

# Assignment - 2

## ① Configure Database:

```

<Resource name = "jdbc/MyDB"
    auth = "Container"
    type = "javax.sql.DataSource"
    maxIdle = "10"
    maxWaitMillis = "10000"
    username = "dbuser"
    password = "db password"
    driverClassName = "com.mysql.cj.jdbc.Driver"
    url = "jdbc:mysql://localhost:3306/mydatabase"/>

```

look up Data Source in Java code using JNDI

```

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;

```

## Executing Set Queries Using JDBC Statement

### ① Using a Statement:

```

try (Connection conn = DatabaseUtil.getConnection();
Statement stmt = conn.createStatement()) {
    String query = "SELECT * from Users";
    ResultSet rs = stmt.executeQuery(query);
    while (rs.next()) {

```

```

        System.out.print("User ID: " + rs.getInt("id") +
                        " Name: " + rs.getString("name"));
    }
}

```

### ② Using a prepared Statement:

```
try (Connection conn = DatabaseUtil.getConnection()) {
    PreparedStatement psmt = conn.prepareStatement("SELECT * FROM users WHERE id=?");
    psmt.setInt(1, 1);
    ResultSet rs = psmt.executeQuery();
    while (rs.next()) {
        System.out.println("User ID: " + rs.getInt("id"));
        System.out.println("Name: " + rs.getString("name"));
    }
}
```

### ③ Using a Callable Statement for Stored procedures:

```
try (Connection conn = DatabaseUtil.getConnection()) {
    CallableStatement cstmt = conn.prepareCall("?
        CALL get_user_By_ID(?);
        ?");
    cstmt.setInt(1, 1);
    ResultSet rs = cstmt.executeQuery();
    while (rs.next()) {
        System.out.println("User ID: " + rs.getInt("id"));
        System.out.println("Name: " + rs.getString("name"));
    }
}
```

### Output:

Statement Example  
User ID : 1 , Name: Meghana Output:  
User ID : 2 ; Name: Abhi

### Prepared Statement :

User ID : 1 , Name: Meghana

### Callable Statement :

User ID : 1 , Name : Meghana

## ② Life Cycle Phases of a Jsp Pages:

1. Translation phase
2. Compilation phase
3. Initialization phase
4. Request Processing phase
5. Destruction phase

## Embedding Java Code in Jsp:

### ① Scriptlets:

```
<% int sum = 5+10; %>
```

```
<p> Sum : <%= sum %>
```

Output: Sum : 15

### ② Declarations:

```
<%! int add (int a,int b) { return a+b; } %>
```

```
<p> Result is : <%= add (3,7) %> </p>
```

Output: Result is #:10

### ③ Expressions:

```
<p> current time : <%= new java.util.Date () %>
```

Output: Current Time: Mon Sep 09 09:08:00

PPT 2024

### ④ PHP Code:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title> Chen Bead </title>
```

```
    <Style>
```

```
        table {
```

```
            width:400px;
```

```
            border-collapse: collapse;
```

```
        }
```

```
    td {
```

```
width: 30px;  
height: 30px;  
}  
</Style>  
</head>  
<body>  
<table>  
<!php>  
for ($row=0; $row<8; $row++) {  
    echo "<tr>";  
    for ($col=0; $col<8; $col++) {  
        if ((($row+$col) % 2 == 0)) {  
            echo "L  |        } else {  
            echo "<td style = \"bg-color: black;></td>";  
        }  
        echo "</tr>";  
    }  
</table>  
</body>  
</html>
```

### Output:

```
[ ] [ # ] [ ] [ # ] [ ] [ # ]  
[ # ] [ ] [ # ] [ ] [ # ] [ ]  
[ ] [ # ] [ ] [ # ] [ ] [ # ]  
[ # ] [ ] [ # ] [ ] [ # ] [ ]  
[ ] [ # ] [ ] [ # ] [ ] [ # ]
```

④ PHP Application to extract Data and Store  
in XML:

Steps:

1. Read content from a text file
2. Extract patterned using Regular
3. Create and store results in an XML file
4. Define XML Schema

PHP Code:

```
<?php  
$filename = 'input.txt';  
$Content = file_get_contents($filename);  
preg_match_all($content, $Emails);  
$XML = new SimpleXMLElement('<data>  
                                </data>')  
  
{  
    $XML->asXML('output.xml');  
    echo "Data extracted and saved to Output XML"  
!>
```

Output:

```
<data>  
  <Email>  
    <Email> example@example.com </Email>  
    <Email> example@example.com </Email>  
  </Emails>  
  <Phones>  
    <Phone> 123-456-7890 </Phone>  
  </Phones>  
</data>
```