CSI3023

Advanced Server-side Programming

Winter Semester 2023 – '24

Digital Assignment - 1

SARAGA S
20MIC0081

1.Develop a Servlet web application with the following using JDBC

(i) Create a student table with Reg No, Name, Age, DOB, Department, Branch. And insert any 5 records.

(ii) Retrieve the names of students who are enrolled in more than one department and display the count of departments for each student.

(iii) Identify students with the same date of birth and list their names along with their common date of birth.

(iv) Retrieve the names and ages of students in the 'Computer Science' department whose names start with the letters 'Ram'.

CODE:

```java
// Import necessary libraries
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class StudentServlet extends HttpServlet {
 public    void    doGet(HttpServletRequest    request,    HttpServletResponse
response) throws
ServletException, IOException {
 response.setContentType("text/html");
 PrintWriter out = response.getWriter();
 try {
 // Load the JDBC driver
 Class.forName("com.mysql.jdbc.Driver");
 // Establish a connection
 Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/your_database", "username",
"password");
 // Create a statement
 Statement stmt = conn.createStatement();
```

```java
   // Execute SQL queries and process the results
   // ...
   // Close the connection
   conn.close();
   } catch (Exception e) {
   out.println("Error: " + e.getMessage());
   }
   }
}
```

<u>TO CREATE TABLE:</u>

```java
stmt.executeUpdate("CREATE TABLE Student (RegNo INT, Name VARCHAR(255), Age INT, DOB

DATE, Department VARCHAR(255), Branch VARCHAR(255))");

stmt.executeUpdate("INSERT INTO Student VALUES (1, 'John', 20, '2000-01-01', 'CS', 'CS1')");

// Insert 4 more records...

ResultSet rs = stmt.executeQuery("SELECT Name, COUNT(DISTINCT Department) as DeptCount

FROM Student GROUP BY Name HAVING DeptCount > 1");

while (rs.next()) {
 out.println(rs.getString("Name") + ": " + rs.getInt("DeptCount"));
}

ResultSet rs = stmt.executeQuery("SELECT Name, DOB FROM Student WHERE DOB IN (SELECT

DOB FROM Student GROUP BY DOB HAVING COUNT(*) > 1)");

while (rs.next()) {
 out.println(rs.getString("Name") + ": " + rs.getDate("DOB"));
}

ResultSet rs = stmt.executeQuery("SELECT Name, Age FROM Student WHERE Department =

'Computer Science' AND Name LIKE 'Ram%'");

while (rs.next()) {
 out.println(rs.getString("Name") + ": " + rs.getInt("Age"));
}
```

OUTPUT:

DISPLAY THE DATA IN THE TABLE teacher before INSERT

TeacherID	Name	Age	Course	Experience
1	yuva	20	computer	2

DISPLAY THE DATA IN THE TABLE teacher

TeacherID	Name	Age	Course	Experience
1	yuva	20	computer	2
2	yuvashree	25	maths	3
3	raj	28	tamil	6
4	kumar	45	computer	4
5	shree	29	histiry	7
6	Preethi	25	biology	10
7	prabha	41	maths	15
8	renu	39	socialscience	9
9	Saran	45	drawing	19
10	Varshini	25	chemistry	4
11	magesh	45	english	15

SELECTING DATA USING WHERE CLAUSE

TeacherID	Name	Age	Course	Experience
2	yuvashree	25	maths	3

DISPLAY THE DATA IN THE TABLE teacher before INSERT

TeacherID	Name	Age	Course	Experience
1	yuva	20	computer	2
2	yuvashree	25	maths	3
3	raj	28	tamil	6
4	kumar	45	computer	4
5	shree	29	histiry	7
6	Preethi	25	biology	10
7	prabha	41	maths	15
8	renu	39	socialscience	9
9	Saran	45	drawing	19
10	Varshini	25	chemistry	4
11	magesh	45	english	15

DISPLAY THE DATA IN THE TABLE teacher before INSERT

| TeacherID	Name	Age	Course	Experience
1	yuva	20	computer	2
2	yuvashree	25	maths	3
3	raj	28	tamil	6
4	kumar	45	computer	4
6	Preethi	25	biology	10
7	prabha	41	maths	15
8	renu	39	socialscience	9
9	Saran	45	drawing	19
10	Varshini	25	chemistry	4
11	magesh	45	english	15

2. Develop a JSP Web application to accept the series of numbers from the user to find the given series is Fibonacci series or not.

CODE:

```
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<%
 String series = request.getParameter("series");
 List<Integer> numbers = new ArrayList<Integer>();
 for (String num : series.split(",")) {
 numbers.add(Integer.parseInt(num.trim()));
 }
 boolean isFibonacci = true;
 for (int i = 2; i < numbers.size(); i++) {
 if (numbers.get(i) != numbers.get(i-1) + numbers.get(i-2)) {
 isFibonacci = false;
 break;
 }
 }
%>
```

```
<html>

<body>

 <h1>The series <%= series %> is <%= isFibonacci ? "" : "not " %>a Fibonacci
series.</h1>

</body>

</html>
```

OUTPUT:
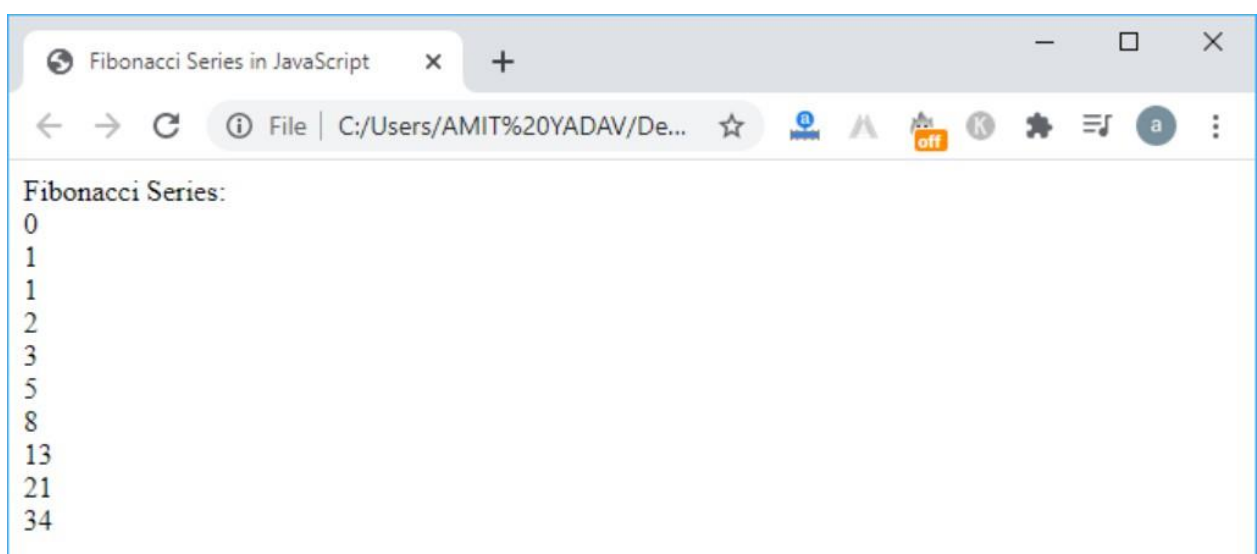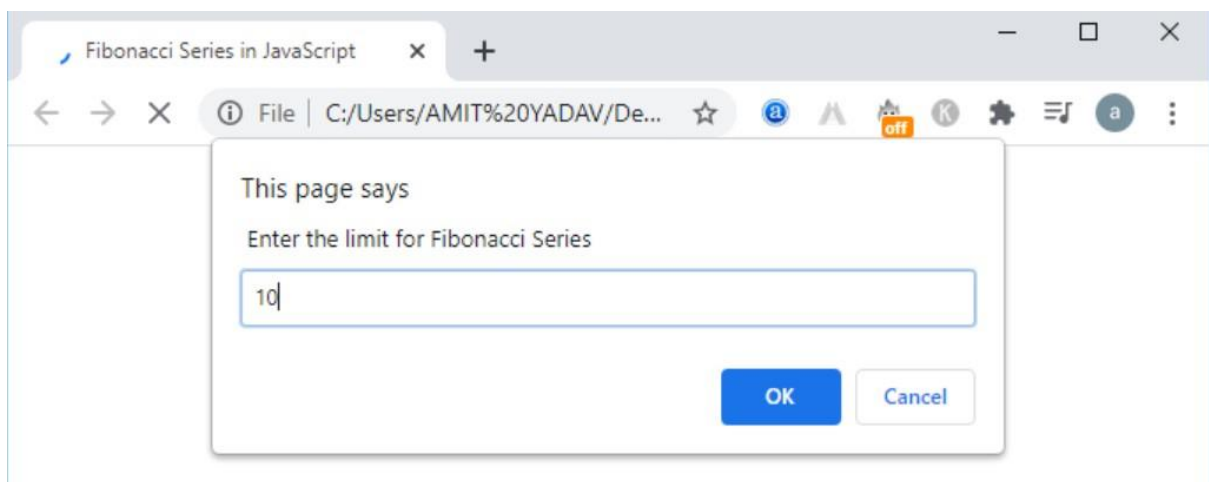
3.Develop a JSP Web application to accept an integer number from user to find the given integer number is Armstrong number or not. An Armstrong number is a number that is the sum of its own digits each raised to the power of the number of digits.

For a 3-digit number like 153:

1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153

CODE:

```
<%@ page import="java.io.*" %>
<%
    String number = request.getParameter("number");
    int num = Integer.parseInt(number);
    int originalNum = num;
    int remainder, result = 0, n = 0;

    for (;originalNum != 0; originalNum /= 10, ++n);
    originalNum = num;

    for (;originalNum != 0; originalNum /= 10) {
        remainder = originalNum % 10;
        result += Math.pow(remainder, n);
    }

    if(result == num)
        out.println(num + " is an Armstrong number.");
    else
        out.println(num + " is not an Armstrong number.");
%>
```

OUTPUT:


```
PS C:\Users\viola\Downloads\ADV SEV LAB\ArmstrongNumber> cd "c:\Users\viola\Downloads\ADV SEV LAB\ArmstrongNumber\src\" ; if ($?) { javac Armstron
gNumber.java } ; if ($?) { java ArmstrongNumber }
Enter a number to check if it's an Armstrong number: 5324
5324 is not an Armstrong number.
PS C:\Users\viola\Downloads\ADV SEV LAB\ArmstrongNumber\src> cd "c:\Users\viola\Downloads\ADV SEV LAB\ArmstrongNumber\src\" ; if ($?) { javac Arms
trongNumber.java } ; if ($?) { java ArmstrongNumber }
Enter a number to check if it's an Armstrong number: 407
407 is an Armstrong number.
PS C:\Users\viola\Downloads\ADV SEV LAB\ArmstrongNumber\src>
```