

Mod 3

Q1) Write a short note on JDBC

⇒ ① JDBC also known as Java database connectivity is a Java-based API that allows java applications to interact with databases

② JDBC acts as a bridge between Java applications and databases , enabling the retrieval and manipulation of data

③ Components of jdbc :

i) DriverManager

(a) Manages database drivers & establishes connection between the java application and the database.

(b) eg:

```
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/db_name", "username", "password");
```

ii) Connection

(a) Represents a connection to the database
(b) Used to create statements and manage transactions.

iii) Statement

(a) Executes SQL queries

iv) ResultSet

(a) Stores the result retrieved from the database after a query execution.

(b) Example:

```
ResultSet rs = stmt.executeQuery("SELECT * FROM users");
```

④ Steps to use JDBC

i) Load the database driver

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

ii) Establish a connection to the database

```
Connection conn = DriverManager.getConnection(
```

```
"jdbc:mysql://localhost:3306/db-name", "username",  
"password");
```

iii) Create a statement

```
Statement stmt = conn.createStatement();
```

iv) Execute SQL queries

```
Result rs = stmt.executeQuery("SELECT *  
from users");
```

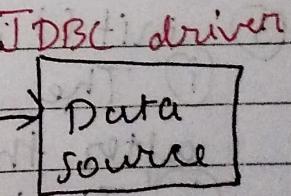
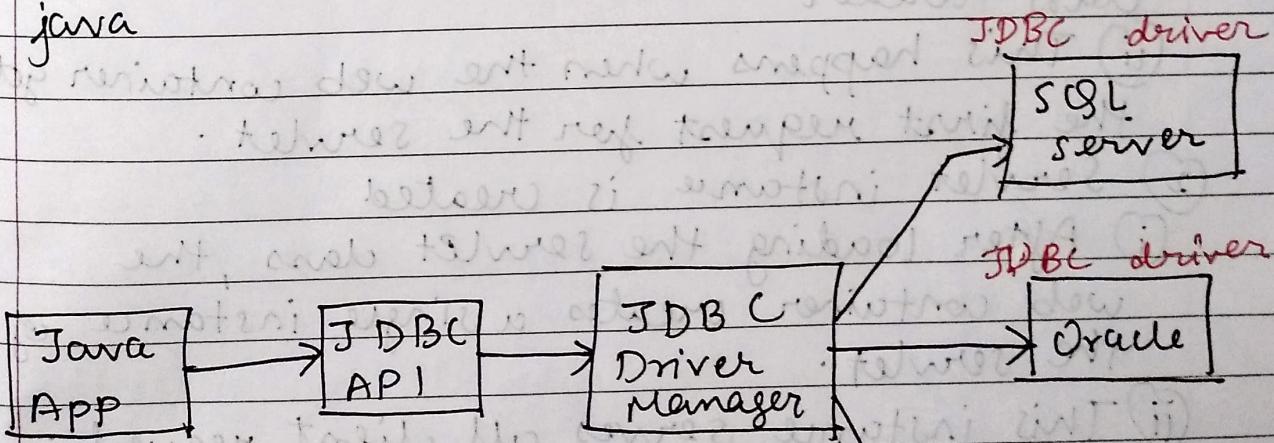
v) Process the results

vi) Close the connection

```
conn.close();
```

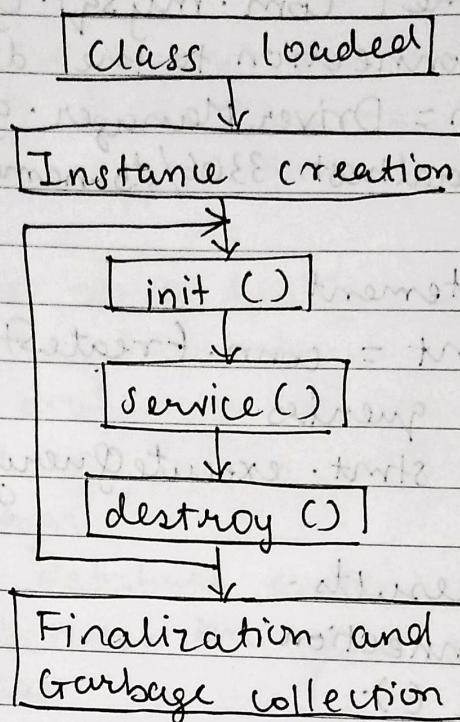
⑤ It supports multiple databases like MySQL, Oracle, PostgreSQL, etc.

⑥ It also works on any platform that supports Java



(Q2) Servlet life cycle

->



- ① Servlet class is loaded
 - i) The servlet class is loaded by the class loader.
 - ii) This happens when the web container gets the first request for the servlet.
- ② Servlet instance is created
 - i) After loading the servlet class, the web container creates a single instance of the servlet.
 - ii) This instance serves all client requests.
- ③ init() method is invoked.
 - i) The init() method is called once, immediately after the servlet instance is created.
 - ii) It initializes the servlet and prepares it for handling requests such as reading config files.

- (4) service() method is invoked
- (i) The service() method is called for every client request.
 - (ii) It determines the type of the request (GET, POST, etc) and routes it to the appropriate method (doGet(), doPost(), etc)
 - (iii) doGet(): Used to handle HTTP GET requests, typically for fetching data.
 - (iv) doPost(): Used to handle HTTP POST request, typically for submitting data.
- (5) destroy() method is invoked
- (i) Before removing the servlet instance, the web container calls the destroy() method.
 - (ii) It allows the servlet to release the resources held by the servlet with browser.

Q3) What is session tracking? Show how session tracking is achieved using cookies

OR

What are cookies and how do they work in Servlets?

- => ① Session simply means a particular interval of time.
- ② Session tracking is a way to maintain state of an user.
- ③ Session tracking using cookie
- (i) A cookie is a small piece of data stored on the client's browser.
 - (ii) Cookies are sent by the server to the client and then returned by the client in the subsequent requests.
 - (iii) Cookies are used to uniquely identify a user and maintain session data.
 - (iv) Working
 - (a) The server creates a cookie and sends it to the client's browser.
 - (b) The client's browser stores the cookie and sends it back to the server with every request.
 - (c) The server uses the cookie to recognize the client and maintain the session.

(i) Syntax of cookies:

(a) Creating a cookie:

Syntax: `Cookie cookie = new Cookie("name", "value")`

To create a cookie, we use the `Cookie` class constructor with a name and a value.

(b) Adding the cookie to the response :

Once a cookie is created , it is sent to the client using addCookie() method .

Syntax : response . addCookie (cookie) ;

(c) Retrieving cookies from a request :

To get the cookies sent by the browser , use the getCookies() method

Syntax : Cookie [] cookies = request . getCookies () ;

(d) Reading cookie data :

After retrieving the cookies , you can access their values using the getName() and getValue() methods

Syntax : String name = cookie . getName () ;

String value = cookie . getValue () ;

(e) Setting cookie expiry :

To specify how long the cookie should exist , use the setMaxAge() method .

Syntax : cookie . setMaxAge (3600) ;

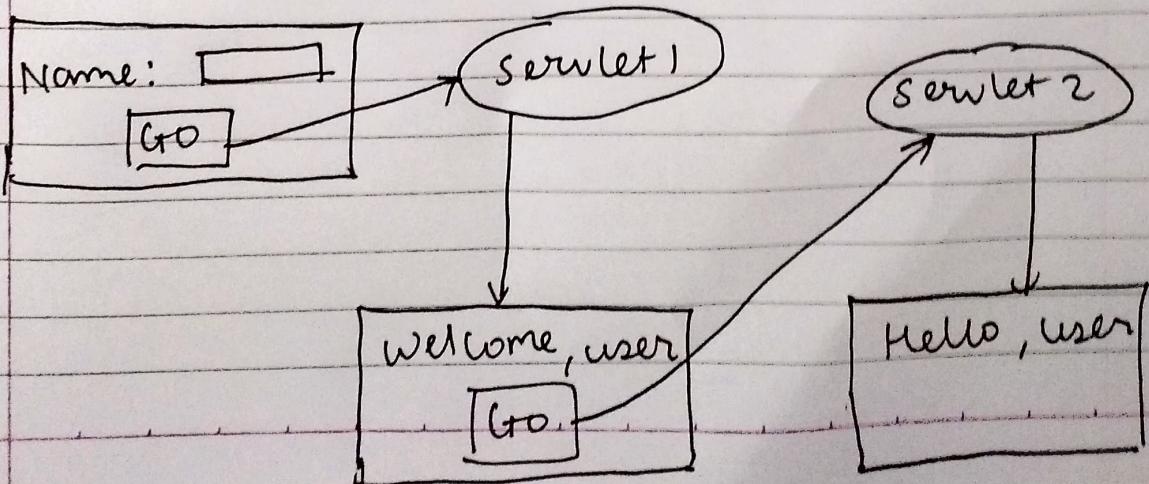
// Expiry in 1 hour (3600 seconds)

(f) Deleting a cookie

To delete a cookie , set its max age to 0 and send it to the client again .

Syntax : cookie . setMaxAge (0) ;

response . addCookie (cookie) ;



Q4. Write a short note on JSP

=>

- a. **Jakarta Server Pages (JSP)** also known as Java Server Pages is a technology used for developing dynamic web pages based on Java.
- b. It allows embedding Java code directly into HTML pages.
- c. JSP is part of the Java EE (Enterprise Edition) standard and provides an easy way to build web applications that can interact with databases and generate dynamic content.
- d. **Key Features of JSP**

1. Separation of Logic and Presentation:

- a. JSP separates the business logic from the presentation layer by using Java code and HTML tags in different sections.
- b. This makes the code easier to maintain and update.

2. Embedded Java Code:

- a. Java code can be embedded within HTML using special tags like <% code %>, <%= expression %>, and <%! declaration %>.
- b. This allows for dynamic content generation based on user input or other variables.

3. Automatic Compilation:

- a. When a JSP page is requested for the first time, the web container compiles it into a servlet, which is then executed.
- b. This compilation is automatic, making it easy for developers to focus on coding rather than configuration.

4. Tag Libraries:

- a. JSP supports **Tag Libraries** (such as JSTL - JavaServer Pages Standard Tag Library) that allow you to use pre-built tags for common operations like loops, conditionals, and database access, without writing Java code directly.

5. Session Management:

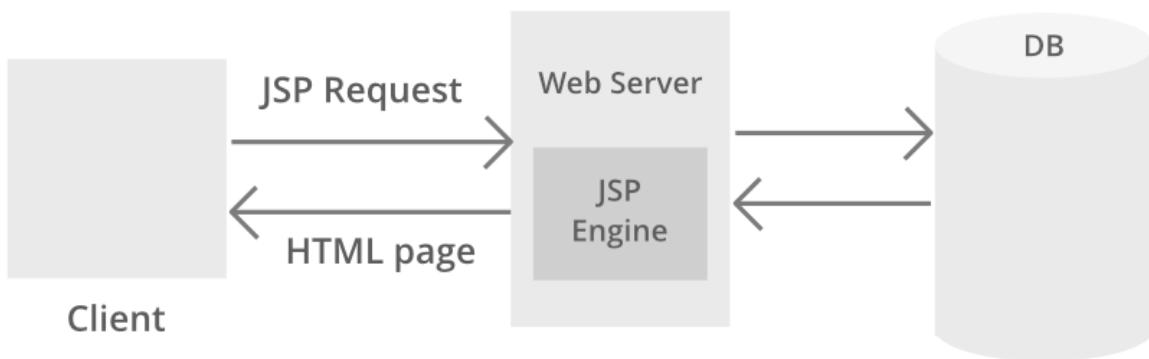
- a. JSP can manage sessions and cookies to track user interactions and maintain session data across multiple requests.

6. Integration with Java Beans:

- a. JSP works well with JavaBeans, which are reusable components in Java. JavaBeans encapsulate data and business logic, making it easier to build scalable applications.

e. JSP Life Cycle

1. **Translation Phase:** When a JSP page is requested for the first time, the container translates it into a servlet.
2. **Compilation Phase:** The servlet is compiled into a Java class.
3. **Initialization Phase:** The servlet is initialized by calling the init() method.
4. **Request Handling Phase:** The servlet processes client requests using the service() method, which in turn calls the appropriate JSP page.
5. **Destruction Phase:** The servlet is destroyed when the server is shut down or when the page is removed from the container's memory.



f. JSP Syntax

- **Directives:** Used to provide information about the JSP page (e.g., page language, imports).
`<%@ page language="java" contentType="text/html" %>`
- **Scriptlets:** Embed Java code inside HTML.
`<% int a = 10; %>`
- **Expressions:** Output values to the client.
`<%= "Hello, World!" %>`
- **Declarations:** Declare variables or methods.
`<%! int count = 0; %>`
- **Actions:** Used for working with components like JavaBeans or custom tags.
`<jsp:useBean id="bean" class="com.example.Bean" />`