# Experiment - 05.

- **Aim :** Implement the sample program demonstrating the use of servlet.

eg: Create a database table cbookshop (book-id, book-title, book-author, book-price, quantity) using database like Oracle / Mysql etc. and display (use sql select query) the table content using servelet.

- **Theory :** SERVELET.

I. Introduction to servelets.

- Servelets are java programs that run on a web server and handle HTTP requests and responses. They are an essential part of Java EE (Enterprise edition) and are used to build dynamic web applications.

- Servelets help in interacting with databases, processing user requests and dynamically generating web content.

II. Servelet Architecture -

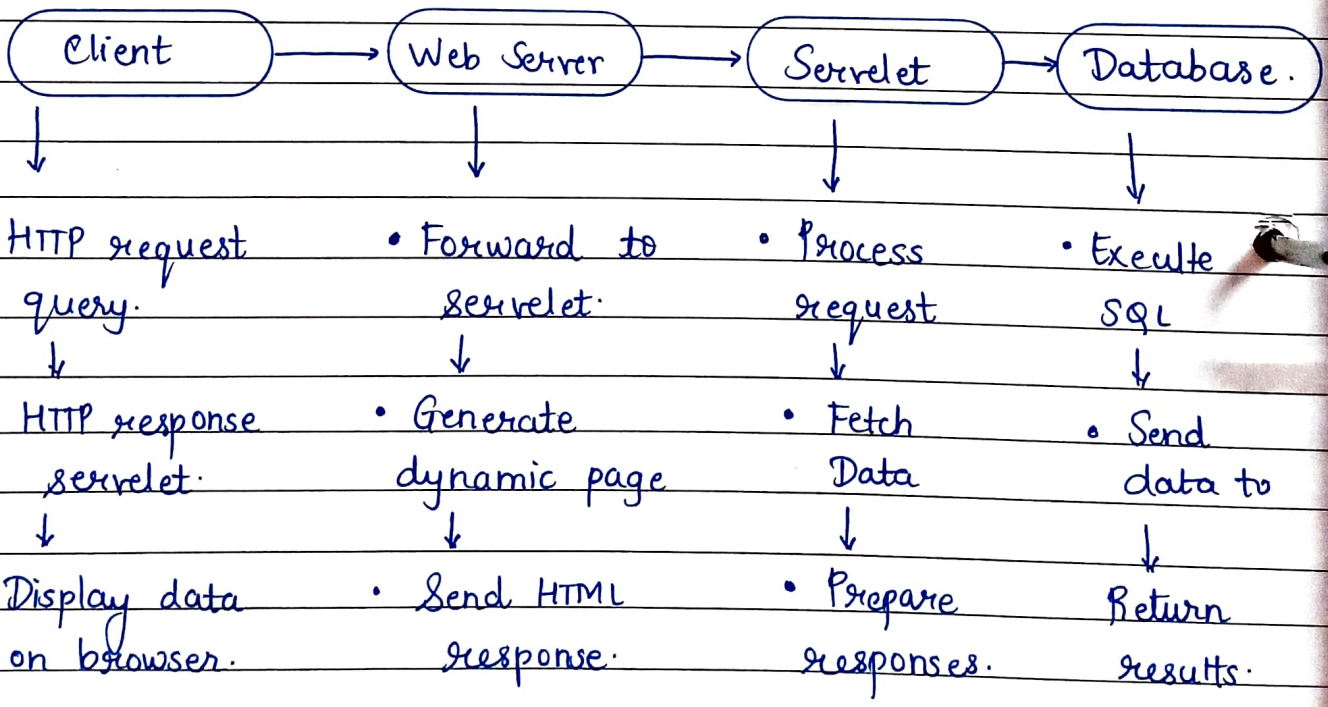
- A servelet operates within a Java EE server, handling client requests (usually from a web browser) and responding with dynamically generated content.

Page No. :
Date. : / /

- **Basic Servelet Flow -**

1. Client sends an HTTP request (eg. clicking a button on a web page)
2. Web server receives the request and passes it to the appropriate Servelet.
3. Servelet processes the request, interacts with a database (if required) and prepares response.
4. Web server sends the response back to the client.

- **Block diagram of servelet processing -**



III] Steps to implement Servelet for displaying database content.

STEP 1 : Set up the database (MySQL/ Oracle).

1. Choose database System.
   - Install mysql or Oracle and configure it.
   - Start the database server and ensure it is running.

2. Create database and table: ebookshop.
   eg - book-id (Primary-key, INT)
   book-title (VARCHAR)
   book-author (VARCHAR)
   book-price (FLOAT)
   book-quantity (INT)

3. Insert Sample data.

STEP 2 : Configure the java EE Environment.

1. Install JDK (Java development kit).
   - Download and install JDK 8 or later.
   - Set up the JAVA-HOME environment variable.

2. Install Tomcat Server.
   - Download and configure Apache Tomcat (or any Java EE Server)
   - Deploy applications inside the webapps folder.

26.

Page No. :
Date. : / /

## STEP 3 : Create a Java Servlet.

- Connect to database using JDBC.
- Execute and SQL select query to retrieve the book details.
- Display the results in an HTML table.

1. **Import required packages.**
   - java.io.* for handling I/O operations.
   - javax.servlet and http. for servlet.
   - java.sql.* for database operations.

2. **Extend HttpServelet Class.**
   Override doGet() method to process the HTTP GET request.

3. **Load JDBC driver** – Use Class.forName() to register the MySQL / Oracle driver.

4. **Estabilish Database connection** –
   - Use DriverManager.getConnection() with the correct database URL, username and password.

5. **Execute SQL Query** –
   Use a statement or prepared Statement. to retriere book records.

6. **Generate HTML outputs** –
   - Dynamically generate an HTML page with the book details in <HTML> table.

## STEP 4 : Configure web.xml deployment descriptor.

Every servelet application requires deployment configuration.

1. Define the servelet in web.xml.
   - Map the servelet to a specific URL pattern
2. Place web.xml inside WEB-INF folder.

## STEP 5: Deploy and run the servelet:

1. Start tomcat server
   - Run startup.bat (Windows) or startup.sh (linux)
2. Deploy the servelet application
   - Copy the WAR file or project folder to webapps.
3. Access the servelet via web browser -
   Open a browser and navigate to.

   http://localhost:8080/YourProjectName/displayBooks.

---

## IV] Handling errors and security Best practices -

- While implementing servelets, it is essential to handle errors properly and follow security guidelines.

Error Handling in servelets -

1. SQL Exceptions - Use try-catch blocks when executing queries.

Page No. :
Date. : / /

2. Servelet Exceptions - Override doPost() or doGet() methods correctly.

3. NullPointerException - Check if request parameters are null before processing.

* Security Best Practices -

- Use prepared statements instead of Statement to prevent SQL injection.
- Use HTTPs instead of HTTP for secure data transmission.
- Use data validation to prevent malicious inputs.

- **Conclusion:** Thus we successfully, implemented the java servelet application for books.

* * * * * * * *.