**Basics of Web Application**

 Java Web Application is used **to build dynamic websites**. Java offers support for the web application through JSPs and Servlets. We can build a website with static HTML web pages but when we want data to be dynamic, we require the web application.

A Web application **contains an application's resources, such as servlets, JavaServer Pages (JSPs), JSP tag libraries, and any static resources such as HTML pages and image files**. A Web application can also define links to outside resources such as Enterprise JavaBeans (EJBs).

**Static and Dynamic web application**

A static website is a website whose web pages are coded in HTML and the content of each page is fixed and does not change unless it is edited and republished. A dynamic website is a website whose web pages are generated in real time.

***server-side programming***

Server-side programming **allows developers to make use of sessions** — basically, a mechanism that allows a server to store information on the current user of a site and send different responses based on that information

**Servlet Life Cycle Methods**

There are three life cycle methods of a Servlet :

* init() -This method is called by the Servlet container to indicate that this Servlet instance is instantiated successfully and is about to put into service.
* service() - This method handles multiple client request and sends response back.
* destroy() -  This method runs only once during the lifetime of a Servlet

Servlets - Annotations

**@WebServlet -** To declare a servlet.

**@WebInitParam -** To specify an initialization parameter.

**@WebFilter -** To declare a servlet filter.

**@WebListener -** To declare a WebListener

Flow of Execution of Servlets

* Create “*Dynamic web application”*project in Eclipse.
* Create “*index.html”*file to take input from the client in the request.
* Create “*ServletName.java”*to process the request and generate the response.
* Create “*web.xml”*file to specify the URL mapping to the respective servlet.

JDBC Architecture

* + JDBC architecture is **an API specifying interfaces for accessing relational databases**.
  + JDBC helps to connect to a database, send queries and updates to the database, and retrieve and process the results obtained from the database for queries
  + JDBC API provides universal data access from the Java programming language. Using the **JDBC API**, you can access virtually ...

JDBC Driver

JDBC Driver is a software component that enables java application to interact with the database.

JDBC client

The JDBC client **provides access to the data server from Java™ and Java-based tools**. The JDBC client is distributed as a JAR (Java archive) file.

DriverManager

DriverManager manages the set of Java Database Connectivity (JDBC) drivers that are available for an application to use.

Applications can use multiple JDBC drivers concurrently if necessary.

CRUD Operations

1. Create- The create function allows users to create a new record in the database.
2. Read- The read function is similar to a search function.
3. Update- The update function is used to modify existing records that exist in the database.
4. Delete.

JDBC Statements

* **Create Statement** -  It is generally used for general**–**purpose access to databases and is useful while using static SQL statements at runtime.Statement

Syntax:

statement = connection.createStatement();

* **Prepared Statement**- represents a recompiled SQL statement, that can be executed many times. This accepts parameterized SQL queries. In this, “?” is used instead of the parameter, one can pass the parameter dynamically by using the methods of PREPARED STATEMENT at run time.

Get vs. Post

**GET-**

In case of Get request, only **limited amount of data**can be sent because data is sent in header.

Get request is **not secured**because data is exposed in URL bar.

 It requests the data from a specified resource

**POST-**

In case of post request, **large amount of data**can be sent because data is sent in body.

Post request is **secured**because data is not exposed in URL bar.

It submits the processed data to a specified resource