JAVA WEB DEVELOPMENT

Web Application

A web-application is an application program that is usually stored on a remote server, and users can access it through the use of **Software** known as **web-browser.**

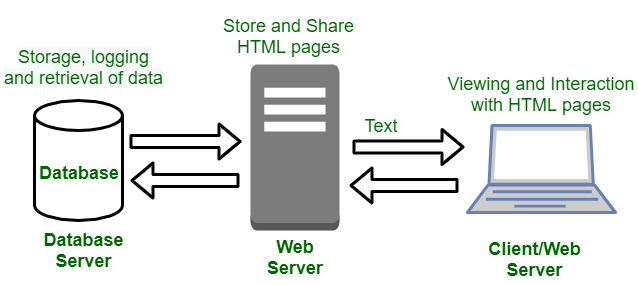
A web application can be developed for several uses, which can be used by anyone like it can be used as an individual or as a whole organization for several reasons.

web application is a computer program that usually resides on the remote server. Any user can access it by using one of the standard web browsers such as [Google Chrome](https://www.javatpoint.com/google-chrome)**, Safari, Microsoft Edge, etc.,** and most of them are available free for everyone. A web application are generally coded using the languages supported by almost every web-browsers such as HTML, JavaScript because these are the languages that rely on the web browsers to render the program executable.

 In Static Websites, Web pages are returned by the server which are prebuilt source code files built using simple languages such as HTML, CSS, or JavaScript. There is no processing of content on the server (according to the user) in Static Websites. Web pages are returned by the server with no change therefore, static Websites are fast. There is no interaction with databases. Also, they are less costly as the host does not need to support server-side processing with different languages.

In Dynamic Websites, Web pages are returned by the server which are processed during runtime means they are not prebuilt web pages but they are built during runtime according to the user’s demand with the help of server-side scripting languages such as PHP, Node.js, ASP.NET and many more supported by the server. So, they are slower than static websites but updates and interaction with databases are possible.

Dynamic Websites are used over Static Websites as updates can be done very easily as compared to static websites (Where altering in every page is required) but in Dynamic Websites, it is possible to do a common change once and it will reflect in all the web pages. 



*Architecture of Dynamic Website*

| Static Website | Dynamic Website |
| --- | --- |
| Content of Web pages can not be change at runtime. | Content of Web pages can be changed. |
| No interaction with database possible. | Interaction with database is possible |
| It is faster to load as compared to dynamic website. | It is slower then static website. |
| Cheaper Development costs. | More Development costs. |
| No feature of Content Management. | Feature of Content Management System. |
| HTML, CSS, Javascript is used for developing the website. | Server side languages such as PHP, Node.js are used. |

[Next](https://www.geeksforgeeks.org/php-vs-node-js/)

# JDBC

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

* JDBC-ODBC Bridge Driver,
* Native Driver,
* Network Protocol Driver, and
* Thin Driver

## **What is API**

API (Application programming interface) is a document that contains a description of all the features of a product or software. It represents classes and interfaces that software programs can follow to communicate with each other. An API can be created for applications, libraries, operating systems, etc.

# Java Database Connectivity with MySQL

1. **Driver class:**The driver class for the mysql database is **com.mysql.jdbc.Driver**.
2. **Connection URL:**The connection URL for the mysql database **is jdbc:mysql://localhost:3306/surya**where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.
3. **Username:**The default username for the mysql database is root.
4. **Password:**It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

# Servlets

**Servlet** technology is used to create a web application (resides at server side and generates a dynamic web page).

**Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology

# What is a Servlet?

Servlet can be described in many ways, depending on the context.

* Servlet is a technology which is used to create a web application.
* Servlet is an API that provides many interfaces and classes including documentation.
* Servlet is an interface that must be implemented for creating any Servlet.
* Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
* Servlet is a web component that is deployed on the server to create a dynamic web page.

# JSP

**JSP** technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

# Advantages of JSP over Servlet

#### **Extension to Servlet**

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

#### **Easy to maintain**

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

#### **Fast Development**: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

#### **Less code than Servlet**

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

# The Lifecycle of a JSP Page

The JSP pages follow these phases:

* Translation of JSP Page
* Compilation of JSP Page
* Classloading (the classloader loads class file)
* Instantiation (Object of the Generated Servlet is created).
* Initialization ( the container invokes jspInit() method).
* Request processing ( the container invokes \_jspService() method).
* Destroy ( the container invokes jspDestroy() method).

# MVC in JSP

**MVC** stands for Model View and Controller. It is a design pattern that separates the business logic, presentation logic and data.

**Controller** acts as an interface between View and Model. Controller intercepts all the incoming requests.

**Model** represents the state of the application i.e. data. It can also have business logic.

**View** represents the presentaion i.e. UI(User Interface).