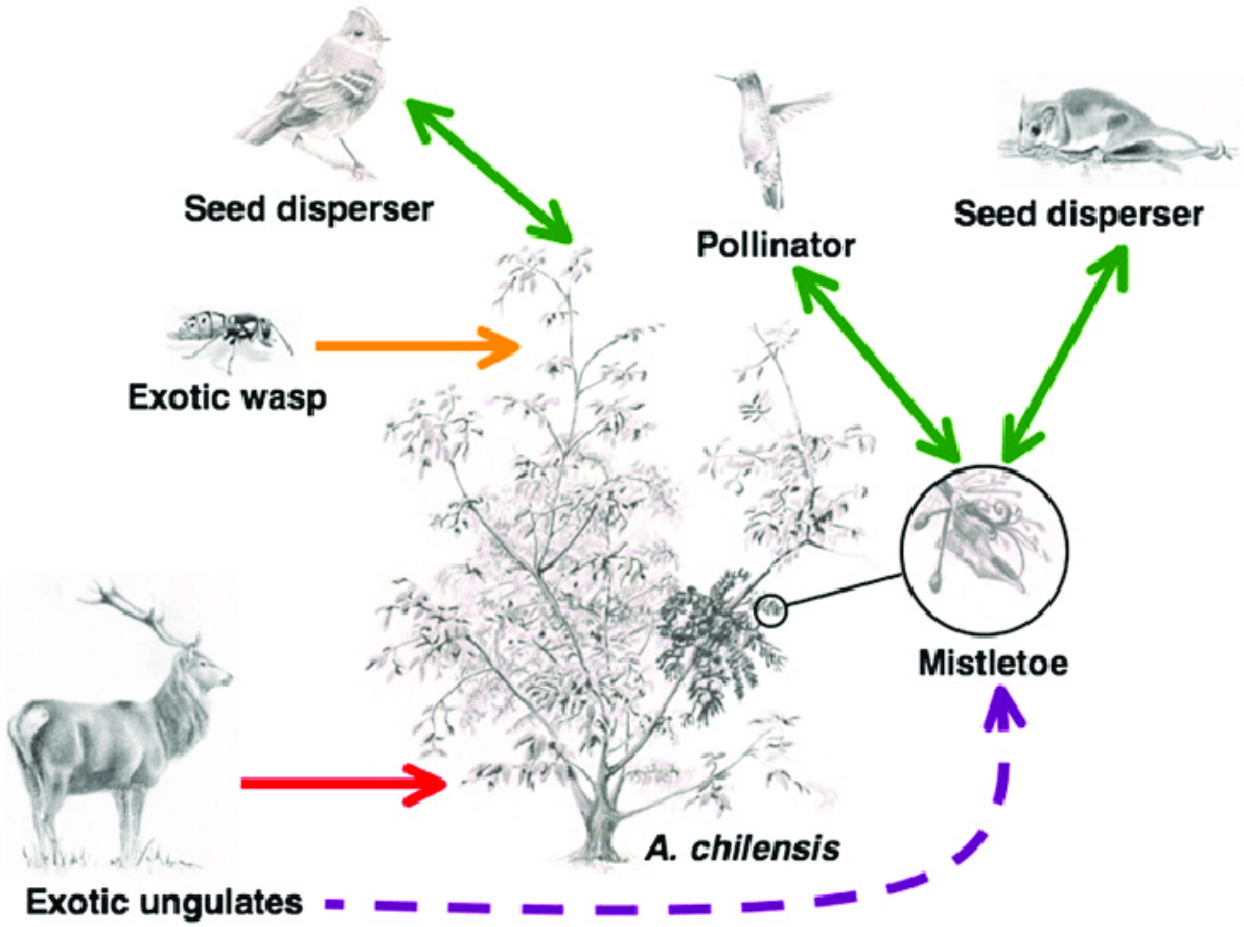
Web interaction

Web interaction is a standard Experience Data Model (XDM) data type that describes information about interactions that happened on a web page after the initial page load was completed. It is intended for recording interactions in rich web applications that do not trigger a new page load such as single-page web apps (SPAs).



Property	Data type	Description
`linkClicks`	Measure	A measurement tracking the click of a web link.
`URL`	String	The actual link or URL used for this web interaction.
`name`	String	The normative name used for this web link. This is used for classification purposes.
`type`	String	The link type. This property must be equal to one of the following enum values: • `download` • `exit` • `other`





Significance Examining the effects of species gains and losses is fundamental to understanding the assembly and disassembly of ecological communities in a changing world. However, field-based empirical studies that demonstrate the disassembly of mutuality webs are exceedingly rare. In this study, we take advantage of an ongoing natural experiment that links the gain of invasive species (introduced large mammals and wasps), the loss of a native keystone species (a mistletoe), and the ensuing node-by-node disassembly of an interaction web in Patagonia, Argentina. Our results demonstrate that the gains and losses of species are both consequences and drivers of global change that can lead to underappreciated cascading coextinctions through the disruption of mutualisms.

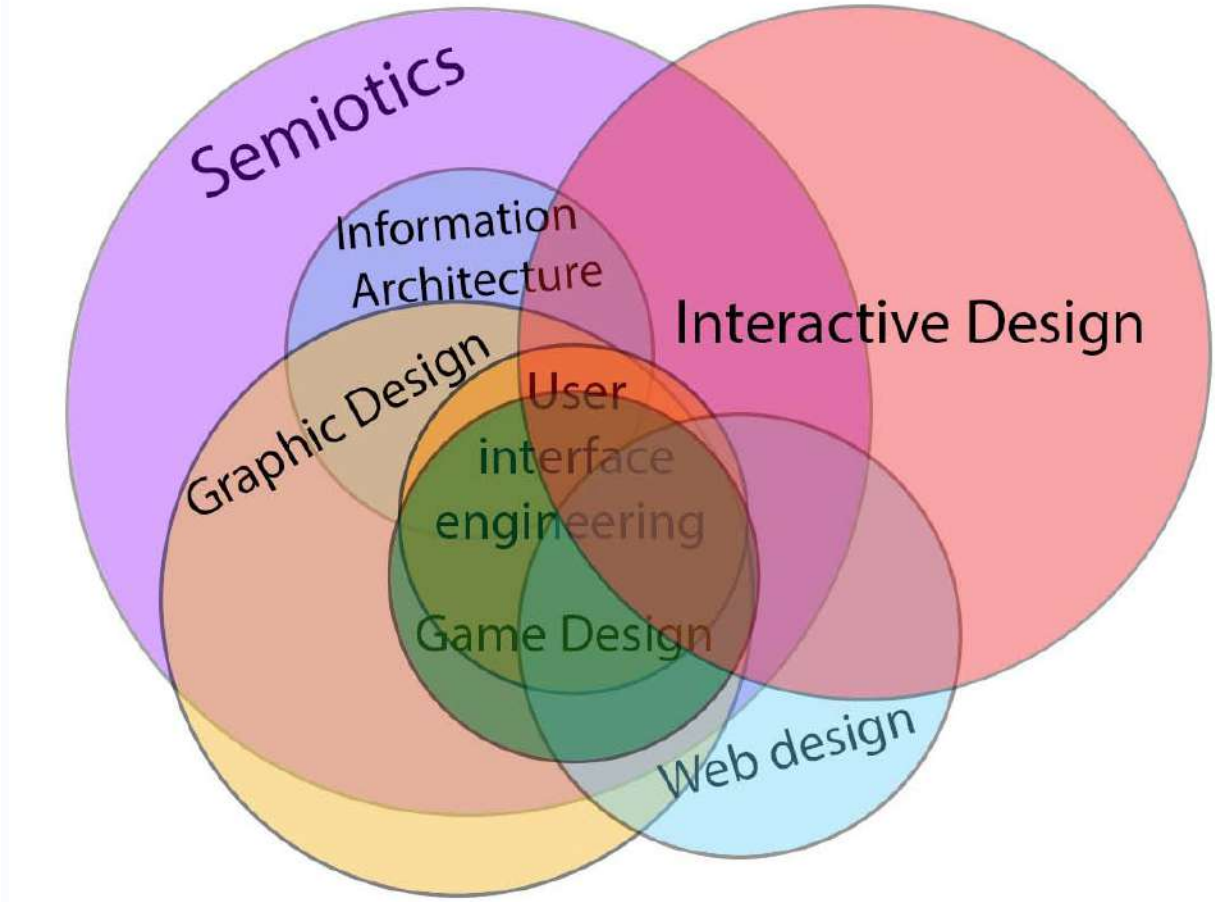
What is an Interactive Website?

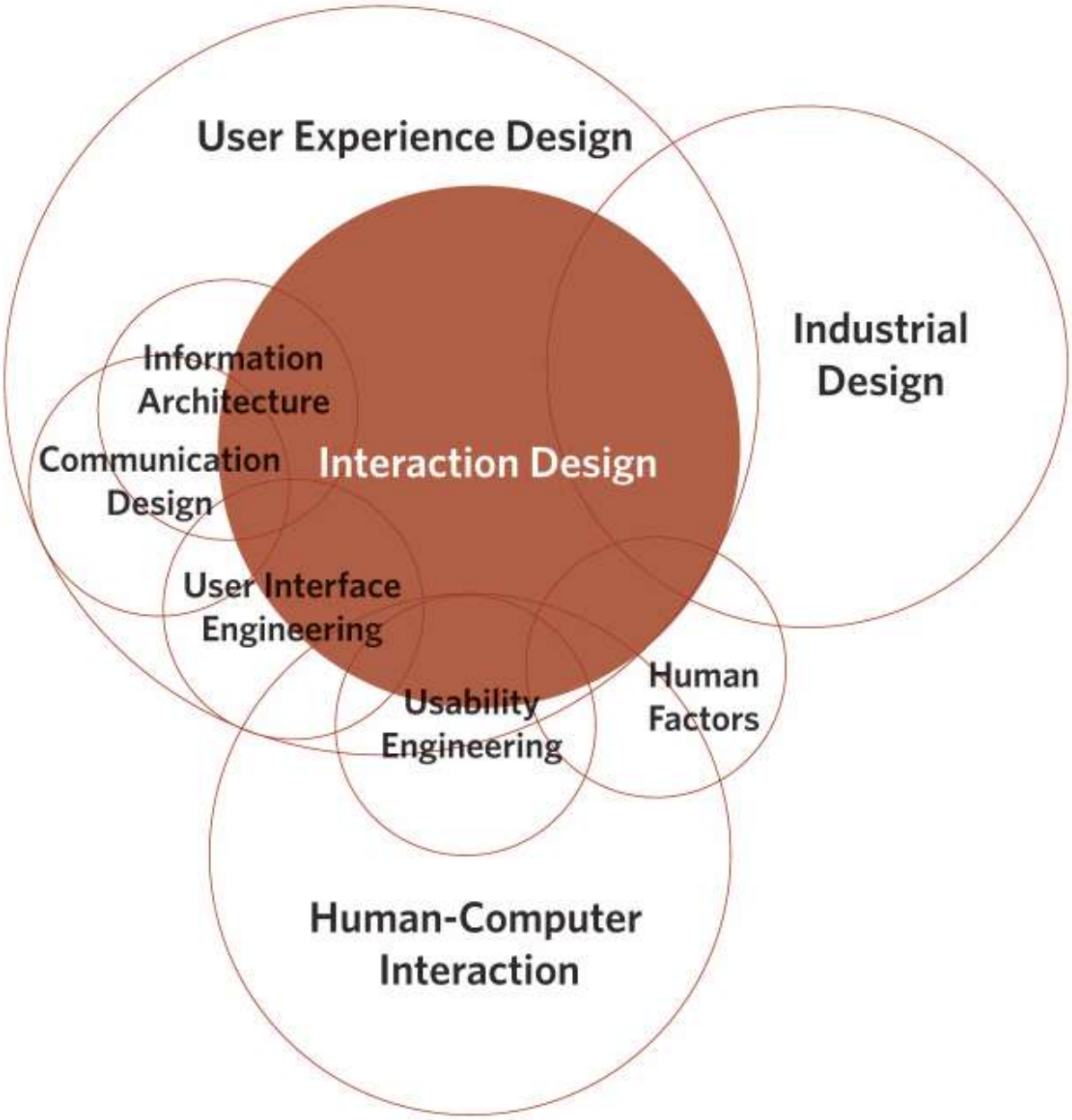
Put simply, an interactive website is a website that's primarily created to provide enough space for communication and interaction with the users. It's a different kind of internet page in that it uses numerous software to create an

interactive experience and does a great job allowing the user to be actively participating and engaged with the website.
An interactive website helps you increase many metrics for your business such as engagement, retention, and activation.

Why is this necessary?

In this constantly developing digital age, you can be sure that putting some random content on your website and simply sitting and waiting for hours for it to work won't really take you anywhere.
With an interactive design for your website, you will show off attractive content and encourage users to communicate and engage greatly with your service.
Before going any further, let's get some essentials out of the way
An interactive website can indeed be just about any website that enables the users to do more than just reading the given text or scrolling through images. But there's always being something and being really great at something. So, what makes a more interactive website?
There are certain features that create room for a better opportunity to build an interactive website that really pushes the limits.
Features like advanced navigation menus, internal links, search bars, forms and buttons, videos, animated elements -3D hover or scroll effects- interactive feedback tools, etc.
As long as your website contains these features within the user experience and manages to make alterations according to the fresh needs and wants of the audience, congrats! You can consider your work as a perfectly interactive website that will do a great job attracting the viewers and keeping the user engagement at its finest.
Now that you're through with the detailed characteristics let's take a closer look at some of the best interactive websites that will inspire you to get started – or get going!


Semiotics
Information Architecture
Interactive Design
Graphic Design
User interface engineering
Game Design
Web design

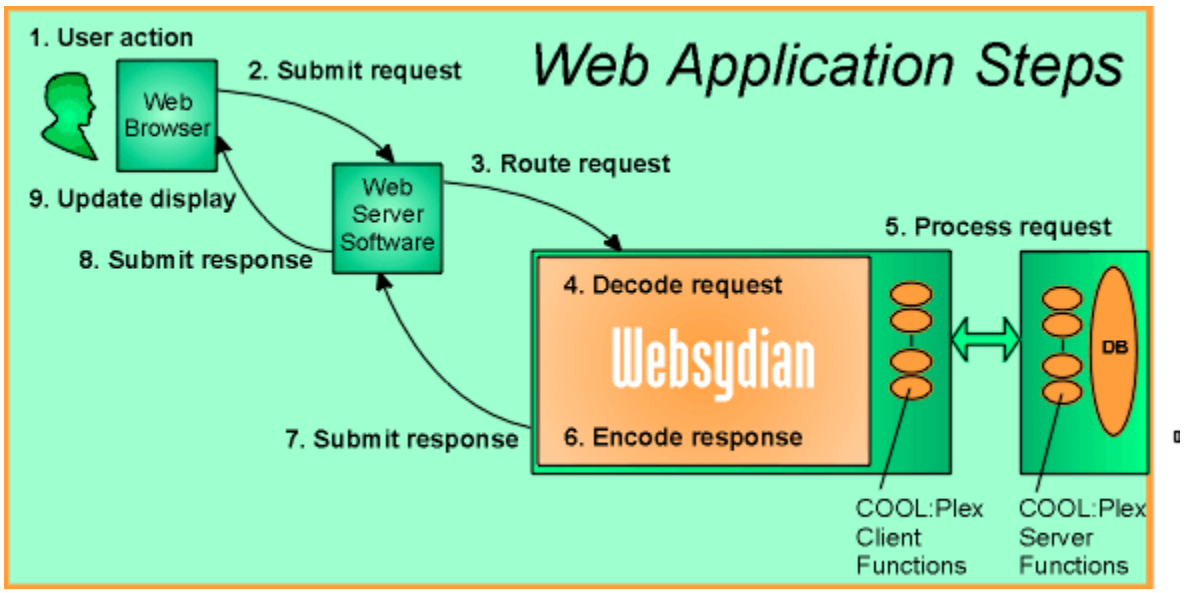


The Steps in an Interaction between Web Browser and Server

It is - especially when debugging web applications - very useful to have a reasonable good understanding of the sequence of events that takes place from the time that a web browser submits a request to get a new web page, to the time it receives it. We can think of this as a *Web Interaction*.
In this section, we go through the steps that occur, and we look at the subset of these steps you have to develop to get a working Websydian application.

Web Application Steps

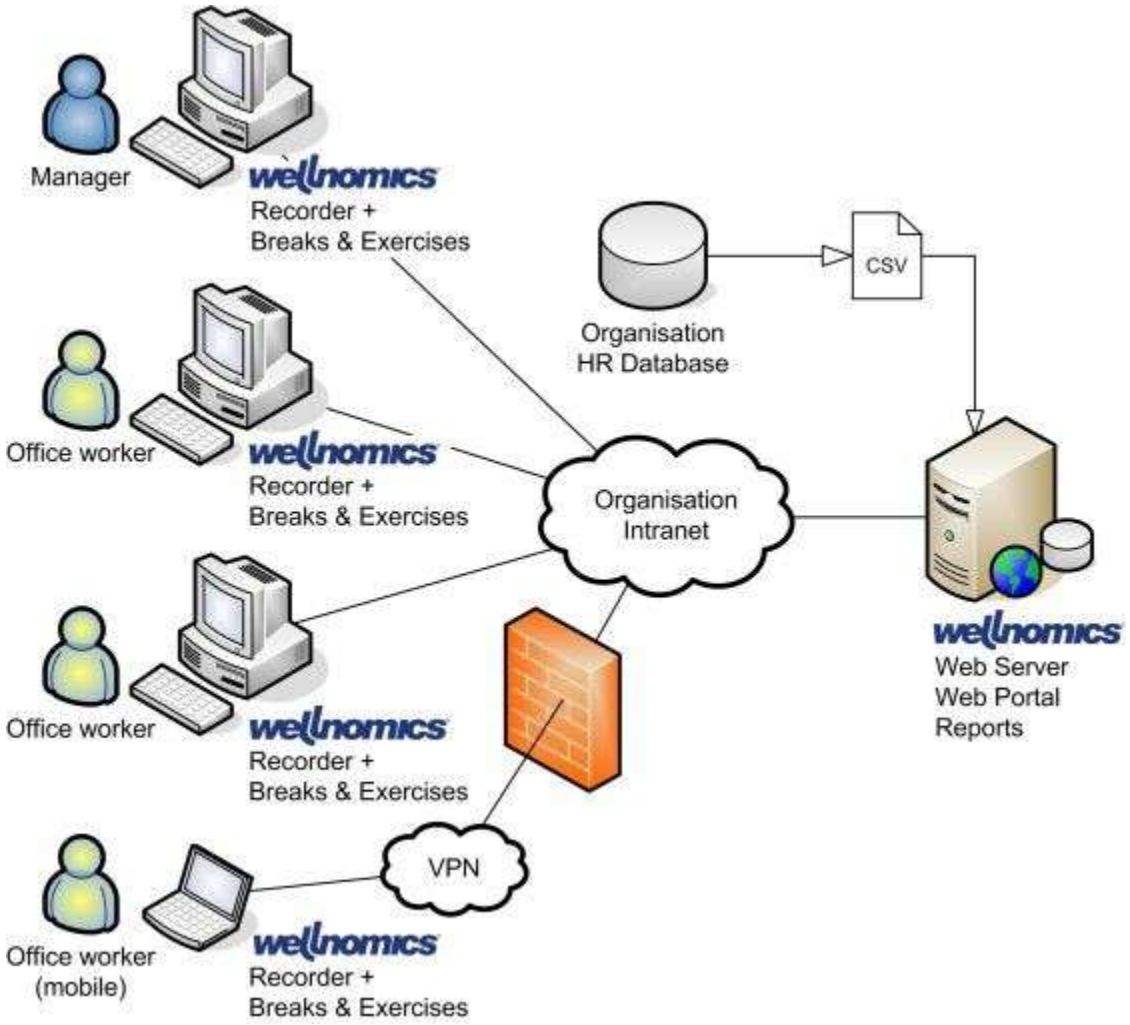
The steps from a user request to a Websydian application to the display of the response in a web browser is depicted in the figure below.



The steps are the following:

1. **The web browser user takes some action.**
2. **The Web Browser submits the corresponding HTTP request.**
3. **The Web Server Software routes the request to the server of the Web application.**
4. **The Web Application Server decodes the request.**
5. **The Web Application Server processes the request.**
6. **The Web Application Server encodes the response as an HTML Web Page.**
7. **The Web Application Server submits the response to the Web Server Software.**
8. **The Web Server Software submits the HTTP response to the Web Browser.**
9. **The Web Browser updates the display.**

An application server typically can deliver web content too but its primary job is to enable interaction between end-user clients and server-side application code the code representing what is often called business logicto generate and deliver dynamic content such as transaction results decision support or real.



Systematic Development of WISs

Web Application Development

Many of the most valuable websites on the internet today are actually web applications. You may wonder: What's the difference? There are quite a few, but the major distinction is that web applications are interactive with the user. Many businesses choose web applications over websites to pick up information they can use in their marketing strategies and create products or services that generate revenue.
Here's what a web application is and how you can get started building one.

What is a web application?

A web application is an interactive program that runs on a web server and is accessed through a web browser. A web app is built so that the user interface provides data back to the development team that designed it. This data offers insights into customer interests, usage, and preferences that can prove invaluable to product and marketing strategies. The data can also inform optimization and other client-centered aspects of the mobile app or desktop applications.
A big difference between web applications and webpages is that user experience dictates web application design. Traditional web design is based on server-side programmers making decisions about what might enhance usability. In contrast, a web application has an application program interface (API) that takes in large-scale data from the user side and then funnels that information into automation.

For example, if a company has implemented an e-commerce web application that can track and report on which products a potential customer lingered over the longest, similar items can be suggested by the web application the next time the customer logs in. If that same company only had a website, the user's experience would be based on what was available to every customer: a static and standardized set of options.

Most mobile applications that appear on smartphones are web apps. Here's a brief list of common web applications:

- **Facebook, Instagram, Twitter, and other social media platforms**
- **Gmail, Yahoo, AOL, and any web-based email programs**
- **Any self-service customer portal**
- **Query sites, such as Quora and Google**

It's important to note the distinctions that exist between native web apps and mobile apps. Mobile apps live on the device and are designed to run on a specified platform (such as iOS or Android). Examples would include Facebook Messenger and Google Maps. A Google search that takes you directly to the internet browser would be an example of a web app, while Google Maps is a mobile app.

Remember that web applications are all about input, and traditional sites are all about output. If you want customers to enter their information, you're likely looking at a web application. If you just want customers to read static information that they don't contribute to as a user, a webpage might better suit your business needs.

What is the difference between web applications and websites?

A website is built on a platform that only the creator or developer can change. On the other hand, a web application is interactive and built on a platform that allows user data to inform iterations of the application. As they are a collection of HTML documents, web applications might be part of an established website or built as freestanding applications.

The application must be delivered over a network and connected to a database. Where traditional websites serve the major purpose of delivering information (such as text or video) to the user, a web application allows its interactions with the user's request to produce a variety of possible outcomes.

What is web application development?

Developing a web app is all about setting goals for the app's purpose. What need does the app fulfill? The user interface should be designed with that answer in mind. Information about the consumer will come from the user interface, so developers should design the app to receive and respond to that information. Web development entails tasks like:

- **Making sure the web app offers compatibility with both Android and iOS**
- **Identifying life cycle and optimization metrics**
- **Building an intelligent, iterative user interface**

To ensure these web development tasks are effectively handled, it might be beneficial to leverage reliable independent talent—such as front-end and back-end developers available on Upwork. This gives you the peace of mind that comes from knowing your web application is being built by knowledgeable professionals. Upwork's Project Catalog™ also provides the option to buy a specific project rather than having to find an independent professional on your own.

Skills needed for web app development

While a background in coding is not necessary for the creation of a web app, knowledge of programming language is extremely helpful. This allows you to design the web app yourself or express your requirements to any development team that you may hire.

Strong web app development relies on the deployment of both front-end and back-end skills. Examples of front-end skills include knowledge of HTML, JavaScript, and CSS. Back-end languages include Python and PHP. Having a grasp of DevOps programs, such as Jenkins and GitHub, is also useful when building your web application.

One final web development skill helpful in building a web app is knowledge of Ajax. Short for "asynchronous JavaScript and XML," Ajax is not a program language but rather a collection of techniques that can be employed to develop a web application. Ajax enjoys popularity in client-side programming for its ability to exchange server-side data without interfering with user experience.

Web application development frameworks

Most web applications are built using a web application framework that simplifies code and helps reduce errors. Popular web application frameworks include:

Front-end:

- **Svelte**
- **Vue.js**
- **React**

Back-end:

- **Laravel**
- **Ruby on Rails**
- **Django**

Some web application frameworks, such as ASP.net, handle both front- and back-end duties. Model-view-controller (MVC) is one such design model that breaks the data within the app into three connected sections: the model, the view, and the controller.

6 steps for getting started with web application development

Web application development is in high demand. Companies and individuals looking to build out web apps have more project options than ever before and access to experienced talent, like the independent professionals found on Upwork, to get the job done. The next segments will offer a step-by-step guide to the web application development process.

1. Isolate a strong app idea

Good apps are born out of need and innovation. Think of your app as a solution to a problem almost everyone has. Don't think of your web app development as having to rely on an idea so new and brilliant it has never been heard of before. Rather, approach the design of your app with a solutions-based attitude and think about how the app can improve on the user's day-to-day life.

2. Complete focused market research

Who will be served by your new web application? Does this app have a highly specified target market (e.g., working parents), or is it something that you believe has a more generalized appeal (e.g., any person aged 15 to 55)?
Finding out whether and where there is a demand for your web app comes down to conducting intense market research.

3. Define your app's functionality

Reining in the expectations for your app's functionality allows you to focus on key elements that will make users prefer it.
Having a clear idea of what constitutes successful functionality is crucial to how you will carry out your web app development. If your app will manage items like payments, passwords, or contact lists, pay special attention to creating a unique user experience with regard to those items. For example, if you're designing a web app that will primarily streamline the buying experience for a retail home decor company and house past customer purchase data, it could be helpful to dedicate time to making sure the checkout experience is positive. Whether this means the shopping cart offers suggestions that may match carted items or allows for one-click buying, designing your app to leave a positive impression in customers' minds is the key to success.

4. Sketch out your app design

Mapping out the actual layout of your app comes next. Using a pencil and paper or another web application like Google Docs, write out the general outline for your app. Include the placement of text, images, buttons, and other interactive or animated components.
Create a workflow that mimics how you imagine the app will flow. For example, from the login screen, what happens next? And after that? Keep designs orderly but detailed so that the prototyping phase can run as smoothly as possible.

5. Work on prototypes

Creating wireframes and prototypes for your web application represents the next step in development. Wireframes are structured, computer-

generated sketches of your app. Prototypes are wireframes that have functionality and at least some small measure of interactivity.
The wireframe and prototype steps are where you will troubleshoot for your web app and come up with ways to present options for the user interface. Let's say you run a small flower shop. An example of a dynamic user interface might include having a little flower bud appear as the cursor or navigational tool within the app and then programming it to bloom when items are selected or placed in the cart by the customer.

6. Web app validation

Now for the fun step of having people try out your potential app. Alpha testing is done by designers, while beta testing is completed by users of the web app. Alpha testing entails addressing major problems or issues within the app by experts. Beta testing is more like having the app polished by public users. In alpha testing, software professionals do everything within their power to trick the app into an error. In beta testing, you want the largest possible number of public users to utilize the app, tell you what they're doing or attempting to do, and report any malfunctions encountered in the process.
Gather friends, colleagues, neighbors, and anyone willing to give you feedback. Let them play around with the app and see where strengths and weaknesses lie. This is where you get to learn information about user experience firsthand. Taking the time to validate your web app saves you immeasurable time and money by locating problems or areas of improvement before the app's formal launch.

Application domain description

The name "app" is a short form of the word application often used in the IT sector. This domain name is to be used by developer companies, professionals and enthusiast developers and entrepreneur's applications, app-support services or other useful related products and tools.

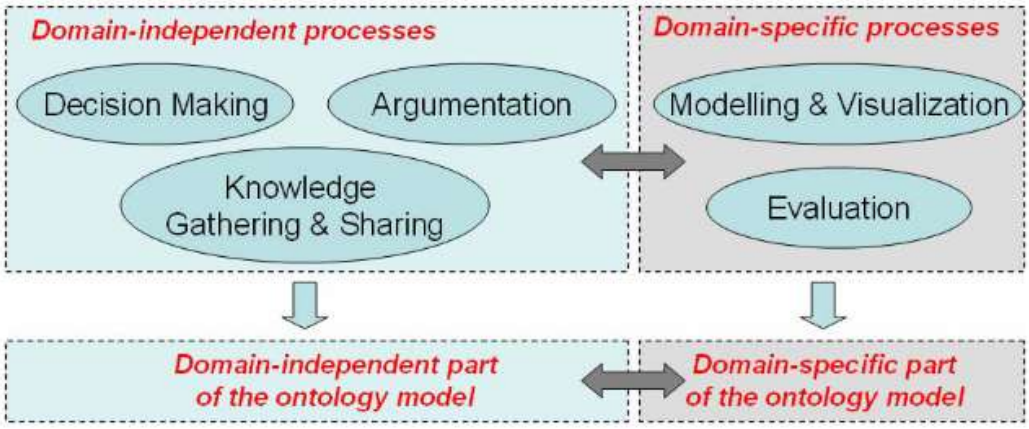
Problem/ Application Domain: The domain in which the software system will be expected to run and survive. As in this is the domain in which the software system will be applied to. Application/Solution Domain: The software solution/application applied to the problem domain to meet a client's requirements.
One of the main advantages of using a .
domain is its added security features. All . app websites must use HTTPS encryption, which helps protect users' data and ensures that connections between websites and users are secure.

What is the difference between application and domain?

The main difference between them is that domain services hold domain logic whereas application services don't. As we discussed in a previous post, domain logic is everything that is related to business decisions.

There is typically a problem domain and a solution domain. The problem domain describes a situation that needs improvement. For example, it can describe the concepts and processes in a human resources department. The solution domain describes one of possibly many solutions to a problem. One solution might be a streamlined process, another might be an application that takes over parts of a process. An application domain would fall under the solution domain. It would be a description of an application that improves the state of the problem domain.

Some may nit pick over a perception of subtle (or major) differences between the application and solution domain. For the most part and at least in all the software engineering text books I've read the terms are often used interchangeably and will differ in their precise meaning from author to author. This can cause confusion as I've seen application used in two conflicting contexts.
Problem/ Application Domain: The domain in which the software system will be expected to run and survive. As in this is the domain in which the software system will be applied to.
Application/Solution Domain: The software solution/application applied to the problem domain to meet a client's requirements
For these reasons, I dislike the phrase Application Domain because of its potential for ambiguity in how it may be interpreted. I prefer to say 'Problem Domain' (situation in which the software will be used) and 'Solution Domain' (the software and systems built to solve issues in the problem domain).



Web design architecture

Web application architecture is a blueprint of simultaneous interactions between components, database instances, middleware systems, user interfaces, and servers in an application.

What is web architecture?

Web Architecture can be defined as the conceptual structure of the internet. Types of web architecture include the client-server model and three-tier model.

What does a web architect do?

The Web Architect is responsible for planning, designing, testing, implementing, and administering interactive technologies and applications. He/She creates, maintains, and updates websites, mobile, and social media apps

How do I create an architecture for my website?

Website Architecture Best Practices

1. **Create a simple top-level navigation menu.**
2. **Keep your URLs simple and user-friendly.**
3. **Model your website architecture after the top players in your industry.**
4. **Keep your website consistent.**
5. **Implement the pillar-cluster internal linking model.**

What is system architecture in web design?

Web architecture refers to the overall structure of a website or web application, including the way it is designed, implemented, and deployed. It involves the use of technologies and protocols such as HTML, CSS, JavaScript, and HTTP to build and deliver web pages and applications to users

What are the 3 types of web?

Websites can be broadly classified into three main types: static websites, dynamic websites, and e-commerce websites. Each type has its own characteristics, advantages, and use cases. Let's take a closer look at each of them.

How an architect works?

Architects are professionals trained in the art and science of building design. They develop the concepts for structures and turn those concepts into images and plans, which eventually may become homes, office buildings and other facilities. Their work involves more than just the appearance of a structure

Do architects need a website?

All architects will want to make sure their brand gains as much credibility as possible. While you do that by completing work on-time and ensuring you always go above and beyond the call of duty for clients, excellent website design can also produce excellent results

How to design system architecture?

In this article, you will learn how to design a system architecture that can evolve with user needs and expectations.

1 Understand the problem domain. ...

2 Choose an architectural style. ...

3 Define the components and interfaces. ...

4 Design the interactions and workflows. ...

5 Evaluate and validate the architecture

Web Application Architecture Components

Typically, a web application architecture diagram comprises three core components:

1) Web Browser: The browser, client-side, or front-end component is the key component that interacts with the user, receives the input, and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well if required.

2) Web Server: The web server, also known as the backend or server-side component, handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.

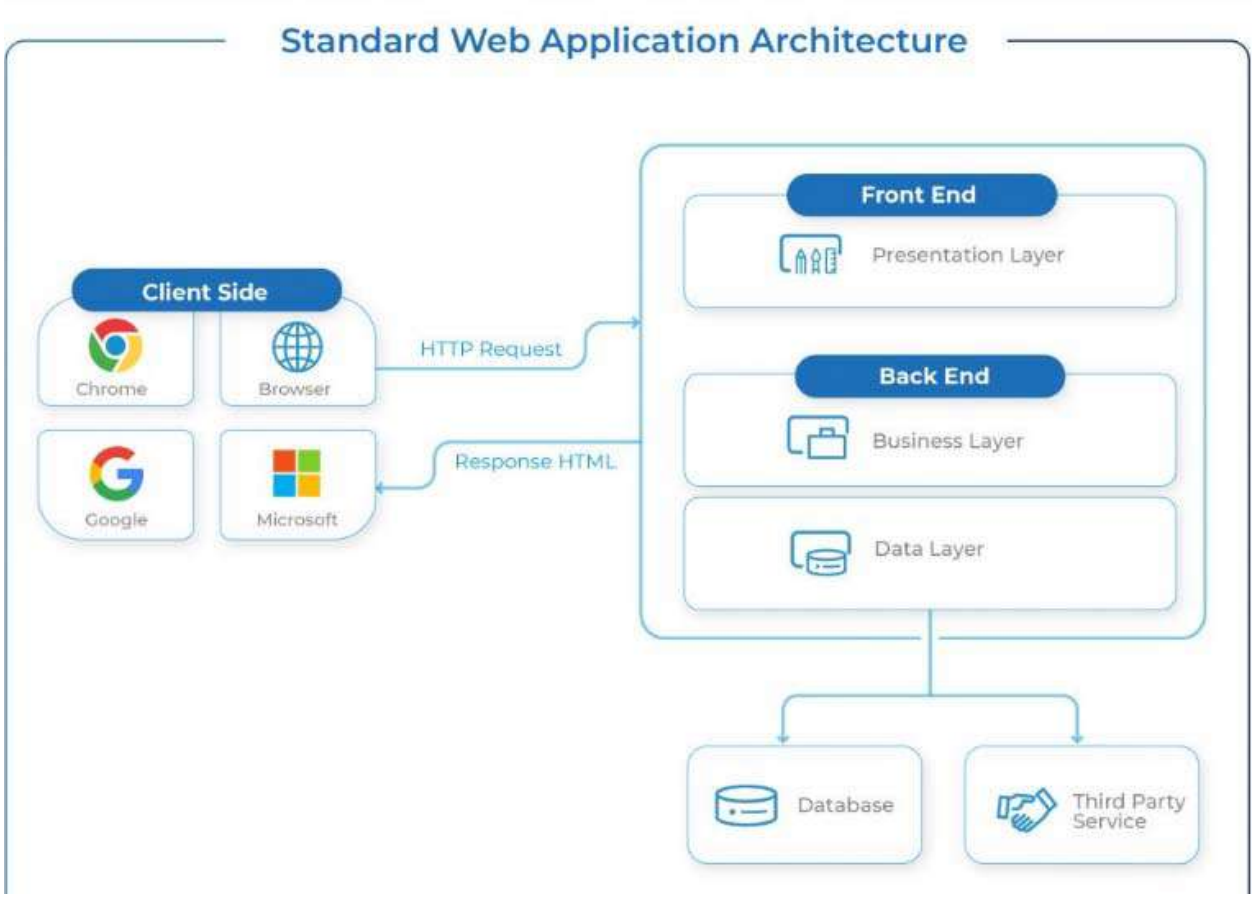
3) Database Server: The database server provides the required data for the application. It handles data-related tasks. In a multi-tiered architecture, database servers can manage business logic with the help of stored procedures.

What is a 3-Tier Architecture?

In a traditional 2-Tier architecture, there are two components: the client side system or the user interface and a backend system, which is usually a database server. The business logic is incorporated into the user interface or the database server. The downside of 2-tier architecture is that the performance decreases with an increased number of users. Moreover, the direct interaction of the database and the user device raises security concerns. Railway reservation systems and content management systems are a couple of applications that are usually built using this architecture.

There are three layers of a 3-Tier architecture:

1. **Presentation layer / Client Layer**
2. **Application Layer / Business Layer**
3. **Data Layer**



In this model, the intermediate servers receive client requests and process them by coordinating with subordinate servers applying the business logic. The communication between the client and the database is managed by the intermediate application layer, enabling clients to access data from different DBMS solutions.

The 3-tier web application architecture diagram is more secure as the client does not directly access the data. Deploying application servers on multiple machines provides higher scalability, better performance, and better re-use. You can scale it horizontally by scaling each item independently. You can abstract the core business from the database server to perform load balancing efficiently. Data integrity is improved as all data goes through the application server, which decides how data should be accessed and by whom. For that reason, a change of management is easy and cost-effective. The client layer can be a thin client, which means hardware costs are reduced. This modular model allows you to modify a single tier without affecting the remaining components.

Application Layer: Web Server

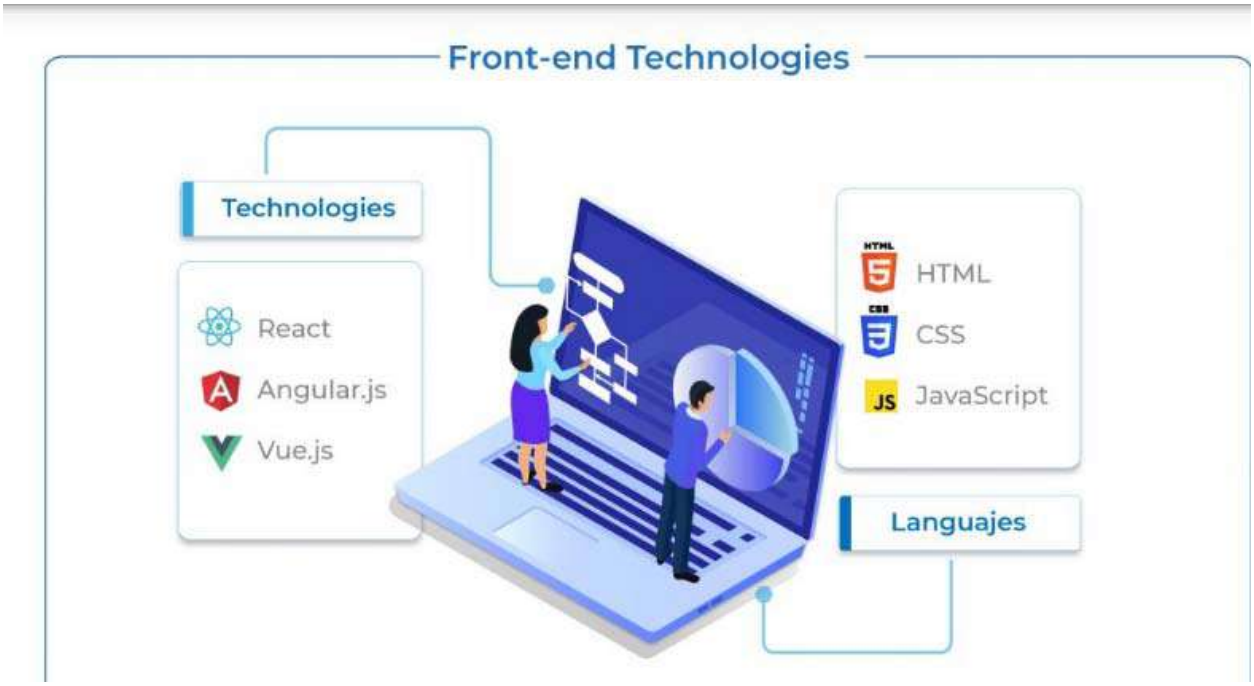
What is a web server? Simply put, a web server runs one or more websites or web apps. The web server uses HyperText Transfer Protocol (HTTP) and other protocols to view user requests via a browser. It processes them by applying business logic and delivering the requested content to the end-user.
A Web Server can be a hardware device or a software program.

- **Hardware Web Server: A computer device connected to the internet and contains web server software and web app components such as images, HTML docs, JS files and CSS style sheets.**
- **Software Web Server: It is software that understands URLs and HTTP protocols. Users can access it via domain names to receive the requested content.**

While a static web server delivers the content to the browser as it is, a dynamic web server updates data before delivering it.
Apache is a popular open-source web server from the Apache Software Foundation. It was developed by Robert McCool in C and XML in 1995. Apache is based on a process-driven model wherein every request results in the creation of a new thread. The modular design of Apache allows you to scale individual resources easily. With minimal configuration, you can manage even large traffic. It works on OS, Windows, and Linux environments. However, Linux is the most preferred environment for Apache. While it uses a file system to process static content, dynamic content is processed within the server.

Presentation Layer: Client-side Component (Front-end)

The client-side component of a web application architecture diagram enables users to interact with the server and the backend service via a browser. The code resides in the browser, receives requests, and presents the user with the required information. This is where UI/UX design, dashboards, notifications, configurationally settings, layout, and interactive elements come into the picture.
Here are some of the commonly used front-end technologies for web application architecture diagram:



CSS

CSS or Cascading Style Sheets is a popular style sheet language that lets developers separate website content and layout for sites developed using markup languages. Using CSS, you can define a style for elements and reuse them multiple times. Similarly, you can apply one style across multiple sites. It is simple and easy to learn. You can apply a style for a single element, an entire webpage, or the entire website. It is device-friendly, too.
Browser compatibility and security are two areas that raise a concern. Similarly, different versions of CSS also create confusion. Developers are advised to check the compatibility before making any changes to the design.

JavaScript

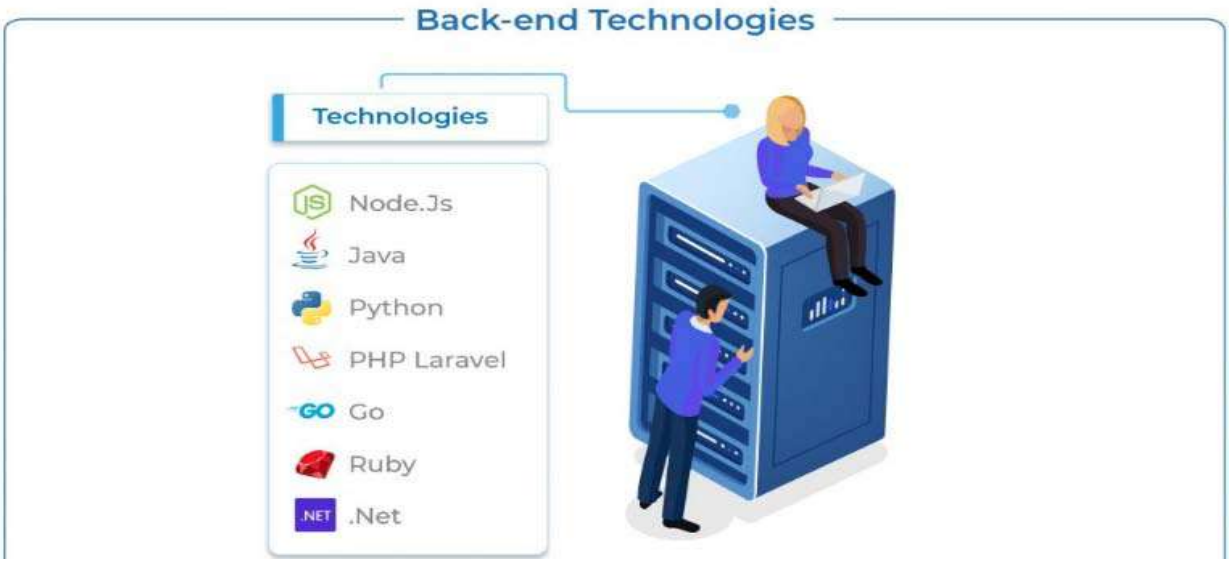
JavaScript or JS is the most popular client-side programming language, which has been used by more than 90% of websites recently. It was designed by Brendan Eich of Netscape in 1995. JavaScript uses a simple, easy-to-learn syntax. The language is so popular that every browser uses a JS engine to run JavaScript code on devices. It is easy to insert JS code on any web page, making it highly interoperable. It allows you to create rich interfaces to deliver

a better UI/UX experience. Being on the client side, JS reduces the server load as well.

However, developers should be careful about the security as the code is executed on the client side, which hackers can sometimes exploit.

React

React is an open-source JavaScript that has gained popularity in recent years. It was developed by Jordan Walke of Facebook in 2013. React benefits enable developers to easily create high-quality dynamic web applications with minimal code and effort.
ReactJS is easy to learn and use. There is extensive documentation and plenty of handy tools available for developers. The code is reusable. ReactJS uses a virtual DOM, which means concerned elements are updated when a change is made instead of the entire DOM tree being rewritten. It improves efficiency and optimizes memory usage. ReactJS uses a one-way Data flow, meaning changes made to the "child" elements do not affect the "parent" element. The code is easy to test and SEO-friendly.
On the downside, the ReactJS development environment is highly dynamic, so developers should proactively monitor the changes and quickly adapt new skills to leverage React. Moreover, React technologies are improving rapidly. However, the documentation cannot catch up with this pace. A critical area of concern is that ReactJS focuses on the UI part, and you need to depend on other libraries for client-side functionalities.



Application Layer: Server Instance / Cloud Instance

Servers or cloud instances are important to a web application architecture diagram. A cloud instance is a virtual server instance built, delivered, and hosted using a public or a private cloud and is accessible over the Internet. It works as a physical server that seamlessly moves across multiple devices or deploys multiple instances on a single server. As such, it is highly dynamic, scalable and cost-effective. You can automatically replace servers without application downtime. You can easily deploy and manage web applications in any environment using cloud instances.

Read our blog on How to Build an App Like Instagram.

Data Layer: Database

A database is a key component of a web application that stores and manages information for a web app. Using a function, you can search, filter and sort information based on user request and present the required info to the end user. They allow role-based access to maintain data integrity.
When choosing a database for your architecture of a web app, the size, speed, scalability, and structure are the four aspects that require your consideration. For structured data, SQL-based databases are a good choice. it suits financial apps wherein data integrity is a key requirement.
To handle unstructured data, NoSQL is a good option. It suits apps wherein the nature of incoming data is not predictable. Key Value databases associate each value with a key and suit databases that store user profiles, reviews, blog comments, etc. For analytics, Wide Column databases are a good choice.
Read our blog on the best programming languages to develop your application.

WIS Requirements Analysis

What is website requirement?

Website requirements are a list of necessary functions, capabilities, or characteristics related to your website and the plans for creating it. There are several types of requirements that may be defined during the process that come together to focus and prioritize the project plan

Website Requirements Analysis

Analysis of requirements is an iterative procedure that starts with a brainstorming meeting and continues through the course of development. In each iteration, the requirements document will improve and become increasingly useful for driving the implementation of the site. After completing the initial brainstorming sessions and identifying general requirements along with your client, a conceptual design should be developed using wireframes as an illustration of your potential approach.

Providing your clients with an example of the design that they can visualise, using wireframes, will enable them to more easily clarify and refine their requirements.

After formulating the initial requirements and reviewing, revising, and prioritising them with your clients, you should have follow up meetings and solicit further customer needs and wants.

These meetings will permit website users to validate the requirements and to determine if these requirements are commensurate with the needs of users. This information can be used by the analysis team to alter your incipient requirements document and develop a suitable web strategy.

In some instances, clients will be able to rapidly work with you to develop the requirements and come to a consensus on which functions and features are the highest priority.

However, more frequently, clients will need some guidance in order to solicit requirements that are detailed enough to be of use for your site development team.

If you are dealing with a client that has multiple stakeholders, obtaining a requirements consensus can be difficult. Obtaining a consensus on requirements often requires

- **Tact and skilful management of involved personalities; and**
- **Asking questions that are focused upon the priority of every requirement**

Querying stakeholders as to their rationale for their requirements requests is a great method to generate group discussions and to come to a determination as to whether a particular requirement should be a final design priority.

Within the process of building a consensus it is crucial that all members of the team are fully invested in ownership of the requirements and the process of analysis.

A large challenge associated with requirements analysis is obtaining requirements specificity from stakeholders, in order to make the requirements useful for people who are implementing the design and development.

Requirements that are most useful will describe in detail what functions are available to site visitors and provide guidelines on designing interactions that will occur on-site.

Website Development Process

The image layout we think at the time of design and developing it, is the messy task. We should follow process steps for design and development to get the finish line for the project.

Similar to the traditional software development life cycle, the website development process too can be subdivided into different steps. A detailed website development process can boost up your work and help your client understand your role in the project.

If you're a client or a business owner, this post is for you! After the 7+ years of experience, WebPixel Technologies knew the all the confusions and problems commonly happen during the development. So we wanted to share our website development process that will set your expectations on next website projects you might have and make the whole process easier to understand.

Innovative requirement

If the client has an idea, we determine the needs with providing custom innovative suggestions and let know client the whole process of development.

Input:

- Potential interview with the client, initial emails, proposals and supporting docs by the client, discussion notes.
- Recorded telephone conversations and Skype Chat
- Estimated Budget
- Portfolio Showcase

Output:

- Development process
- Estimating cost
- Team requirements (No of designers, developers, BA, QA, SEO etc)
- Hardware-software requirements
- Report documents
- Final client approval for the project.

Information Gathering

We define goals, analysis the requirements and make the project schedule

Input:

Reports from clients and documentations from Business Analyst

Output:

Complete final project documentation with requirement specifications and individual work define to the designers as well as developers.

Strategies & Planning

Planning the project is an essential step and it's even the most critical step more than development.

Input:

Final project documentation

Output:

clickable prototype and sitemap containing all WebPages

Web Design

We design the website that supports good look, feel and makes different from others

Input:

Wireframes

Output:

Site design with layout templates and images

Web Development

We build HTML templates and CSS, Code programming functionalities with developing new content data that appear on the site.

Input:

Website with forms and complete requirement specifications

Output:

Website with database driven functions, Coding docs

Testing

For review purpose, we launch the beta release and check the page speed and w3c validation

Input:

The site, Requirement specifications, supporting documents, technical specifications, and technical documents.

Output:

Complete website testing and error logs reports, frequent interaction with the developers and designers.

Launch & Maintenance

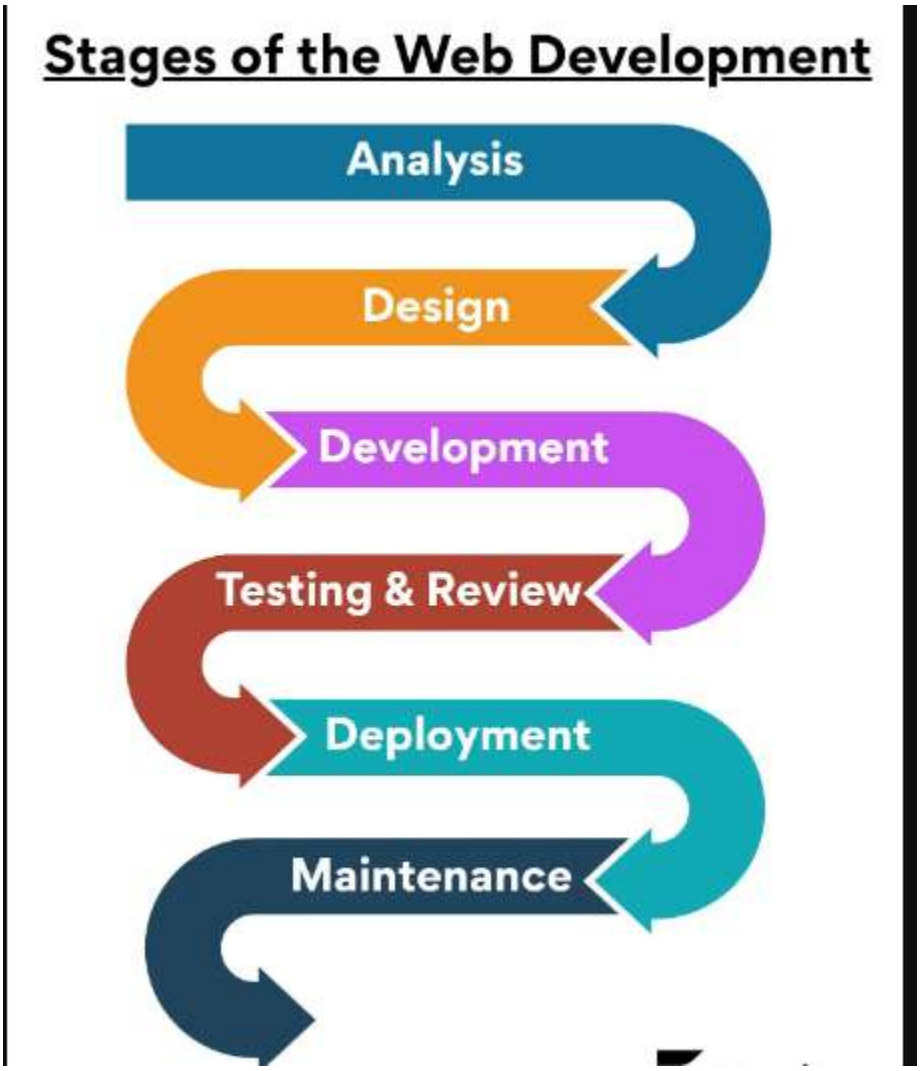
We get the final approval and go live. It's not over yet! We keep up to date website with updated content & latest technologies.

Input:

Live Website, Analysis reports.

Output:

Updated Website, Maintenance reports



WIS specification (Design and Development)

Web Specification

In essence, it describes a document that outlines the techniques and goals of a web project. It must highlight various constraints, including technological challenges, budget, or timeframe. Additionally, a specification can also outline other project details like the team.

How do you write a website requirements specification?

How to Write a Website Specification

1. **Start by introducing yourself. ...**
2. **Lay out your objectives. ...**
3. **Pull out your key audiences. ...**
4. **Your competition. ...**
5. **Website structure (don't worry, it's provisional) ...**
6. **The meaty part: functional specification. ...**
7. **Now for the non-functional requirements. ...**
8. **The good, the bad, and the ugly...**

What is a web design document?

Design documentation is a set of tools and guidelines that will help the owner of a site understand how to maintain the design and use backend

functionality to make updates and changes. This includes everything from logging into the website and changing a comma to adding content or substituting a logo.

What makes good web design?

PRINCIPLES OF GOOD WEBSITE DESIGN. An effective website design should fulfil its intended function by conveying its particular message whilst simultaneously engaging the visitor. Several factors such as consistency, colors, typography, imagery, simplicity, and functionality contribute to good website design.