**Java Web Application**

Web Application-

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.

Example-Apache, IIs

Example- IE, Chrome, Safari, Firefox

**HTTP**

**response**

**request**

Web Client

Web Server

To access web site used URL(Uniform Resource Locator) it is comprised of a protocol which is HTTP.

URL- https://www.example.com:80

Protocol domain name port no(default port)

Static Web Application-

A Static Web Application is any web application that can be delivered directly to an end user's browser without any server-side alteration of the HTML, CSS, or JavaScript content.

HTML files

Image Files

Multimedia Files

**Web Server**

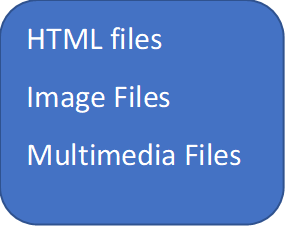
**HTTP Request**

Browser

**HTTP Response**

Dynamic Web Application-

A dynamic web application generates the pages/data in real time, as per the request, a respective response will trigger from the server end and will reach the client end.



**JDBC**

**Database**

**HTTP Request**

**HTTP Response**

Server Side Programming-

The server side programming is the name given to all types of programs which run on the web server. They process the user input, interact with the databases and control what content is served back to the client as a response to his request.

* Capturing the user input
* Communicate with the database
* Processing of data
* Procedure the response page
* Handling the response page to the server

Servlet-

Servlets are the Java programs that run on the Java-enabled web server or application server. They are Generic and HTTP servlets.

Three life cycle service-

1. init() – 1 times called
2. service()- n thimes called
3. destroy – 1 time called

four life cycle phases associated with these three methods

1.instantiation

2.initalization

3.servicing

4.destruction

The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.

JDBC-

**Java Database Connectivity** (JDBC) is an Application Programming Interface (API), from Sun microsystem that is used by the Java application to communicate with the relational databases from different vendors. JDBC and database drivers work in tandem to access spreadsheets and databases. **Design of JDBC** defines the components of JDBC, which is used for connecting to the database.

1. JDBC Client –

The JDBC client provides access to the data server from Java and Java-based tools.

1. JDBC API-

JDBC API provides various interfaces and methods to establish easy connection with different databases.

1. JDBC Driver-

JDBC Drivers are used to connect the database drivers to the database.

1. Driver Manager-

JDBC Driver manager loads the database-specific driver into an application in order to establish the connection with the database. The JDBC Driver manager is also used to make the database-specific call to the database in order to do the processing of a user request.

These steps are as follows:

* Register the Driver class –

The **forName()** method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

Syntax of forName() method-

**public** **static** **void** forName(String className)**throws** ClassNotFoundException

* Create connection-

The **getConnection()** method of DriverManager class is used to establish connection with the database.

Syntax of getConnection() method-

1) **public** **static** Connection getConnection(String url)**throws** SQLException

2) **public** **static** Connection getConnection(String url,String name,String password)  **throws** SQLException

* Create statement-

The createStatement() method of Connection interface is used to create statement. The object of statement is responsible to execute queries with the database.

Syntax of createStatement() method

**public** Statement createStatement()**throws** SQLException

* Execute queries-

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

Syntax of executeQuery() method

public ResultSet executeQuery(String sql)throws SQLException

* Close connection-

By closing connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

Init Param-

The init-param sub-element of servlet is used to specify the initialization parameter for a servlet.

Servlet Context-

An object of ServletContext is created by the web container at time of deploying the project. This object can be used to get configuration information from web.xml file. There is only one ServletContext object per web application.

1. The object of ServletContext provides an interface between the container and servlet.
2. The ServletContext object can be used to get configuration information from the web.xml file.
3. The ServletContext object can be used to set, get or remove attribute from the web.xml file.
4. The ServletContext object can be used to provide inter-application communication.

Prepared Statements-

The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query.

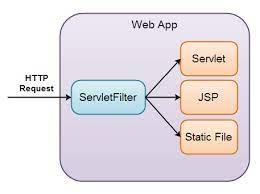
Inter Servlet Communication-

A process where two or more servlets communicates with each other to process the client request.A servlet can *forward* the request to another servlet to process the client request.A servlet can *include* the output of another servlet to process the client request.

The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

Servlet Filter-

The servlet filter is pluggable, i.e. its entry is defined in the web.xml file, if we remove the entry of filter from the web.xml file, filter will be removed automatically and we don't need to change the servlet.



Session Management-

Session management refers to the process of securely handling multiple requests to a web-based application or service from a single user or entity. Websites and browsers use HTTP to communicate, and a session is a series of HTTP requests and transactions initiated by the same user.

JSP-

JavaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

JSP Action-

1. **jsp:include**

Includes a file at the time the page is requested.

1. **jsp:useBean**

Finds or instantiates a JavaBean.

1. **jsp:setProperty**

Sets the property of a JavaBean.

1. **jsp:getProperty**

Inserts the property of a JavaBean into the output.

1. **jsp:forward**

Forwards the requester to a new page.

MVC Design Pattern-

The model-view-controller (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

Connection Pooling-

Connection pooling means that connections are reused rather than created each time a connection is requested. To facilitate connection reuse, a memory cache of database connections, called a connection pool, is maintained by a connection pooling module as a layer on top of any standard JDBC driver product.