Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT 4 QUESTIONS

S.No	Questions	Marks	CO	BTL
1	From a developer's perspective, discuss why JDBC is essential in building database-driven applications. How to achieve JDBC CONNECTION pooling using JDBC Data Source and JNDI in Apache Tomcat Server. Provide examples of executing SQL queries using JDBC statements. Discuss the differences between Statement, Prepared Statement, and Callable Statement.	20	CO5	3
2	Describe the lifecycle phases of a JSP page. Explain the significance of each phase in the JSP execution process. Discuss the different ways to embed Java code within a JSP page with examples. Explain the advantages and disadvantages of using scriptlets, declarations, and expressions in JSP.	20	CO4	2
3	You need to develop a PHP program that generates a chessboard using HTML tables. The table should have a total width of 400px, and each cell should have a height and width of 30px. The chessboard should alternate colors between black and white for each cell to represent a typical chessboard layout. How would you write a PHP program using nested for loops to create a chessboard? The chessboard should be displayed using an HTML table with a total width of 400px, and each cell should have a height and width of 30px. Explain how you would use the nested for loops to alternate the cell colors and ensure the chessboard pattern is correctly displayed. Provide the code for this program	20	CO4	2
	You are developing a PHP application that reads content from a text file and uses regular expressions to extract specific patterns, such as email addresses and phone numbers. After extracting the data, the application should store the results in a new XML file following a defined schema for the data. Additionally, you need to compare and contrast DTD (Document Type Definition) and XML Schema for defining the XML structure. How would you create a PHP application that reads content from a text file and uses regular expressions to extract specific patterns (e.g.,	20	CO5	2

SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

	email addresses and phone numbers)? After extracting the data, store the results in a new XML file following a defined schema for the data. Additionally, compare and contrast DTD and XML Schema for defining XML structure. Describe the steps and provide the code for the application.			

Assignment-4

Questions	Mark Split Up	Marks	Total Marks
1.	Explanation of JDBC	(5)	20
	Connection pooling	(6)	
	SQL Queries	(5)	
	Statement Types	(4 M)	
2.	Life cycle Phase Explanation	(6)	20
	Embedding Java Code	(5)	
	Advantages and Disadvantages	(5)	
	Clarity and Depth	(4)	
3.	Code Implementation	(8)	20
	HTML Table Structure	(5)	
	Alternating Color Logic	(4)	
	Explanation	(3)	
4.	Code Implementation	(8)	20
	Pattern Extraction	(5)	
	XML file Generation	(4)	
	DTD & XML Schema	(3)	

ASSIGNMENT-4

(1) Why JDBC is essential in building database-driven applications.

JDBC is essential because it provides a standardized API for java applications to interact with databases. Achieving JDBC connection pooling using JDBC data source and JNDI in Apache Tomcat

Configure Database:

```
<Resource name = "jdbc/MYDB"
          auth = "container"
          type = "javax.sql.DataSource"
          maxTotal = "20"
          maxIdle = "10"
          maxWaitMills = "10000"
          username = "dbuser"
          password = "db password"
          driverClassName = "com.mysql.cj.jdbc.
                             Driver"
          url = "jdbc:mysql://localhost:3306/
                 mydatabase" />
```

lookup DataSource in Java Code Using JNDI

```
import javax.naming.context;
import javax.naming.Inital Context;
import javax.sql.DataSource;
import java.sql.connection;
```

Executing SQL 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.println("User ID: " + rs.getInt
            ("id") + Name: " + rs.getString("
    }
}
```

2. Using a Prepared Statement :

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

3. Using
try (Co
call

}

Ou

St

(2)

3. Using a Callable Statement for stored Procedures :-

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

Output :-

Statement Example Output :-

User ID : 1, Name : Vishnu

User ID : 2, Name : Harshu

Prepared Statement :

User ID : 1, Name : Vishnu

Callable Statement

User ID : 1, Name : Vishnu.

(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. Scriplets :-

```
<% 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() %> </p>
```

Output :-

Current time : Mon Sep 09 09:03:00 PDT 2024

(3) Generates a chessboard using HTML tables, width of 400px (total) and each cell is at a height and width of 30px.

PHP Code :-

```
<!DOCTYPE html>
<html>
<head>
    <title> ChessBoard </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 "<td style = "bg-color: white></td>";
        } 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 patterns using Regular
3. Create and Store results in an XML file
4. Define XML Schema.

PHP Code:

```
<? php
$filename = 'input.txt';
$content = file_get_contents ($file name);
preg_match_all ($content, $emails);
$XML = new simple XML Element ('<data></data>');
```

```
$emailElement = $xml->addChild('emails');
foreach ($emails[0] as $email) {
    $emailElement->addChild('email', $email);
}

$xml->asXML('output.xml');
echo "Data extracted and saved to output.xml";
?>
```

Output :-

```
<data>
  <emails>
    <email>example@example.com</email>
    <email>example@example.com</email>
  </emails>
  <phones>
    <phone>+123-456-7890</phone>
    <phone>987-654-3210</phone>
  </phones>
</data>
```