1. **What is the Internet? Explain the Importance of Web Technology.**

**Ans:** The Internet is a global network of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to link devices worldwide. It is a vast system that allows computers around the world to communicate with one another, sharing information and resources.

**Information Access**: Web technology provides instant access to vast amounts of information on almost any topic. This has revolutionized education, research, and knowledge dissemination.

**Communication**: It enables seamless communication through emails, instant messaging, social media platforms, and video conferencing, breaking down geographical barriers.

**Commerce**: E-commerce platforms, powered by web technology, allow businesses to sell products and services online, reaching a global audience and operating 24/7.

**Entertainment**: The web offers a plethora of entertainment options, including streaming services, online games, and social media, transforming how people consume media and spend their leisure time.

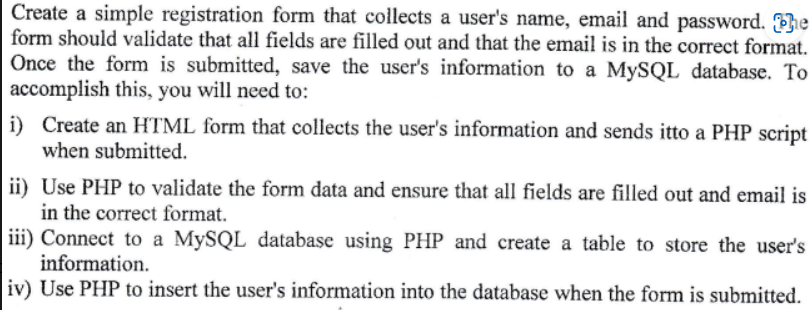
**Collaboration**: Web technologies facilitate collaboration through tools like cloud storage, project management software, and collaborative workspaces, enhancing productivity and teamwork.

**Healthcare**: Telemedicine and online health resources have improved access to medical advice and healthcare services, especially in remote areas.

**Government and Public Services**: Governments use web technologies to provide services to citizens, such as online tax filing, e-governance, and public information portals.

**Education**: Online courses, educational resources, and virtual classrooms have made education more accessible and flexible, catering to diverse learning needs.

**Social Connectivity**: Social networking sites and online communities enable people to connect, share experiences, and build relationships, fostering a sense of global community.



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registration Form</title>

</head>

<body>

<h2>Registration Form</h2>

<form action="register.php" method="POST">

<label for="name">Name:</label><br>

<input type="text" id="name" name="name" required><br><br>

<label for="email">Email:</label><br>

<input type="email" id="email" name="email" required><br><br>

<label for="password">Password:</label><br>

<input type="password" id="password" name="password" required><br><br>

<input type="submit" value="Register">

</form>

</body>

</html>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$name = $\_POST['name'];

$email = $\_POST['email'];

$password = $\_POST['password'];

// Basic validation

if (empty($name) || empty($email) || empty($password)) {

echo "All fields are required.";

exit;

}

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

echo "Invalid email format.";

exit;

}

// Hash the password

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "nce-project-2024";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Insert data into the database

$sql = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("sss", $name, $email, $hashed\_password);

if ($stmt->execute()) {

echo "Registration successful.";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$stmt->close();

$conn->close();

}

?>

Step 3: Connect to MySQL Database and Create Table

Ensure your database is set up correctly. You can create the users table using the following SQL query:

sql

Copy code

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL

);

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

email VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL

);

**3. Differentiating Between Web Frameworks and Normal Programming Languages.**

**Ans :** Programming Languages: The Foundation

* Core building blocks of software development.
* Define syntax, grammar, and semantics for writing code.
* Provide fundamental operations but lack specific tools for web applications.
* Examples: Python, JavaScript, Java, Ruby, C++, etc.

Web Frameworks: The Structure

* Specialized toolkits built on top of programming languages.
* Offer pre-built components, libraries, and conventions for web development.
* Accelerate development, improve code organization, and enhance security.
* Examples: Django (Python), React (JavaScript), Ruby on Rails (Ruby), Flask (Python), Express.js (JavaScript), etc

**3. Explain the role of XML documents and DTDs in the creation of structured data on the web.**

**Ans :** XML (Extensible Markup Language) documents and DTDs (Document Type Definitions) play a crucial role in the creation of structured data on the web. Here's an explanation of their roles:

**XML Documents:**

XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The primary purposes of XML are:

1. Data Storage and Transport:

XML is used to store and transport data. It allows the definition of custom tags that suit the specific needs of an application or context, facilitating the sharing of structured data across different systems and platforms.

1. Data Structure and Hierarchy:

XML documents provide a hierarchical structure to data through elements, attributes, and nesting. This structure makes it easier to organize, navigate, and manipulate data.

1. Platform Independence:

XML is platform-independent, meaning it can be used across different hardware, operating systems, and software applications. This makes it ideal for web applications where interoperability is crucial.

1. Self-Descriptive Format:

The tags in XML documents are self-descriptive, meaning that each tag clearly defines the type of data it contains. This enhances readability and understanding, both for humans and machines.

**DTDs (Document Type Definitions):**

DTDs define the structure and the legal elements and attributes of an XML document. They serve several important functions:

1. Validation:

DTDs are used to validate the structure of an XML document. They ensure that the document adheres to the predefined rules and structure, which helps maintain data integrity and consistency.

1. Specification of Document Structure:

A DTD specifies the allowed elements and attributes, their relationships, and the order in which they appear in the document. This provides a clear blueprint for what the XML document should look like.

1. Interoperability:

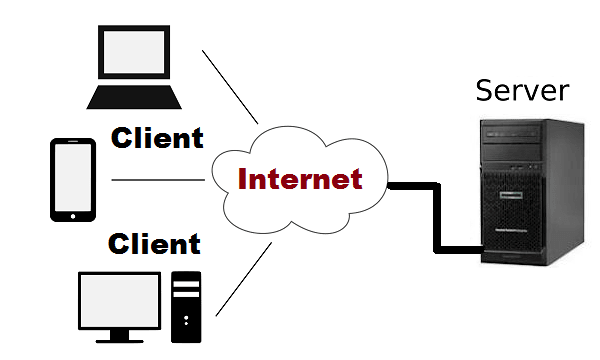
By providing a common structure, DTDs ensure that different systems and applications can understand and process the XML documents in a consistent manner. This is particularly important in web services and applications that need to exchange data.

1. Reusability and Modularity:

DTDs promote reusability and modularity by allowing the definition of reusable components and entities. This makes it easier to manage and maintain large and complex XML documents.

In summary, XML documents and DTDs are fundamental to the creation and management of structured data on the web. XML provides a flexible and readable way to encode data, while DTDs ensure that the data is well-structured and adheres to specific rules, facilitating validation, interoperability, and consistency.

1. **Explain with the diagram of client-server architecture.**

Ans:

The Client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and delivers the data packets requested back to the client. Clients do not share any of their resources.

* **Client:** When we say the word **Client**, it means to talk of a person or an organization using a particular service. Similarly in the digital world, a **Client** is a computer (**Host**) i.e. capable of receiving information or using a particular service from the service providers (**Servers**).
* **Servers:** Similarly, when we talk about the word **Servers**, It means a person or medium that serves something. Similarly in this digital world, a **Server** is a remote computer that provides information (data) or access to particular services.

1. **Explain how CGI server-side script execution works.**

CGI (Common GatewayInterfaces) The Common Gateway Interface (CGI) is a standard (protocol) for providing an interface, or a gateway, between an information server and an external process (that is, a process external to the server)

CGI scripts are executables that will execute on the server to produce dynamic and interactive web pages.

A diagram of a program

Description automatically generated

**Working Principle of CGI:**

* The browser sends an HTTP GET or POST command to the web server with the parameters, along with the path to the requested document (which in this case is a CGI program, although the browser doesn't know or care about this distinction.
* The web server notes that the requested document is a CGI program rather than a static HTML document.
* The web server initializes certain environment variables with the passed parameters runs the CGI program in a newly-created OS process, and the CGI program in turn produces an HTML document as its output, making use of the parameters as necessary.
* The web server returns the generated HTML document to the browser, passing along the header information as bs before.

1. **What is a content management system? Explain information system at different levels of management?**

Ans: A content management system (CMS) is an application that is used to manage web content, allowing multiple contributors to create, edit and publish.

Web information system, or web-based information system, is an information system that uses Internet web technologies to deliver information and services, to users or other information systems/applications.

It is a software system that whose primary aim is to publish and maintain data by using hypertext- based principles.

1. Cross platform compatibility:

Most web-based applications are compatible in different platforms than traditional installed software. The minimum requirement would be a web browser (Internet Explorer, Firefox, Netscape etc.). You can use different OS such as Windows, Linux or Mac to run the web applications.

2. More Manageable:

WBIS only need to be installed on the server placing minimal requirements on the end user workstation, which makes the system easier to maintain and update as usually it can all be done on the server.

3. Multiple concurrent users:

Web-based applications can indeed be used by multiple users at the same time. It’s not necessary to share screen or send a screenshot when multiple users see and even edit the same document at the same time. Web conferencing and online collaboration companies regulate some key transformations and users only explore what they really need to work effectively and co-edit documents together.

4. Reduced cost:

Web-based applications can reduce cost due to support and maintenance, lower requirements on the end user system and simplified architecture. It doesn’t require any distribution or marketing infrastructure.

5. Secure live data:

These applications can decrease the risk of losing data due to an unexpected disk crash or computer virus. Companies of web-based applications provide extensive data backup service either as an integral part or basic service or sometimes as a paid service.

1. **Define XML. Write internal and external DTD to describe "mail" as root element and "to" "from", "subject" and "message" as the child elements.**

Internal DTD:

<!DOCTYPE mail [

<!ELEMENT mail (to, from, subject, message)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT subject (#PCDATA)>

<!ELEMENT message (#PCDATA)>

]>

<mail>

<to>recipient@example.com</to>

<from>sender@example.com</from>

<subject>Important Message</subject>

<message>This is the message body.</message>

</mail>

External DTD

<!ELEMENT mail (to, from, subject, message)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

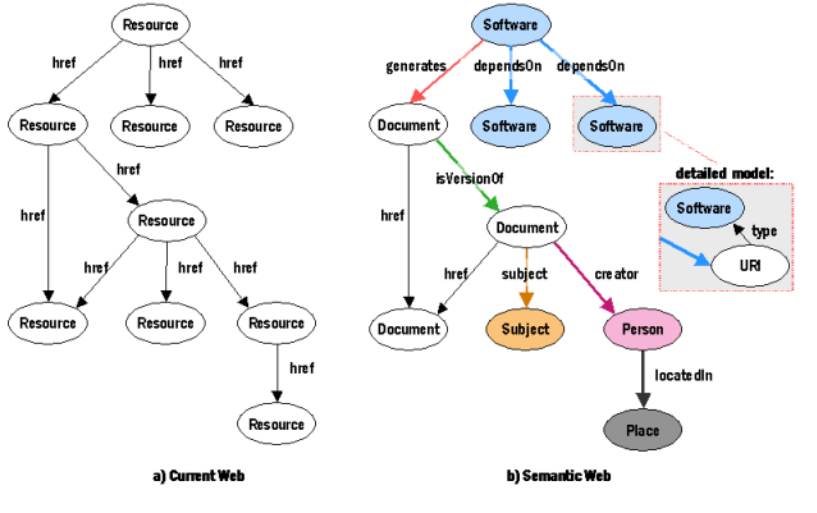
<!ELEMENT subject (#PCDATA)>

<!ELEMENT message (#PCDATA)>

What is semantic web explain with the example?

Ans :

Semantic Web is an extension to the World Wide Web (web 3.0). The purpose of the semantic web is to provide structure to the web and data in general. It emphasizes on representing a web of data instead of web of documents. It allows computers to intelligently search, combine and process the web content based on the meaning that the content has. The semantic web provides more context to websites so that not only humans but also machines can interpret the purpose of the content and data on the web to fetch and deliver coherent results.



1. **Explain the semantic web component RDF and OWL.**

RDF is the fundamental language for representing information in the Semantic Web. It provides a basic structure for describing resources and their relationships.

Key components of RDF:

**Resources**: Anything that can be identified by a URI (Uniform Resource Identifier), such as a person, place, thing, or concept.

**Properties**: Describe characteristics or relationships between resources.

**Statements**: Express facts about resources, consisting of a subject (resource), a predicate (property), and an object (resource or literal).

Example:

Subject: John Doe

Predicate: hasAge

Object: 30

In RDF, this would be represented as a triple: (John Doe, hasAge, 30).

* OWL (Web Ontology Language):

OWL is a more complex language built on top of RDF. It provides a richer vocabulary for defining classes, properties, and individuals, as well as expressing complex relationships between them. OWL allows for reasoning and inference, which means that computers can deduce new information from the knowledge base.

Key features of OWL:

Classes: Define groups of individuals with common properties.

Properties: Describe relationships between individuals and classes.

Individuals: Represent specific instances of classes.

Reasoning: Allows for inferring new information based on existing knowledge.

Example:

Class: Person

Property: hasName

Individual: John Doe

OWL can also express more complex relationships, such as subclass relationships, disjoint classes, and cardinality constraints.

1. **What do u mean by server-side processing? Explain multiplier architecture.**

* **Server-Side Processing**

Think of a website as a conversation between you (the client) and a powerful computer (the server). When you interact with a website, your actions and requests are sent to the server. Server-side processing takes place entirely on that server's end.

Here's what happens:

You make a request: You click a button, submit a form, or simply load a webpage. This request travels from your computer (client) to the website's server.

Server processes: The server receives your request and performs the necessary actions. This could involve:

Fetching data from a database

Running calculations or algorithms

Generating dynamic content (like personalized recommendations)

Interacting with other systems or services

Response sent back: The server then compiles the results of its processing and sends them back to your computer. This is what you see as the webpage loading, displaying information, or updating in response to your actions.

Key Points:

Behind the scenes: All the heavy lifting of processing and logic happens on the server, hidden from the user's view.

Dynamic content: Server-side processing allows for creating web pages that change based on user input or other factors, making websites more interactive and personalized.

Security: Sensitive data processing often occurs on the server, adding a layer of security as user devices are not directly handling critical information.

**Multitier Architecture:**

Imagine a website as a complex building with multiple floors. Multitier architecture is like organizing the work of that building into different specialized floors or "tiers." Each tier has its own role:

Presentation Tier: This is what you see and interact with - the website's user interface. It runs on your computer (client-side).

Logic Tier: This tier handles the website's business logic and application processing. It usually resides on one or more servers.

Data Tier: This tier stores and manages the website's data, typically in databases. It's also located on servers, often separate from the logic tier for better performance and security.

Benefits of Multitier Architecture:

Scalability: You can easily add more servers to each tier to handle increased traffic or workload.

Maintainability: Changes to one tier don't necessarily affect others, making development and updates easier.

Security: Sensitive data is isolated in the data tier, enhancing protection.

Performance: Distributing work across multiple tiers can improve website speed and responsiveness.