BLACK N WHITE

Learn Today Lead Tomorrow

jag....

NAME	
ROLL NUMBER	
SEMESTER	
COURSE CODE	
COURSE NAME	

Question 1 . A) Brief the role of search engine and plug-ins.

Answer 1 . A :-

Search Engine : A search is a software program or online service that helps user's to find information on the internet . Its primary role is to crawl , index, and retrieve web pages based on a user's query .

Here's a brief of its role :

1.) Crawling :- Search engines use automated bots , often called "spiders" or "crawlers," to systematically browse the web and discover web pages. These bots follows links one page to another , building a database of web content .
2.) Indexing :- Once web pages are crawled , search engines create an index, which is like a massive database of information about each page. This indexes keywords , metadata and other relevant data to help with search queries .
3.) Retrieval :- When a user enters a search query , the search engine it's index to quickly identify relevant web pages . Algorithms are applied to determine the most relevant results based on factors like keyword relevance , and user behavior .
4.) Ranking :- Search engines often present search result in order of relevance . They use complex algorithms to rank web pages, ensuring that the most relevant and high-quality content appears at the top of the results .

➔ **Plug-ins :-** Plug-ins, also known as extension or add-ons , are software components that add specific functionalities or fractures to an existing c0mputer program or application . Their role varies depending on the context , but generally , they serve to enhance or extend the capabilities of a base software . Here's a brief of their role :-

1.) Enhancing Functionality :- Plug-ins are used to extend the capabilities of software applications. For example : web browser like Chorme and Firefox support plug-ins that can add features like ad-blocking , password management , or language translation .
2.) Customization : User can tailor software to their specific needs by installing plug-ins . This allows for a personalized user experience and can increase productivity or convenience .
3.) Integration : Plug-ins can integrate with external services or Apls, enabling seamless interaction between different software system .
4.) Modular Approach :- Using plug-ins allows developers to create modular software , making it easier to update and maintain . Users can choose which plug-ins to install , reducing bloat and improving performance .

In summary , search engines help user find information on the internet by crawling , indexing , and retrieving web content , while plug-ins enhance and extend the functionalities of software applications by adding specific features or customizations .

SET-I

Question 1 . B) What is markup language and briefly explain any two popular markup language .

Answer 1 . B) A markup language is a system of annotating text to add structure, formatting, or metadata to the content. It involves using special codes or tags embedded within the text to define various elements and their attributes. Markup languages are commonly used in various fields, including web development, document formatting, and data representation

1. HTML (Hypertext Markup Language):- HTML is the standard markup language used for creating web pages and web applications. It defines the structure and content of a webpage by using various elements and attributes. HTML tags are enclosed in angle brackets (< >) and often come in pairs, with an opening tag and a closing tag, surrounding the content they affect. For example, `<p>` represents a paragraph element, `<a>` represents a hyperlink, and `<img>` represents an image.

2. XML (Extensible Markup Language):- XML is a versatile markup language used for structuring and storing data in a human-readable format. Unlike HTML, XML allows you to define your custom tags and document structures, making it highly adaptable for various data exchange and storage purposes. XML documents are hierarchical and must have a single root element. Elements are enclosed in angle brackets and can have attributes to provide additional information.

Both HTML and XML serve essential roles in organizing and presenting data, but HTML is primarily used for creating web content, while XML is used for data storage, interchange, and configuration purposes.

Question 2) State the functions of DNS and protocol used. Explain the way in which a DNS server resolves the address ?

Answer 2) :- DNS (Domain Name System) is a critical internet infrastructure that translates user-friendly domain names into IP addresses and vice versa. It serves several vital functions:

i) Hostname to IP Address Resolution: DNS maps domain names (e.g., www.example.com) to IP addresses (e.g., 192.0.2.1), facilitating user access to websites.

ii) IP Address to Hostname Resolution: DNS supports reverse DNS lookup, finding domain names associated with IP addresses.

iii) Load Balancing: DNS can distribute traffic across multiple servers or data centers for optimal performance.

iv) Redundancy and Failover: Multiple IP addresses for a domain provide redundancy and failover capabilities.

v) Mail Server Routing: DNS specifies MX records, directing email to the appropriate mail servers.

iv) Caching: DNS servers cache resolved queries to reduce response times and lessen the load on authoritative servers.

DNS primarily uses UDP for its low overhead and speed, switching to TCP for larger responses or specific cases.

DNS resolution involves these steps:

- Local Cache Check: Device checks its local DNS cache for the IP address.
- Local DNS Server: Device queries a local DNS server, which may have the answer in its cache.
- Recursive DNS Resolution: Local server, if necessary, contacts authoritative DNS servers hierarchically, starting from root servers, to obtain the IP address.
- Response to User: The local DNS server stores the IP address in its cache and returns it to the device.
- User Accesses the Server: The device establishes a connection to the web server via the obtained IP address, enabling access to the desired content.

DNS simplifies internet navigation by enabling users to use domain names instead of numerical IP addresses, playing a vital role in the internet's user-friendliness and efficiency.

Question 3 . A) How would you insert an image file named elephant.jpg at the very top of a web page ? Write Code in HTML

Answer 3.A):-

To insert an image file named "elephant.jpg" at the very top of a web page, you can use the HTML <img> element within the <body> section of your HTML document.

Here's the HTML code to achieve this:

```
<!DOCTYPE html>
<html>
<head>
   <title>Your Web Page Title</title>
</head>
<body>
   <img src="elephant.jpg" alt="An elephant"
style="display: block; margin: 0 auto;">
</body>
</html>
```

In this code:

- <img> :- Tis is the image element . It has two important attributes :
  i) src:- Tis attribute specifies the source (URL or file path) of the image , in this case, “elephant.jpg”.
  ii) alt :- This attribute provide alternative text for image , which is important for accessibility . It should describe the content or purpose of the image .
- style :- This inline CSS style is used to centre the image horizontally on the page . The ‘display: block; ‘ property makes the image a block-level element , and ‘margin :0 auto ;’ centres it horizontally by setting it’s left and right margins to “auto”.

Question 3 . B) What is the difference between the external style sheet and embedded style sheet ?

Answer 3.B):- External style sheets and embedded style sheets are two common methods for applying CSS (Cascading Style Sheets) to HTML documents. Here are the key differences between them:

1.) Location :-

- External Style Sheet: An external style sheet is a separate .css file that is linked to an HTML document using the <link> element in the document's <head> section. The CSS rules are defined in this external file, and multiple HTML documents can reference and share the same external style sheet. This approach promotes consistency across multiple web pages.

- Embedded Style Sheet :- An embedded style sheet, also known as an internal or <style> style sheet, is defined within the <style> element directly in the HTML document's <head> section. Each HTML document that uses this style sheet will have its CSS rules within its own <style> element. It's typically used when styling is specific to a single webpage.

2.) Ease of Maintenance :-

- External Style Sheet: External style sheets are easier to maintain because changes to the styling can be made in one central .css file, and those changes will automatically apply to all HTML pages that link to it. This is particularly useful for maintaining a consistent look and feel across a website.

- Embedded Style Sheet: Embedded style sheets require editing each individual HTML file to make changes to the styling. This can become cumbersome and error-prone, especially for larger websites with many pages.

3.) Flexibility :-

- External Style Sheet: External style sheets offer the most flexibility in terms of reusability. You can use the same style sheet across multiple websites if needed, and it allows for a clear separation of content and presentation.

- Embedded Style Sheet: Embedded style sheets are specific to the HTML document in which they are defined. They are less flexible for reuse and are generally better suited for cases where the styling is unique to a single page.

4.) Load Time :-

- External Style Sheet: The external style sheet is cached by the browser after the initial load, so subsequent pages referencing the same style sheet will load faster because they don't need to download the CSS file again.

- Embedded Style Sheet: The CSS rules are part of the HTML document itself, so they are loaded each time the HTML page is accessed. This can result in slightly longer load times for each page, especially if the CSS is extensive.

In summary, the choice between using an external style sheet and an embedded style sheet depends on your specific requirements. Use an external style sheet for consistency and ease of maintenance across multiple pages or websites. Use an embedded style sheet when styling is unique to a single page or when you want to experiment with specific styles within a single document.

SET-II

Question 4 A.) Differentiate between get request type and post request type.

Answer:- GET and POST are two common HTTP request methods used for communication between a client (typically a web browser) and a server. They serve different purposes and have distinct characteristics:

1.) **Purpose:-**

- GET: The GET request method is used to request data from a specified resource. It is primarily used for retrieving information and should not have any significant side effects on the server. GET requests are idempotent, meaning making the same GET request multiple times should produce the same result.

- POST: The POST request method is used to submit data to be processed to a specified resource. It is often used when the client wants to send data to the server, such as form submissions, file uploads, or data to be processed for creating, updating, or deleting resources. POST requests are not idempotent as they can have different effects on the server each time they are sent.

2.) Data Sending :-

- GET: Data is sent in the URL as query parameters. It appends data to the URL, making it visible in the browser's address bar. GET requests are limited in the amount of data they can send, and data is not as secure because it's exposed in the URL.

- POST: Data is sent in the request body. It is not visible in the URL, making it a more secure method for sending sensitive information. POST requests can handle larger data payloads.

3.) Caching :-

- GET: GET requests are generally cached by browsers and intermediaries (like proxy servers), which can lead to improved performance when the same resource is requested multiple times.

- POST: POST requests are typically not cached because they are considered non-idempotent, and their results may vary with each request.

4.) Bookmarking And History :-

- GET: Since GET requests are typically used for retrieving data, the resulting URLs can be bookmarked and saved in browser history.

- POST: POST requests are not suitable for bookmarking or saving in browser history because they represent actions that change the server's state.

5.) Security :-

- GET: Data is visible in the URL, which can be a security risk for sensitive information. It is less secure for transmitting confidential data.

- POST: Data is not visible in the URL, making it a more secure option for transmitting sensitive data.

In summary, GET requests are used for retrieving data, are visible in the URL, and are typically cached. POST requests are used for submitting data, keep data hidden in the request body, and are not cached. Choosing between GET and POST depends on the specific use case and the nature of the data being transferred.

Question 4 B.) How can you create JDBC statements ?

Answer 4 B.) Creating JDBC Statements In Java, you can create JDBC (Java Database Connectivity) statements to interact with relational databases. JDBC provides three types of statements: Statement, PreparedStatement, and CallableStatement.

1.) Statement :-

- Usage: The Statement interface is used to execute simple SQL queries without parameters.
- When to Use: Use Statement when your SQL queries are straightforward and don't involve parameters.
- Key Points: Statements can be vulnerable to SQL injection if user input is directly incorporated into queries. Therefore, it's important to sanitize user input or use other methods to prevent SQL injection.

2.) PreparedStatement:-

- Usage: The PreparedStatement interface is used to execute parameterized SQL queries. It allows you to provide input parameters safely.
- When to Use: Use PreparedStatement when your SQL queries involve parameters (e.g., user input) to prevent SQL injection attacks and improve query performance.
- Key Points: Prepared statements are precompiled, which can lead to better performance when executing the same query with different parameter values. They are more secure because they automatically handle parameterized input.

3.) CallableStatement :

- Usage: The CallableStatement interface is used to call stored procedures in a database or execute some special SQL operations specific to the database system.
- When to Use: Use CallableStatement when you need to execute stored procedures or perform advanced database-specific operations.
- Key Points: Callable statements are essential when working with databases that use stored procedures extensively. They allow you to pass input parameters and retrieve output parameters from stored procedures.

➔ General Steps for Creating JDBC Statements:

i) Establish a Database Connection: Before creating any JDBC statement, establish a connection to your database using the Connection class. You'll need to specify the database URL, username, and password.

ii) Create the Statement: Depending on your requirements, create a Statement, PreparedStatement, or CallableStatement object. Pass your SQL query as a parameter to the appropriate constructor or method.

iii) Set Parameters (if applicable): For PreparedStatement and CallableStatement, set parameter values using setter methods. This is essential when your SQL query contains placeholders for parameterized input.

iv) Execute the Statement: Use the appropriate method (execute, executeQuery, or executeUpdate) to execute your statement.

v) Retrieve Results (if applicable): If your statement returns results (e.g., SELECT queries), use methods to retrieve and process the result set.

vi) Close Resources: Always close the statement and the database connection when you're done to release resources and prevent resource leaks.

In summary, JDBC statements are crucial for interacting with relational databases from Java applications. Choose the appropriate statement type based on your SQL query and parameterization needs. Prepared statements are recommended for security and performance reasons. Proper exception handling and resource management are essential for reliable database interactions.