

Index

Sr. No.	Title	Date	Signature
1	Program on JDBC using statements.		
2	Program to demonstrate Generic and HTTP Servlet response and request objects.		
3	Program to demonstrate Http session using cookies, URL rewritten, Hidden Fields.		
4	Program to demonstrate online web applications (Online Quiz)		
5	Program to demonstrate Page directives with JSP		
6	Program to demonstrate Http session using cookies		
7	Program to demonstrate Error Page in JSP.		
8	Program to demonstrate JSP Actions, JSP with beans		
9	Program to demonstrate custom tags, JSTL tags		
10	Program to demonstrate simple XML tags		
11	Program to demonstrate DTD, XML Parser, Validator.		
12	Program to Write a Hibernate Application, Configuring Hibernate for our application.		
13	Program to fetch data from the database using Hibernate and Annotations.		
14	Building first Spring Application		

15	Simple web application using Spring		
----	-------------------------------------	--	--

Practical No :1

Objective: Program on JDBC using statements.

Program:

pom.xml:

```
<dependencies>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.33</version>
</dependency>
<dependency>
  <groupId>jakarta.servlet</groupId>
  <artifactId>jakarta.servlet-api</artifactId>
  <version>6.1.0</version>
</dependency>
</dependencies>
```

Servlet.java:

```
package org.example.servlets;
import java.io.*;
import java.sql.*;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
@WebServlet("/StudentServlet")
public class Servlet extends HttpServlet {

    Connection con;
    Statement stmt;
    final String DB_NAME = "studentdatabase";

    public void init() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection rootCon =
DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "");
```

```

Statement rootStmt = rootCon.createStatement();
rootStmt.executeUpdate("CREATE DATABASE IF NOT EXISTS " + DB_NAME);
rootCon.close();

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/" +
DB_NAME, "root", "");
stmt = con.createStatement();
stmt.executeUpdate("CREATE TABLE IF NOT EXISTS studentdetails ("
    + "ID INT PRIMARY KEY AUTO_INCREMENT,"
    + "Name VARCHAR(100),"
    + "Email VARCHAR(100),"
    + "City VARCHAR(100),"
    + "Java INT,"
    + "Python INT,"
    + "Cprogram INT)");
} catch (Exception e) {
    e.printStackTrace();
}
}

public void doPost(HttpServletRequest req, HttpServletResponse res)
    throws IOException, ServletException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    String action = req.getParameter("action");

    try {
        if (action.equals("insert")) {
            String name = req.getParameter("name");
            String email = req.getParameter("email");
            String city = req.getParameter("city");
            int java = Integer.parseInt(req.getParameter("java"));
            int python = Integer.parseInt(req.getParameter("python"));
            int c = Integer.parseInt(req.getParameter("c"));

            String q = "INSERT INTO studentdetails (Name, Email, City, Java,
Python, Cprogram) "
                + "VALUES ('" + name + "','" + email + "','" + city +
"'," + java + "," + python + "," + c + ")";
            stmt.executeUpdate(q);
            out.println("<h3> Student inserted!</h3>");

        } else if (action.equals("delete")) {
            int id = Integer.parseInt(req.getParameter("id"));
            int rows = stmt.executeUpdate("DELETE FROM studentdetails WHERE
ID = " + id);
            out.println(rows > 0 ? "<h3> Deleted!</h3>" : "<h3> ID not
found.</h3>");
        }
    }
}

```

```

    } else if (action.equals("update")) {
        int id = Integer.parseInt(req.getParameter("id"));
        String email = req.getParameter("email");
        String city = req.getParameter("city");
        int rows = stmt.executeUpdate("UPDATE studentdetails SET
Email='" + email + "', City='" + city + "' WHERE ID=" + id);
        out.println(rows > 0 ? "<h3> Updated!</h3>" : "<h3> ID not
found.</h3>");

        } else if (action.equals("show")) {
            ResultSet rs = stmt.executeQuery("SELECT * FROM
studentdetails");
            out.println("<h2>All Students:</h2><table
border='1'><tr><th>ID</th><th>Name</th><th>Email</th><th>City</th><th>Java</th>
<th>Python</th><th>C</th></tr>");
            while (rs.next()) {
                out.println("<tr><td>" + rs.getInt("ID") + "</td><td>" +
                    rs.getString("Name") + "</td><td>" +
                    rs.getString("Email") + "</td><td>" +
                    rs.getString("City") + "</td><td>" +
                    rs.getInt("Java") + "</td><td>" +
                    rs.getInt("Python") + "</td><td>" +
                    rs.getInt("Cprogram") + "</td></tr>");
            }
            out.println("</table>");
        } else {
            out.println("<h3> Unknown action.</h3>");
        }

        out.println("<br><a href='index.html'>Back to Home</a>");

    } catch (Exception e) {
        out.println("<h3>Error: " + e.getMessage() + "</h3>");
    }
}

public void destroy() {
    try {
        stmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

index.html :

```

<!DOCTYPE html>
<html>

```

```

<head>
    <title>Student Operations</title>
</head>
<body>
<h1>Student Management</h1>

<form method="post" action="StudentServlet">
    <input type="hidden" name="action" value="insert" />
    <fieldset style="display: flex;align-items: center;justify-content:
center;gap: 10px;flex-wrap: wrap;width: 50%;">
        <legend>Insert Student</legend>
        Name: <input type="text" name="name" /><br>
        Email: <input type="text" name="email" /><br>
        City: <input type="text" name="city" /><br>
        Java: <input type="number" name="java" /><br>
        Python: <input type="number" name="python" /><br>
        C: <input type="number" name="c" /><br>
        <input type="submit" value="Insert" />
    </fieldset>
</form>

<form method="post" action="StudentServlet">
    <input type="hidden" name="action" value="delete" />
    <fieldset style="display: flex;align-items: center;justify-content:
center;gap: 10px;flex-wrap: wrap;width: 50%;">

        <legend>Delete Student</legend>
        ID: <input type="number" name="id" /><br>
        <input type="submit" value="Delete" />
    </fieldset>
</form>

<form method="post" action="StudentServlet">
    <input type="hidden" name="action" value="update" />
    <fieldset style="display: flex;align-items: center;justify-content:
center;gap: 10px;flex-wrap: wrap;width: 50%;">
        <legend>Update Student</legend>
        ID: <input type="number" name="id" /><br>
        New Email: <input type="text" name="email" /><br>
        New City: <input type="text" name="city" /><br>
        <input type="submit" value="Update" />
    </fieldset>
</form>

<form method="post" action="StudentServlet">
    <fieldset style="display: flex;align-items: center;justify-content:
center;gap: 10px;flex-wrap: wrap;width: 50%;">
        <input type="hidden" name="action" value="show" />
        <legend>Show All Students</legend>

```

```

        <input type="submit" value="Show" />
    </fieldset>
</form>
</body>
</html>

```

Output:

Student Management

Insert Student

Name: BENJAMIN SEBASTIAN
Email: benjaminsebastian156@gr
City:
Kochi
Java: 98
Python:
90
C: 99
Insert

Delete Student

ID:
Delete

Update Student

ID:
New Email:
New City:
Update

Show All Students

Show

All Students:

ID	Name	Email	City	Java	Python	C
3	BENJAMIN SEBASTIAN	benjaminsebastian156@gmail.com	Kochi	98	90	99

[Back to Home](#)

Practical No :2

Objective: Program to demonstrate Generic and HTTP Servlet response and request objects.

Program:

pom.xml :

```
<dependencies>
    <dependency>
        <groupId>jakarta.servlet</groupId>
        <artifactId>jakarta.servlet-api</artifactId>
        <version>6.1.0</version>
    </dependency>
</dependencies>
```

ServletGeneric.java :

```
package org.example;

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.annotation.WebServlet;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/genericServlet")
public class ServletGeneric extends GenericServlet {
    @Override
    public void service(ServletRequest req, ServletResponse resp) throws
ServletException, IOException {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Generic Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>This is a Generic Servlet</h1>");
    }
}
```

```

        out.println("Request prameter value
:"+req.getParameter("param").toString());
        out.println("<br><br>");
        out.println("<a href='httpervlet?param=100'>Go to Http servlet</a>");
        out.println("</body>");
        out.println("</html>");
    }
}

```

ServletHttp.java :

```

package org.example;

import jakarta.servlet.GenericServlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.ServletRequest;
import jakarta.servlet.ServletResponse;
import jakarta.servlet.annotation.WebServlet;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/genericservlet")
public class ServletGeneric extends GenericServlet {
    @Override
    public void service(ServletRequest req, ServletResponse resp) throws
ServletException, IOException {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Generic Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>This is a Generic Servet</h1>");
        out.println("Request prameter value
:"+req.getParameter("param").toString());
        out.println("<br><br>");
        out.println("<a href='httpervlet?param=100'>Go to Http servlet</a>");
        out.println("</body>");
        out.println("</html>");
    }
}

```

Output:

This is a Generic Servlet

Request parameter value :200

[Go to Http servlet](#)

This is a Http Servlet

Request parameter value :100

[Go to Generic servlet](#)

Practical No :3

Objective: Program to demonstrate Http session using cookies, URL rewritten, Hidden Fields.

Program:

pom.xml:

```
<dependencies>
  <dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <version>6.1.0</version>
  </dependency>
</dependencies>
```

ServletHttp.java :

```
package org.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/index.html")
public class ServletHttp extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie cookie=new Cookie("username","Benjamin");
        response.addCookie(cookie);
        cookie.setMaxAge(3600);

        HttpSession session=request.getSession();
```

```

        session.setAttribute("age", 21);
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Http Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Http Servlet - Session Management</h1>");
        out.println("<form action='servletData' method='GET'>");
        out.println("<input type='hidden' value='KJSIM' name='college'>");
        out.println("<input type='submit' value='See Data'>");
        out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }
}

```

ServletData.java :

```

package org.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/servletData")
public class ServletData extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        String course=req.getParameter("course")!=null ?
        req.getParameter("course"):"<a
href=\"servletData?course=MCA&college=KJSIM\">Click here to load course</a>";

        Cookie[] cookies = req.getCookies();
        Cookie cookie = (cookies != null && cookies.length > 0) ? cookies[0] :
        null;

        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>ServletData</title>");
        out.println("</head>");
    }
}

```

```

        out.println("<body>");
        out.println("<h1>Servlet Data - Session Management</h1>");
        out.println("<fieldset><legend>Cookies</legend>");
        out.println("<h3>Username : "+cookie.getValue()+"</h3>");
        out.println("</fieldset>");
        out.println("<fieldset><legend>Session</legend>");
        out.println("<h3>Age : "+req.getSession().getAttribute("age")+"</h3>");
        out.println("</fieldset>");
        out.println("<fieldset><legend>Url Re-writing</legend>");
        out.println("<h3>Course : "+course+"</h3>");
        out.println("</fieldset>");
        out.println("<fieldset><legend>Hidden Form</legend>");
        out.println("<h3>College : "+req.getParameter("college")+"</h3>");
        out.println("</fieldset>");
        out.println("</body>");
    }
}

```

Output :

Http Servlet - Session Management

See Data

Servlet Data - Session Management

Cookies
Username : Benjamin
Session
Age : 21
Url Re-writing
Course : Click here to load course
Hidden Form
College : KJSIM

Servlet Data - Session Management

Cookies
Username : Benjamin
Session
Age : 21
Url Re-writing
Course : MCA
Hidden Form
College : KJSIM

Practical No :4

Objective: Program to demonstrate online web applications (Online Quiz)

Program :

pom.xml :

```
<dependencies>
  <dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <version>6.1.0</version>
  </dependency>
</dependencies>
```

index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Play Quiz</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 flex items-center justify-center h-screen">

<div class="bg-white p-8 rounded-lg shadow-md w-96 border-solid border-sky-500
border-1">
  <h2 class="text-2xl font-bold mb-4 text-center">Play Quiz</h2>

  <form action="PlayQuiz" method="GET" class="space-y-4"> <div>
```

```

        <label for="rollno" class="block text-gray-700 font-medium">Roll
No:</label>
        <input type="number" id="rollno" name="rollno" class="w-full border
border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2
focus:ring-blue-500" required>
    </div>
        <input type="hidden" name="q" value="1">
    <div>
        <label for="name" class="block text-gray-700
font-medium">Name:</label>
        <input type="text" id="name" name="name" class="w-full border
border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2
focus:ring-blue-500" required>
    </div>

    <div class="text-center"> <button type="submit" class="bg-blue-500
hover:bg-blue-700 text-white font-bold py-2 px-4 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-300">
        Play Quiz
    </button>
    </div>
</form>
</div>

</body>
</html>

```

PlayQuizServlet.java :

```

package mygame;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.WebServlet;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/PlayQuiz")
public class PlayQuizServlet extends HttpServlet {
    final String[] aptitudeQuestions = {
        "\"What is the square root of 144?\"",
        "\"If a box contains 15 apples and you take away 5, how many do you have?\"",
        "\"A shopkeeper sells an item for $80, making a 25% profit. What was the cost price?\"",
        "\"What is the missing number in the sequence: 3, 6, 9, 12, ?\"",
        "\"A train covers 360 km in 6 hours. What is its speed?\""
    };
}

```

```

        "\"A bag contains 4 red balls, 5 green balls, and 6 blue balls. What
is the probability of picking a red ball?\", \"1/2\", \"2/5\", \"3/15\", \"4/15\",
        \"If the sum of two numbers is 90 and one number is 35, what is the
other number?\", \"45\", \"35\", \"60\", \"55\",
        \"What is 20% of 250?\", \"40\", \"45\", \"60\", \"50\",
        \"A shopkeeper gives a discount of 20% on a $500 product. What is
the final price?\", \"$350\", \"$450\", \"$300\", \"$400\",
        \"If a car travels 90 km in 1.5 hours, what is its average
speed?\", \"50 km/h\", \"55 km/h\", \"40 km/h\", \"60 km/h\"
    };
};

```

```

@Override
protected void service(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
    HttpSession session = req.getSession();
    PrintWriter out = resp.getWriter();

    String userName = req.getParameter("name");
    String userRollNo = req.getParameter("rollno");

    if (userName == null && session.getAttribute("UserName") != null) {
        userName = session.getAttribute("UserName").toString();
    } else if (userName != null) {
        session.setAttribute("UserName", userName);
    }

    if (userRollNo == null && session.getAttribute("UserRollNo") != null) {
        userRollNo = session.getAttribute("UserRollNo").toString();
    } else if (userRollNo != null) {
        session.setAttribute("UserRollNo", userRollNo);
    }

    String questionNumberStr = req.getParameter("q");

    resp.setContentType("text/html");
    resp.setCharacterEncoding("UTF-8");

    out.println("<!DOCTYPE html>");
    out.println("<html lang=\"en\">");
    out.println("<head>");
    out.println("    <meta charset=\"UTF-8\">");
    out.println("    <meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0\">");
    out.println("    <title>Play Quiz</title> <link
href='https://img.icons8.com/bubbles/50/brain.png' rel='icon'>");
    out.println("    <script src=\"https://cdn.tailwindcss.com\"></script>");
    out.println("</head>");
    out.println("<body class=\"bg-blue-50 min-h-screen\">");

```

```

        if (userName != null && userRollNo != null && questionNumberStr != null)
        {
            try {
                int questionNumber = Integer.parseInt(questionNumberStr);

                if (questionNumber < 1 || questionNumber >
apitudeQuestions.length) {
                    showError(out, "Invalid question number");
                    return;
                }

                Cookie nameCookie = new Cookie("UserName", userName);
                Cookie rollCookie = new Cookie("UserRollNo", userRollNo);
                nameCookie.setMaxAge(60*5);
                rollCookie.setMaxAge(60*5);
                resp.addCookie(nameCookie);
                resp.addCookie(rollCookie);

                out.println("<header class=\"border-b-8 border-blue-800
bg-blue-100 shadow-lg mb-6\">");
                out.println("    <div class=\"container mx-auto px-4 py-4\">");
                out.println("        <div class=\"flex flex-col md:flex-row
justify-between items-center\">");
                out.println("            <h1 class=\"text-3xl font-bold
text-blue-900\">Play Quiz</h1>");
                out.println("            <div class=\"flex flex-col sm:flex-row gap-4
mt-2 md:mt-0\">");
                out.println("                <div class=\"bg-white px-4 py-2 rounded-lg
shadow-sm border border-gray-200\">");
                out.println("                    <span class=\"text-gray-500 text-sm
font-medium\">Name:</span>");
                out.println("                    <span class=\"ml-2 font-semibold
text-gray-800\">" + userName + "</span>");
                out.println("                </div>");
                out.println("                <div class=\"bg-white px-4 py-2 rounded-lg
shadow-sm border border-gray-200\">");
                out.println("                    <span class=\"text-gray-500 text-sm
font-medium\">Roll Number:</span>");
                out.println("                    <span class=\"ml-2 font-semibold
text-gray-800\">" + userRollNo + "</span>");
                out.println("                </div>");
                out.println("            </div>");
                out.println("        </div>");
                out.println("    </div>");
                out.println("</header>");

                out.println("<div class=\"container mx-auto px-4 mb-6\">");
                out.println("    <div class=\"w-full bg-gray-200 rounded-full
h-2.5 mb-2\">");

```

```

        double progressPercentage = ((double) questionNumber /
apptitudeQuestions.length) * 100;
        out.println("    <div class=\"bg-blue-600 h-2.5 rounded-full\"
style=\"width: " + progressPercentage + "%\"></div>");
        out.println("    </div>");
        out.println("    <div class=\"flex justify-between text-xs
text-gray-600\">");
        out.println("        <span>Question " + questionNumber + " of " +
apptitudeQuestions.length + "</span>");
        out.println("        <span>" + Math.round(progressPercentage) + "%
complete</span>");
        out.println("    </div>");
        out.println("</div>");

        out.println("<main class=\"container mx-auto px-4 flex
justify-center items-center mb-12\">");
        out.println("    <div class=\"bg-white rounded-xl shadow-lg p-6
w-full max-w-md border border-gray-200\">");

        String[] questionParts = apptitudeQuestions[questionNumber -
1].replaceAll("^\"|\"$", "").split("\\", "\\");

        out.println("        <div class=\"mb-6\">");
        out.println("            <h2 class=\"text-lg font-semibold
text-gray-700 mb-4\">Question " + questionNumber + "</h2>");
        out.println("            <p class=\"text-xl font-medium text-gray-900
mb-6\">" + questionParts[0] + "</p>");
        out.println("        </div>");

        out.println("        <form action='StoreValues' method='GET'
class=\"space-y-4\">");
        out.println("            <div class=\"space-y-3\">");

        for (int i = 1; i <= 4; i++) {
            String optionValue = (i == 4) ? "1" : "0";
            String optionId = "option" + i;

            out.println("                <div class=\"flex items-center p-3
bg-gray-50 border border-gray-200 rounded-lg hover:bg-gray-100
cursor-pointer\">");
            out.println("                    <input type=\"radio\" id=\"" +
optionId + "\" name=\"o\" value=\"" + optionValue + "\" class=\"w-4 h-4
text-blue-600 bg-gray-100 border-gray-300 focus:ring-blue-500\" required>");
            out.println("                    <label for=\"" + optionId + "\"
class=\"ms-2 w-full text-gray-700 cursor-pointer\">" + questionParts[i] +
"</label>");
            out.println("                </div>");
        }

```



```

        out.println("        </div>");

        out.println("        <input type=\"hidden\" name=\"q\" value=\"" +
questionNumber + "\">");
        out.println("        <input type=\"hidden\" name=\"result\"
value=\"" + (questionNumber == aptitudeQuestions.length ? "1" : "0") + "\">");
        out.println("        <div class=\"pt-4\">");
        out.println("            <button type=\"submit\" class=\"w-full px-6
py-3 bg-blue-600 text-white font-medium rounded-lg hover:bg-blue-700
transition-colors shadow-md focus:outline-none focus:ring-4
focus:ring-blue-300\">");
            out.println("                " + (questionNumber ==
aptitudeQuestions.length ? "Submit Quiz" : "Next Question") + "");
            out.println("            </button>");
        out.println("        </div>");
        out.println("    </form>");
        out.println(" </div>");
        out.println("</main>");

    } catch (NumberFormatException e) {
        showError(out, "Invalid question number format");
    }
} else {
    showError(out, "Missing user information or question number!");
}

out.println("</body>");
out.println("</html>");
}

private void showError(PrintWriter out, String errorMessage) {
    out.println("<div class=\"flex items-center justify-center
h-screen\">");
    out.println("    <div class=\"max-w-md w-full bg-white rounded-lg
shadow-lg p-8 text-center\">");
        out.println("        <svg class=\"w-16 h-16 text-red-500 mx-auto mb-4\"
fill=\"none\" stroke=\"currentColor\" viewBox=\"0 0 24 24\"
xmlns=\"http://www.w3.org/2000/svg\">");
            out.println("            <path stroke-linecap=\"round\"
stroke-linejoin=\"round\" stroke-width=\"2\" d=\"M12 8v4m0 4h.01M21 12a9 9 0
11-18 0 9 9 0 0118 0z\"></path>");
            out.println("        </svg>");
        out.println("        <h2 class=\"text-2xl font-bold text-gray-800
mb-2\">Error</h2>");
        out.println("        <p class=\"text-gray-600 mb-6\">" + errorMessage +
"</p>");
    }
}

```

```

        out.println("    <a href=\"index.html\" class=\"px-6 py-3 bg-blue-600
text-white font-medium rounded-lg hover:bg-blue-700 transition-colors shadow-md
inline-block\">Back to Home</a>");
        out.println("  </div>");
        out.println("</div>");
    }
}

```

StoreValues.java :

```

package mygame;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.WebServlet;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/StoreValues")
public class StoreValues extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String getValue = request.getParameter("o");
        String getQuestion = request.getParameter("q");
        String result = request.getParameter("result");

        out.println("<!DOCTYPE html>");
        out.println("<html lang=\"en\">");
        out.println("<head>");
        out.println("  <meta charset=\"UTF-8\">");
        out.println("  <meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0\">");
        out.println("  <title>Quiz Results</title> <link
href='https://img.icons8.com/bubbles/50/brain.png' rel='icon'>");
        out.println("  <script src=\"https://cdn.tailwindcss.com\"></script>");
        out.println("</head>");
        out.println("<body class=\"bg-blue-50 min-h-screen\">");

        if (getValue == null || getQuestion == null || result == null) {
            out.println("<div class=\"flex items-center justify-center
h-screen\">");
            out.println("  <div class=\"bg-red-100 border-l-4 border-red-500
text-red-700 p-4 rounded shadow-md\" role=\"alert\">");
            out.println("    <p class=\"font-bold\">Error</p>");
            out.println("    <p>Missing parameters. Please try again.</p>");
            out.println("  </div>");
            out.println("</div>");
        }
    }
}

```

```

        out.println("</body></html>");
        return;
    }

    HttpSession session = request.getSession();
    session.setAttribute(getQuestion, getValue);

    int sumOfMarks = 0;

    if ("1".equals(result)) {
        out.println("<header class=\"border-b-8 border-blue-800 bg-blue-100 shadow-lg mb-8 \">>");
        out.println("    <div class=\"container mx-auto px-4 py-6\">");
        out.println("        <h1 class=\"text-3xl font-bold text-blue-900 text-center\">Quiz Results</h1>");
        out.println("    </div>");
        out.println("</header>");

        out.println("<main class=\"container mx-auto px-4 py-6 max-w-4xl\">");
        out.println("    <div class=\"bg-white rounded-lg shadow-md p-6 mb-8\">

            try {
                int totalQuestions = Integer.parseInt(getQuestion);
                out.println("        <h2 class=\"text-xl font-semibold mb-4 text-gray-800\">Your Answers</h2>");
                out.println("        <div class=\"space-y-4\">

                    for (int i = 1; i <= totalQuestions; i++) {
                        String answer = (String)
session.getAttribute(String.valueOf(i));

                        if (answer != null) {
                            boolean isCorrect = answer.trim().equals("1");
                            sumOfMarks += Integer.parseInt(answer);

                            out.println("                <div class=\"p-4 \" + (isCorrect ?
\"bg-green-50 border-l-4 border-green-500\" : \"bg-red-50 border-l-4 border-red-500\") + \" rounded\">");
                            out.println("                <div class=\"flex items-center\">

                                out.println("                    <div class=\"flex-shrink-0\">");
                                out.println("                    <svg class=\"h-5 w-5 \" +
(isCorrect ? \"text-green-500\" : \"text-red-500\") + \"\" fill=\"currentColor\"
viewBox=\"0 0 20 20\">");
                                    if (isCorrect) {
                                        out.println("                        <path
fill-rule=\"evenodd\" d=\"M10 18a8 8 0 10-16 8 8 0 00 16zm3.707-9.293a1 1 0

```

```

00-1.414-1.414L9 10.586 7.707 9.293a1 1 0 00-1.414 1.414L12 2a1 1 0 001.414
014-4z\" clip-rule=\"evenodd\"/>");
        } else {
            out.println("                <path
fill-rule=\"evenodd\" d=\"M10 18a8 8 0 100-16 8 8 0 000 16zM8.707 7.293a1 1 0
00-1.414 1.414L8.586 101-1.293 1.293a1 1 0 101.414 1.414L10 11.41411.293
1.293a1 1 0 001.414-1.414L11.414 1011.293-1.293a1 1 0 00-1.414-1.414L10 8.586
8.707 7.293z\" clip-rule=\"evenodd\"/>");
        }

        out.println("                </svg>");
        out.println("            </div>");
        out.println("        <div class=\"ml-3\">");
        out.println("            <p class=\"text-sm font-medium
" + (isCorrect ? "text-green-800" : "text-red-800") + "\">Question " + i +
"</p>");

            out.println("                <p class=\"text-sm " +
(isCorrect ? "text-green-700" : "text-red-700") + "\">" + (isCorrect ? "Correct
answer!" : "Incorrect answer") + "</p>");
            out.println("            </div>");
            out.println("        </div>");
            out.println("    </div>");
        } else {
            out.println("        <div class=\"p-4 bg-gray-50
border-l-4 border-gray-300 rounded\">");
            out.println("            <p class=\"text-gray-700\">No
answer recorded for Question " + i + "</p>");
            out.println("        </div>");
        }
    }
    out.println("    </div>");

    double percentage = (double) sumOfMarks / totalQuestions * 100;
    String scoreColorClass = percentage >= 70 ? "bg-green-100
text-green-800" :
        (percentage >= 40 ? "bg-yellow-100 text-yellow-800" :
"bg-red-100 text-red-800");

    out.println("    <div class=\"mt-8 p-6 rounded-lg " +
scoreColorClass + "\">");
    out.println("        <h2 class=\"text-2xl font-bold mb-2\">Your
Score</h2>");
    out.println("        <div class=\"flex items-center\">");
    out.println("            <div class=\"text-4xl font-bold mr-4\">" +
sumOfMarks + "/" + totalQuestions + "</div>");
    out.println("            <div class=\"ml-4\">");
    out.println("                <div class=\"w-64 bg-gray-200
rounded-full h-4 overflow-hidden\">");

```

```

        out.println("                <div class=\"h-full \" + (percentage >=
70 ? \"bg-green-500\" : (percentage >= 40 ? \"bg-yellow-500\" : \"bg-red-500\")) +
\"\" style=\"width: \" + percentage + \"%\"></div>");
        out.println("                </div>");
        out.println("                <p class=\"mt-1 text-sm\">\" +
String.format("%.1f", percentage) + "%</p>");
        out.println("                </div>");
        out.println("                </div>");
        out.println("        </div>");

    } catch (NumberFormatException e) {
        out.println("        <div class=\"bg-red-100 border-l-4
border-red-500 text-red-700 p-4 rounded\">");
        out.println("                <p class=\"font-bold\">Error</p>");
        out.println("                <p>Invalid question number format.</p>");
        out.println("        </div>");
    }

    out.println("    </div>");

    out.println("    <div class=\"flex justify-center gap-4 mt-6\">");
    out.println("        <a href=\"index.html\" class=\"px-6 py-3
bg-blue-600 text-white font-medium rounded-lg hover:bg-blue-700
transition-colors shadow-md\">Back to Home</a>");
    out.println("        <a href=\"PlayQuiz?q=1\" class=\"px-6 py-3
bg-green-600 text-white font-medium rounded-lg hover:bg-green-700
transition-colors shadow-md\">Try Again</a>");
    out.println("    </div>");
    out.println("</main>");

    } else {
        try {
            int nextQuestion = Integer.parseInt(getQuestion) + 1;
            response.sendRedirect("PlayQuiz?q=" + nextQuestion);
        } catch (NumberFormatException e) {
            out.println("<div class=\"flex items-center justify-center
h-screen\">");
            out.println("        <div class=\"bg-red-100 border-l-4 border-red-500
text-red-700 p-4 rounded shadow-md\" role=\"alert\">");
            out.println("                <p class=\"font-bold\">Error</p>");
            out.println("                <p>Invalid question number format.</p>");
            out.println("        </div>");
            out.println("</div>");
        }
    }
}

out.println("</body>");

```

```
out.println("</html>");
Cookie[] cookie=request.getCookies();
for (Cookie c:cookie) {
    if(c.getName().equals("name")||c.getName().equals("rollno")) {
        if(c.getValue()==null){
            response.sendRedirect("index.html");
        }
    }
}
}
```

Output :

Play Quiz

Roll No:

6

Name:

Benjamin

Play Quiz

Play Quiz

Name: Benjamin

Roll Number: 6

Question 5 of 10

50% complete

Question 5

A train covers 360 km in 6 hours. What is its speed?

☐ 30 km/h

☐ 65 km/h

☐ 50 km/h

☒ 60 km/h

Next Question

Quiz Results

Your Answers

✓ Question 1
Correct answer!

✗ Question 2
Incorrect answer

✓ Question 3
Correct answer!

✓ Question 4
Correct answer!

✓ Question 5
Correct answer!

✓ Question 6
Correct answer!

✓ Question 7
Correct answer!

✗ Question 8
Incorrect answer

✓ Question 9
Correct answer!

✓ Question 10
Correct answer!

Your Score

8/10

80.0%

[Back to Home](#)

[Try Again](#)

Practical No:5

Objective :Program to demonstrate Page directives with JSP.

Program :

directives.jsp :

```
<%@ page import="java.util.Date" %>
<%@ page contentType="text/html" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html"; charset="UTF-8">
    <title>Page Directives Demo</title>
</head>
<body>
<h1>Page Directives Demo</h1>
<p>Current Date and Time: <%= new Date() %> </p>
</body>
</html>
```

Output :

Page Directives Demo

Current Date and Time: Fri Jun 06 20:15:50 IST 2025

Practical No:6

Objective : Program to demonstrate Http session using cookies.

Program :

pom.xml :

```
<dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <version>6.1.0</version>
</dependency>
```

ServletLogin.java :

```
package org.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/index.html")
public class ServletLogin extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>ServletLogin</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet Login</h1>");

        if (req.getParameter("email") == null || req.getParameter("password") ==
null) {
            out.println("<form action=\"index.html\" method=\"post\">");
            out.println("<input type=\"email\" name=\"email\" "
required><br><br>");
            out.println("<input type=\"password\" name=\"password\" "
required><br><br>");
            out.println("<input type=\"submit\" value=\"Submit\">");
            out.println("</form>");
        }
        out.println("</body>");
    }
}
```

```

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        if
        (req.getParameter("email").equals("ben@google.com") && req.getParameter("password")
        .equals("12345")) {
            Cookie cookie = new Cookie("email", req.getParameter("email"));
            cookie.setMaxAge(3600);
            resp.addCookie(cookie);
            HttpSession session = req.getSession();
            session.setAttribute("password", req.getParameter("password"));

            String url = resp.encodeURL("servletdata?count=100");
            resp.sendRedirect(url);

        } else {
            PrintWriter out = resp.getWriter();
            out.println("<h2>Invalid email or password</h2>");
        }
    }
}

```

ServletData.java :

```

package org.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/servletdata")
public class ServletData extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        HttpSession session = req.getSession();
        String sessionData = session.getAttribute("password").toString();
        Cookie[] cookies = req.getCookies();
        String email = null;

        if (cookies != null) {
            for (Cookie cookie : cookies) {
                if ("email".equals(cookie.getName())) {
                    email = cookie.getValue();
                    break;
                }
            }
        }
    }
}

```

```

        }
    }
    String urldata = req.getParameter("count");
    resp.setContentType("text/html");
    PrintWriter out = resp.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>ServletData</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>ServletData</h1>");
    out.println("<p> Session data : " + sessionData + "</p>");
    out.println("<p> Cookie data : " + email + "</p>");
    out.println("<p> Url data : " + urldata + "</p>");
    out.println("</body>");
    out.println("</html>");
}
}

```

Output :

Servlet Login ServletData

Session data : 12345

Cookie data : ben@google.com

Url data : 100

Practical No:7

Objective : Program to demonstrate Error Page in JSP.

Program :

index.jsp :

```
<%@ page import="java.util.Date" %>
<%@ page contentType="text/html" pageEncoding="UTF-8" %>
<%@ page errorPage="error.jsp" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html"; charset="UTF-8">
    <title>Page Directives Demo</title>
</head>
<body>
<h1>Page Directives </h1>
<p>Current Date and Time: <%= new Date() %> </p>
<%
int x=100/0;
%>
</body>
</html>
```

error.jsp :

```
<%@ page isErrorPage="true" %>
<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
    <title>Error Page</title>
</head>
<body>
<h2>Error Occurred!</h2>
<p>Exception: <%= exception.getClass().getName() %></p>
<p>Message: <%= exception.getMessage() %></p>
</body>
</html>
```

Output :

Error Occurred!

Exception: java.lang.ArithmeticException

Message: / by zero

Practical No:8

Objective : Program to demonstrate JSP Actions, JSP with beans.

Program :

Student.java :

```
package org.example;

public class Student {
    private String name;
    private String email;

    public Student() {}
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

login.jsp :

```
<html>
<head>
    <title>Student Login</title>
</head>
<body>
<h2>Enter Student Info</h2>
<form action="studentResult.jsp" method="post">
    Name: <input type="text" name="name"><br><br>
    Email: <input type="email" name="email"><br><br>
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

studentResult.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="org.example.Student" %>
<html>
<head>
```

```
<title>Student Details</title>
</head>
<body>
<jsp:useBean id="student1" class="org.example.Student" scope="request" />
<jsp:setProperty name="student1" property="*" />

<h2>Student Information</h2>
Name: <jsp:getProperty name="student1" property="name" /><br>
Email: <jsp:getProperty name="student1" property="email" /><br>
</body>
</html>
```

Output :

Enter Student Info

Name:

Email:

Student Information

Name: Ben

Email: benjaminsebastian156@gmail.com

Practical No:9

Objective : Program to demonstrate to custom tags, JSTL tags.

Program :

pom.xml :

```
<dependencies>
  <dependency>
    <groupId>jakarta.servlet.jsp</groupId>
    <artifactId>jakarta.servlet.jsp-api</artifactId>
    <version>3.0.0</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>2.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>2.0.0</version>
  </dependency>
  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>9.2.0</version>
  </dependency>
</dependencies>
```

Simple.tld:

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/jakartaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/jakartaee
  https://jakarta.ee/xml/ns/jakartaee/web-jsptaglibrary_3_0.xsd"
  version="3.0">

  <tlib-version>1.0</tlib-version>
  <short-name>hello</short-name>
  <uri>http://example.com/hello</uri>

  <tag>
    <name>hello</name>
    <tag-class>org.example.tags.GreetingTag</tag-class>
    <body-content>empty</body-content>
    <attribute>
```



```

        <name>name</name>
    </attribute>
    <attribute>
        <name>password</name>
    </attribute>
</tag>
</taglib>

```

GreetingTag.java :

```

package org.example.tags;

import jakarta.servlet.jsp.JspException;
import jakarta.servlet.jsp.tagext.SimpleTagSupport;

import java.io.IOException;

public class GreetingTag extends SimpleTagSupport {
    public String name;
    public String password;

    public void setName(String name) {
        this.name = name;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public void doTag() throws JspException, IOException {

        getJspContext().getOut().println("<h1>Hello " + name + " password : "
        +password+" !</h1>");
    }
}

```

coreTag.jsp :

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page import="java.util.*" %>
<%
List<String> Fruits =new ArrayList<>();
Fruits.add("Apple");
Fruits.add("Orange");
Fruits.add("Mango");
Fruits.add("Lychee");

```

```
Fruits.add("Chikku");
request.setAttribute("Fruits", Fruits);
%>

<html>
<head>
    <title>CoreTags
    </title>
</head>
<body>
<c:forEach var="Fruit" items="${Fruits}">
    <h1>${Fruit}</h1>
</c:forEach>
</body>
</html>
```

Output :

Hello BenjaminSeb password : 123 !

Apple

Orange

Mango

Lychee

Chikku

Practical No:10

Objective : Program to demonstrate simple XML tags

Program :

XMLCreation.java :

```
package org.example;

import org.w3c.dom.Document;
import org.w3c.dom.Element;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.util.Scanner;

public class XmlCreation {
    public static void main(String[] args) throws ParserConfigurationException,
TransformerException {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter how many student details you would like to enter
: ");
        int n = sc.nextInt();
        for (int i = 0; i < n; i++) {
            System.out.print("Id : " );
            int id = sc.nextInt();
            System.out.print("Name : " );
            String name = sc.next();
            System.out.print("Course : " );
            String course = sc.next();
            System.out.print("Age : " );
            int age = sc.nextInt();
            createStudent(id,name,age,course);
            System.out.println("\u001B[31m"+"Student created in
student.xml!");
        }

    }

    public static void createStudent(int id,String name,int age ,String
courseVar) throws ParserConfigurationException, TransformerException {
        Document document=
DocumentBuilderFactory.newInstance().newDocumentBuilder().newDocument();

        Element root=document.createElement("university");
```

```

document.appendChild(root);

Element student = document.createElement("student");
student.setAttribute("id", String.valueOf(id));
student.setAttribute("name", name);
student.setAttribute("age", String.valueOf(age));

Element course = document.createElement("course");
course.setTextContent(courseVar);
student.appendChild(course);
student.appendChild(document.createComment("This is a comment of student
"+id));
root.appendChild(student);

File dir = new File("src/main/resources/data");
File xmlFile=new File(dir,"students.xml");
TransformerFactory tf = TransformerFactory.newInstance();
Transformer transformer=tf.newTransformer();
DOMSource source=new DOMSource(document);
StreamResult result=new StreamResult(xmlFile);
transformer.transform(source,result);
}
}

```

students.xml :

```

<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student>
    <roll>101</roll>
    <name>Benjamin Seb</name>
    <course>Computer Science</course>
  </student>
</students>

```

XMLReader.java :

```

package org.example;

import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.*;

public class XMLReader {
    public static void main(String[] args) throws Exception {
        File file = new File("src/main/resources/data/students.xml");
        DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

```

```

Document doc = dBuilder.parse(file);
doc.getDocumentElement().normalize();

NodeList list = doc.getElementsByTagName("student");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    Element element = (Element) node;

    System.out.println("Roll: " +
element.getElementsByTagName("roll").item(0).getTextContent());
    System.out.println("Name: " +
element.getElementsByTagName("name").item(0).getTextContent());
    System.out.println("Course: " +
element.getElementsByTagName("course").item(0).getTextContent());
    System.out.println();
}
}
}

```

Output :

Enter how many student details you would like to enter : 1

Id : 1001

Name : Benjamin

Course : MCA

Age : 21

Student created in student.xml!

Process finished with exit code 0

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<university>
```

```
    <student age="21" id="1001" name="Benjamin">
```

```
        <course>MCA</course><!--This is a comment of student 1001-->
```

```
    </student>
```

```
</university>
```

"E:\User Applications\Java 24\jdk-24.

Roll: 101

Name: Benjamin Seb

Course: Computer Science

Practical No:11

Objective : Program to demonstrate DTD, XML Parser, Validator.

Program :

students.dtd :

```
<!ELEMENT students (student+)>
<!ELEMENT student (id,name,grade)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT grade (#PCDATA)>
```

students.xml :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE students SYSTEM "students.dtd">
<students>
  <student>
    <id>1</id>
    <name>Benjamin</name>
    <grade>A</grade>
  </student>
</students>
```

XmlValidate :

```
package org.example;

import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.DefaultHandler;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

public class XmlValidate extends DefaultHandler {
    public static void main(String[] args) {
        String xmlFile="src/main/resources/data/students.xml";

        try{
            XmlValidate validator = new XmlValidate();
            validator.validate(xmlFile);
            System.out.println("Validation successful");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```

private void validate(String xmlFile) throws Exception {
    SAXParserFactory factory = SAXParserFactory.newInstance();
    factory.setValidating(true);
    factory.setNamespaceAware(true);

    SAXParser parser = factory.newSAXParser();
    XMLReader reader = parser.getXMLReader();
    reader.setErrorHandler(this);

    reader.parse(new InputSource(xmlFile));
}

public void error(SAXParseException exception) throws SAXException {
    System.err.println(exception.getMessage());
    throw new SAXException("SAXParseException", exception);
}
}

```

Output :

```

"E:\User Applications\Java 24\jdk-24.0.1\
Validation successful

```

```

Process finished with exit code 0

```

```

"E:\User Applications\Java 24\jdk-24.0.1\bin\java.exe" "-javaagent:
The content of element type "student" must match "(id,name,grade)".
SAXParseException

```

```

Process finished with exit code 0

```

Practical No:12

Objective : Program to Write a Hibernate Application, Configuring Hibernate for our application.

Program :

pom.xml :

```
<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>
    <dependency>
        <groupId>org.hibernate.orm</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>7.0.0.Alpha1</version>
    </dependency>
</dependencies>
```

hibernate.cfg.xml :

```
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/studentsdbhibernate
</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password"></property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <mapping class="org.example.Student"/>

    </session-factory>
</hibernate-configuration>
```

Student.java :

```
package org.example;

import jakarta.persistence.*;

@Entity
@Table(name ="students_data")
public class Student {
    public Student() {

    }
    public Student(String id,String firstName, String lastName, String
email,String date) {
        this.firstName=firstName;
        this.lastName=lastName;
    }
}
```



```

        this.email=email;
        this.date=date;
        this.id= id;
    }
    @Id //primary key
    //@GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="Id")
    private String id;
    @Column(name="First_Name")
    private String firstName;
    @Column(name="Last_Name")
    private String lastName;
    @Column(name="Email")
    private String email;
    @Column(name="date")
    private String date;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

```

```

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

```

Main.java :

```

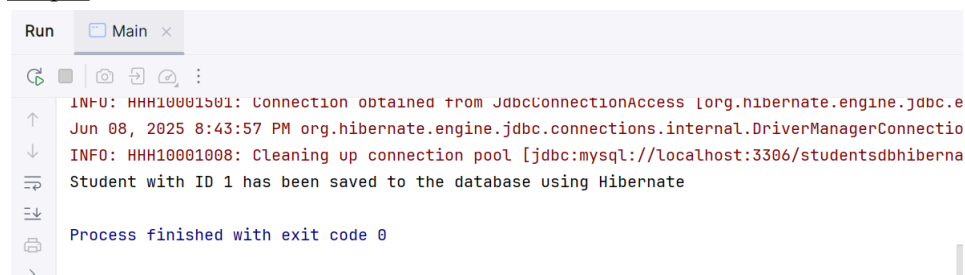
package org.example;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Main {
    public static void main(String[] args) {
        try{
            Configuration config = new Configuration().configure();
            SessionFactory sessionFactory = config.buildSessionFactory();
            Session session = sessionFactory.openSession();
            session.beginTransaction();
            Student student = new Student();
            student.setFirstName("Benjamin");
            student.setLastName("Sebastian");
            student.setEmail("b.sebastian@somaiya.edu");
            student.setId("1");
            student.setDate("1-1-2025");
            session.save(student);
            session.getTransaction().commit();
            session.close();
            sessionFactory.close();
            System.out.println("Student with ID " + student.getId() + " has been
saved to the database using Hibernate");
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

Output :



```

Run Main x
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.e
Jun 08, 2025 8:43:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/studentsdbhiberna
Student with ID 1 has been saved to the database using Hibernate
Process finished with exit code 0

```

Practical No:13

Objective : Program to fetch data from the database using Hibernate and Annotations.

Program :

pom.xml :

```
<dependencies>
  <dependency>
    <groupId>org.hibernate.orm</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>7.0.0.Alpha1</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
  </dependency>
</dependencies>
```

hibernate.cfg.xml :

```
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/studentsdbhibernate
</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <mapping class="org.example.Student"/>

  </session-factory>
</hibernate-configuration>
```

Student.java :

```
package org.example;

import jakarta.persistence.*;

@Entity
@Table(name = "students_data")
public class Student {
    public Student() {

    }

    public Student(String id, String firstName, String lastName, String email,
String date) {
        this.firstName=firstName;
    }
}
```

```

        this.lastName=lastName;
        this.email=email;
        this.date=date;
        this.id= id;
    }
    @Id //primary key
    //@GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="Id")
    private String id;
    @Column(name="First_Name")
    private String firstName;
    @Column(name="Last_Name")
    private String lastName;
    @Column(name="Email")
    private String email;
    @Column(name="date")
    private String date;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}

```

```

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

```

Main.java :

```

package org.example;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        try{
            Configuration configuration = new
Configuration().configure("hibernate.cfg.xml");
            SessionFactory sessionFactory = configuration.buildSessionFactory();
            Session session = sessionFactory.openSession();
            Transaction transaction = session.beginTransaction();
            Scanner scanner = new Scanner(System.in);
            boolean boo = true;
            while(boo) {
                System.out.print("Choose Menu : \n 1: Insert \n 2: Fetch \n 3:
Fetch all\n 4:Exit \n your selection : ");

                int choice = scanner.nextInt();
                if (choice == 1) {
                    System.out.print("Enter no of student u want to enter : ");
                    int count = scanner.nextInt();
                    for (int i = 0; i < count; i++) {
                        Student student = new Student();
                        System.out.print("Enter the id of student " + (i+1) +
":");

                        student.setId(scanner.next());
                        System.out.println();

```

```

        System.out.print("Enter the first name of student " +
(i+1) + ":");

        student.setFirstName(scanner.next());
        System.out.println();
        System.out.print("Enter the last name of student " +
(i+1) + ":");

        student.setLastName(scanner.next());
        System.out.println();
        System.out.print("Enter the email of student " + (i+1) +
":");

        student.setEmail(scanner.next());
        System.out.println();
        System.out.print("Enter the date of student " + (i+1) +
":");

        student.setDate(scanner.next());
        System.out.println();
        System.out.println("Data Inserted");
        session.persist(student);
        transaction.commit();
        session.close();

    }
} else if (choice == 2) {
    System.out.print("Enter the id of student :");
    Student student = session.get(Student.class,
scanner.next());

    System.out.println("id of student :"+student.getId());
    System.out.println("first name of student
:"+student.getFirstName());
    System.out.println("last name of student
:"+student.getLastName());
    System.out.println("email of student :"+student.getEmail());
    System.out.println("date of student :"+student.getDate());

} else if (choice == 3) {
    String hqlQuery = "from Student " ;
    List<Student> students=new ArrayList<>();
    students=session.createQuery(hqlQuery).getResultList();
    for (Student student : students) {
        System.out.println("id of student :"+student.getId());
        System.out.println("first name of student
:"+student.getFirstName());
        System.out.println("last name of student
:"+student.getLastName());
        System.out.println("email of student
:"+student.getEmail());
        System.out.println("date of student
:"+student.getDate());
    }
}

```

```

        } else {
            return;
        }
        System.out.print(" Do you want to exit ? \n 1: yes \n 2: no \n
your selection : ");
        int choice2 = scanner.nextInt();
        if (choice2 == 1) {boo=false;}
        System.out.println();
    }
}

catch(Exception e){
    System.out.println(e.getMessage());
}

}

}

```

Output :

```

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.e
Choose Menu :
1: Insert
2: Fetch
3: Fetch all
4:Exit
your selection : 3
id of student :1
first name of student :Benjamin
last name of student :Sebastian
email of student :b.sebastian@somaiya.edu
date of student :1-1-2025
Do you want to exit ?
1: yes
2: no
your selection : 1

Process finished with exit code 0

```

Practical No:14

Objective : Building first Spring Application.

Program :

pom.xml :

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.32</version>
  </dependency>
</dependencies>
```

applicationContext.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="helloService" class="org.example.HelloService">
    <property name="message" value="Welcome to Spring Framework!" />
  </bean>
</beans>
```

HelloService.java :

```
package org.example;

public class HelloService {

    private String message;

    public void setMessage(String message) {
        this.message = message;
    }

    public void sayHello() {
        System.out.println("Spring says: " + message);
    }

}
```

Main.java :

```
package org.example;
```

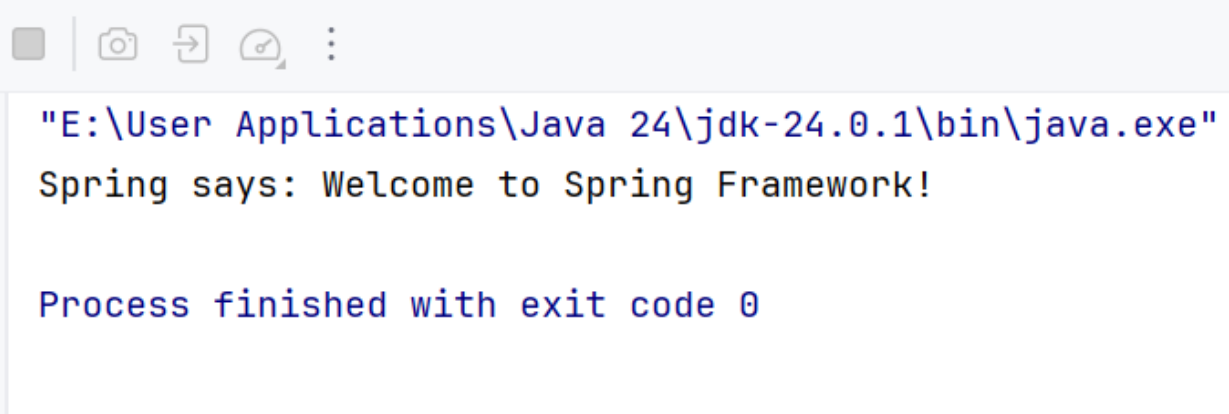


```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {
    public static void main(String[] args) {
        ApplicationContext context = new
            ClassPathXmlApplicationContext("applicationContext.xml");

        HelloService service = (HelloService)
context.getBean("helloService");
        service.sayHello();
    }
}
```

Output :



```
"E:\User Applications\Java 24\jdk-24.0.1\bin\java.exe"
Spring says: Welcome to Spring Framework!

Process finished with exit code 0
```

Practical No:15

Objective : Simple web application using Spring

Program :

pom.xml :

```
<dependencies>
  <dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <version>4.0.2</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>6.1.5</version>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>3.3.1</version> <!-- or 3.2.3 -->
    </plugin>
  </plugins>
</build>
```

dispatcher-servlet.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd">
  <context:component-scan base-package="controllers"/>
  <mvc:annotation-driven/>
  <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/" />
    <property name="suffix" value=".jsp" />
```

```
</bean>
```

```
</beans>
```

web.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
<servlet>
    <servlet-name>
        dispatcher
    </servlet-name>
    <servlet-class>
        org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <init-param>
        <param-name>
            contextConfigLocation
        </param-name>
        <param-value>
            /WEB-INF/dispatcher-servlet.xml
        </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

LoginController.java :

```
package controllers;

import controllers.classfiles.Question;
import controllers.classfiles.Quiz;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

@Controller
public class LoginController {
```

```

@RequestMapping("/login")
public String sayhello(Model model){
    model.addAttribute("message","This is a page loaded using Spring MVC");
    return "login";
}
@RequestMapping("/quiz")
public String quizHome(Model model){

    return "quiz-landing";
}
@RequestMapping("/quiz-playground")
public String quizPlayground(Model model){
    List<Question> questions = new ArrayList<>();

    String[] q1Options = {"2 bytes", "4 bytes", "8 bytes", "Depends on OS"};
    String[] q2Options = {"this", "import", "extends", "instanceof"};
    String[] q3Options = {"To define a constant", "To inherit a class", "To
start a loop", "To throw an exception"};
    String[] q4Options = {"true", "false", "0", "null"};
    String[] q5Options = {"To call parent class constructor", "To define a
subclass", "To stop execution", "To catch exceptions"};
    String[] q6Options = {"Overriding methods", "Using variables", "Method
overloading", "Both A and C"};
    String[] q7Options = {"HashSet", "ArrayList", "TreeSet", "None of the
above"};
    String[] q8Options = {"They are the same", "== compares references,
.equals() compares values", ".equals() compares references", "== compares
object content"};
    String[] q9Options = {"Using try-catch blocks", "Only by using throws",
"With main method only", "Using finally block only"};
    String[] q10Options = {"ArrayList is thread-safe", "LinkedList is faster
for random access", "ArrayList uses array internally, LinkedList uses nodes",
"Both are same"};
    int[] correctAnswers = {1, 2, 0, 1, 0, 3, 1, 1, 0, 2};
    questions.add(new Question(1,"What is the size of int in
Java?",correctAnswers[0],q1Options));
    questions.add(new Question(2," Which keyword is used to inherit a class
in Java?",correctAnswers[1],q2Options));
    questions.add(new Question(3," What is the purpose of the final keyword
in Java?",correctAnswers[2],q3Options));
    questions.add(new Question(4,"What is the default value of a boolean
variable in Java?",correctAnswers[3],q4Options));
    questions.add(new Question(5," What is the use of the super
keyword?",correctAnswers[4],q5Options));
    questions.add(new Question(6," What is polymorphism in
OOP?",correctAnswers[5],q6Options));
    questions.add(new Question(7,"Which collection class allows duplicate
elements?",correctAnswers[6],q7Options));

```

```

        questions.add(new Question(8,"What is the difference between == and
.equals()?",correctAnswers[7],q8Options));
        questions.add(new Question(9,"How does exception handling work in
Java?",correctAnswers[8],q9Options));
        questions.add(new Question(10,"What is the difference between ArrayList
and LinkedList?",correctAnswers[9],q10Options));

        List<Quiz> quizzes = new ArrayList<>();
        quizzes.add(new Quiz("Java",questions));

        model.addAttribute("quiz",quizzes);

        return "quiz-playground";
    }

@PostMapping("/quizresult")
public String submitQuiz(@RequestParam Map<String, String> params, Model model)
{
    List<Quiz> quizzes = generateQuiz(); //  Reuse your method
    List<Question> allQuestions = quizzes.get(0).getQuestions();

    List<Integer> selectedAnswers = new ArrayList<>();
    List<Integer> correctAnswers = new ArrayList<>();

    for (Question q : allQuestions) {
        String param = params.get("q" + q.getId());
        if (param != null) {
            try {
                selectedAnswers.add(Integer.parseInt(param));
                correctAnswers.add(q.getAnswer());
            } catch (NumberFormatException ignored) {}
        }
    }

    model.addAttribute("answers", selectedAnswers);
    model.addAttribute("correctAnswers", correctAnswers);
    model.addAttribute("quiz", quizzes); //  Send the quiz back to JSP
    return "quiz-result";
}

private List<Quiz> generateQuiz() {
    List<Question> questions = new ArrayList<>();

    String[] q1Options = {"2 bytes", "4 bytes", "8 bytes", "Depends on OS"};
    String[] q2Options = {"this", "import", "extends", "instanceof"};
    String[] q3Options = {"To define a constant", "To inherit a class", "To
start a loop", "To throw an exception"};
    String[] q4Options = {"true", "false", "0", "null"};
    String[] q5Options = {"To call parent class constructor", "To define a
subclass", "To stop execution", "To catch exceptions"};

```

```

        String[] q6Options = {"Overriding methods", "Using variables", "Method overloading", "Both A and C"};
        String[] q7Options = {"HashSet", "ArrayList", "TreeSet", "None of the above"};
        String[] q8Options = {"They are the same", "== compares references, .equals() compares values", ".equals() compares references", "== compares object content"};
        String[] q9Options = {"Using try-catch blocks", "Only by using throws", "With main method only", "Using finally block only"};
        String[] q10Options = {"ArrayList is thread-safe", "LinkedList is faster for random access", "ArrayList uses array internally, LinkedList uses nodes", "Both are same"};

        int[] correctAnswers = {1, 2, 0, 1, 0, 3, 1, 1, 0, 2};

        questions.add(new Question(1, "What is the size of int in Java?", correctAnswers[0], q1Options));
        questions.add(new Question(2, "Which keyword is used to inherit a class in Java?", correctAnswers[1], q2Options));
        questions.add(new Question(3, "What is the purpose of the final keyword in Java?", correctAnswers[2], q3Options));
        questions.add(new Question(4, "What is the default value of a boolean variable in Java?", correctAnswers[3], q4Options));
        questions.add(new Question(5, "What is the use of the super keyword?", correctAnswers[4], q5Options));
        questions.add(new Question(6, "What is polymorphism in OOP?", correctAnswers[5], q6Options));
        questions.add(new Question(7, "Which collection class allows duplicate elements?", correctAnswers[6], q7Options));
        questions.add(new Question(8, "What is the difference between == and .equals()?", correctAnswers[7], q8Options));
        questions.add(new Question(9, "How does exception handling work in Java?", correctAnswers[8], q9Options));
        questions.add(new Question(10, "What is the difference between ArrayList and LinkedList?", correctAnswers[9], q10Options));

        List<Quiz> quizzes = new ArrayList<>();
        quizzes.add(new Quiz("Java", questions));

        return quizzes;
    }
}

```

Quiz.java :

```

package controllers.classfiles;

import java.util.List;

```

```

public class Quiz {
    String title;
    List<Question> questions;

    public Quiz(String title, List<Question> questions) {
        this.title = title;
        this.questions = questions;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public List<Question> getQuestions() {
        return questions;
    }

    public void setQuestions(List<Question> questions) {
        this.questions = questions;
    }
}

```

Question.java :

```

package controllers.classfiles;

public class Question {
    int id;
    String question;
    String[] options;
    int answer;

    public Question(int id, String question, int answer, String[] options) {
        this.id = id;
        this.question = question;
        this.answer = answer;
        this.options = options;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}

```

```

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public String[] getOptions() {
        return options;
    }

    public void setOptions(String[] options) {
        this.options = options;
    }

    public int getAnswer() {
        return answer;
    }

    public void setAnswer(int answer) {
        this.answer = answer;
    }
}

```

Login.jsp :

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 flex items-center justify-center min-h-screen">

<div class="bg-white p-8 rounded-2xl shadow-xl w-full max-w-md">
    <h2 class="text-2xl font-bold mb-6 text-center text-gray-800">Login</h2>

    <form class="space-y-4">
        <div>
            <label for="username" class="block text-gray-600 mb-1">Username</label>
            <input type="text" id="username" name="username" required
                class="w-full px-4 py-2 border rounded-xl focus:outline-none
focus:ring-2 focus:ring-blue-500" />
        </div>

        <div>

```



```

        <label for="password" class="block text-gray-600 mb-1">Password</label>
        <input type="password" id="password" name="password" required
            class="w-full px-4 py-2 border rounded-xl focus:outline-none
focus:ring-2 focus:ring-blue-500" />
    </div>

    <button type="submit"
        class="w-full bg-blue-500 text-white py-2 rounded-xl
hover:bg-blue-600 transition duration-300">
        Log In
    </button>
</form>

<p class="mt-6 text-center text-sm text-gray-500">
    Don't have an account? <a href="#" class="text-blue-500
hover:underline">Register</a>
</p>
<p class="mt-6 text-center text-sm text-gray-500">${message}</p>
</div>

</body>
</html>

```

quiz-landing.jsp :

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Quiz App - Home</title>
    <link href="https://img.icons8.com/cute-clipart/64/ask-question.png"
rel="icon">
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gradient-to-br from-blue-100 to-blue-300 min-h-screen flex
items-center justify-center">

<div class="bg-white p-10 rounded-2xl shadow-2xl max-w-lg w-full text-center">
    <h1 class="text-4xl font-bold text-gray-800 mb-4">Welcome to the Quiz
App!</h1>
    <p class="text-gray-600 text-lg mb-8">Test your knowledge across various
topics. Click below to begin!</p>

    <a href="quiz-playground"
        class="inline-block bg-blue-600 text-white text-lg font-semibold px-6
py-3 rounded-full shadow-lg hover:bg-blue-700 transition duration-300">
        Start Quiz
    </a>

```

```
</div>
```

```
</body>
```

```
</html>
```

quiz-playground.jsp :

```
<%@ page import="java.util.*, controllers.classfiles.Quiz,
controllers.classfiles.Question" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Available Quizzes</title>
    <link href="https://img.icons8.com/cute-clipart/64/ask-question.png"
rel="icon">

    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 min-h-screen flex items-center justify-center p-6">

<div class="bg-white p-10 rounded-2xl shadow-2xl w-full max-w-5xl">
    <h1 class="text-3xl font-bold text-center text-gray-800 mb-8">Available
Quizzes</h1>

    <!-- Quiz Cards -->
    <div class="grid grid-cols-1 md:grid-cols-2 gap-6">
        <%
            List<Quiz> quizzes = (List<Quiz>) request.getAttribute("quiz");
            if (quizzes != null && !quizzes.isEmpty()) {
                for (Quiz quiz : quizzes) {
                    %>
                        <form action="quizresult" method="post" class="bg-blue-50 p-6 rounded-xl
shadow flex flex-col">
                            <span class="text-xl font-semibold text-gray-800 mb-2"><%=
quiz.getTitle() %> Quiz</span>
                            <span class="text-sm text-gray-600 mb-4">Questions: <%=
quiz.getQuestions().size() %></span>

                            <!-- Send quiz title as hidden -->
                            <input type="hidden" name="quizTitle" value="<%= quiz.getTitle()
%>">

                            <input type="hidden" name="quiz" value="<%= quiz %>">

                            <div class="space-y-4">
                                <%
                                    int questionCounter = 0;
                                    for (Question q : quiz.getQuestions()) {
```

```

        %>

        <div class="bg-white p-4 rounded-lg border border-gray-200">
            <p class="text-md font-medium text-gray-700 mb-2">
                <%= ++questionCounter %>. <%= q.getQuestion() %>
                <input name="answer<%=q.getId()%>"
value="<%=q.getAnswer()%>" hidden="hidden">

            </p>

            <div class="space-y-1 ml-4">
                <%

                    String[] options = q.getOptions();
                    for (int i = 0; i < options.length; i++) {

                        %>

                            <label class="flex items-center text-sm text-gray-600">
                                <input type="radio" name="q<%=q.getId()%>"
value="<%= i %>" class="mr-2" required>

                                    <%= options[i] %>
                                </label>
                                <% } %>
                            </div>
                        </div>
                    <% } %>
                </div>

                <button type="submit" class="mt-6 bg-green-600 text-white px-4 py-2
rounded-xl hover:bg-green-700 transition">
                    Submit Quiz
                </button>
            </form>
            <%

                }
            } else {
                %>
                <p class="text-red-500 font-semibold">No quizzes available.</p>
                <%

                    }
                %>
            </div>
        </div>

    </body>
</html>

quiz-result.jsp :
<%@ page import="java.util.*, controllers.classfiles.Question" %>

```

```

<%@ page import="controllers.classfiles.Quiz" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Quiz Result</title>
    <link href="https://img.icons8.com/cute-clipart/64/ask-question.png"
rel="icon">

    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 min-h-screen p-6">
<div class="bg-white p-8 rounded-xl max-w-4xl mx-auto shadow-2xl">
    <h1 class="text-2xl font-bold text-center mb-6">Quiz Results</h1>

    <%
        List<Integer> gotAnswers = (List<Integer>)
request.getAttribute("answers");
        List<Integer> crtAnswers = (List<Integer>)
request.getAttribute("correctAnswers");
        List<Quiz> quizzes = (List<Quiz>) request.getAttribute("quiz");

        int correct = 0;
        for (int i = 0; i < crtAnswers.size(); i++) {
            if (gotAnswers.get(i).equals(crtAnswers.get(i))) {
                correct++;
            }
        }
        int percentage = (correct * 100) / crtAnswers.size();
    %>

    <p class="text-lg text-center text-green-700 mb-2">Score: <%= percentage %>
%</p>
    <p class="text-center text-gray-700">Total Questions: <%= crtAnswers.size()
%></p>
    <p class="text-center text-blue-700 mb-6">Correct: <%= correct %></p>

    <%
        int qIndex = 0;
        for (Quiz quiz : quizzes) {
            for (Question q : quiz.getQuestions()) {
                String[] options = q.getOptions();
                int correctIndex = q.getAnswer();
                int userIndex = gotAnswers.get(qIndex);
            }
        }
    %>
    <div class="mb-4 p-4 border border-gray-300 rounded-lg bg-white shadow-sm">
        <p class="font-semibold mb-2 text-gray-800">
            Q<%= (qIndex + 1) %>: <%= q.getQuestion() %>

```

```

</p>
<ul class="ml-4 space-y-1">
  <%
    for (int i = 0; i < options.length; i++) {
      boolean isCorrect = (i == correctIndex);
      boolean isUser = (i == userIndex);
      String colorClass = isCorrect ? "text-green-600 font-bold" :
        isUser ? "text-red-600 font-semibold" :
"text-gray-700";
      %>
      <li class="<%= colorClass %>">
        <%= (char) ('A' + i) %>. <%= options[i] %>
        <% if (isCorrect && isUser) { %> - Correct
        <% } else if (isUser && !isCorrect) { %> - Your Answer
        <% } else if (isCorrect) { %> - Correct Answer
        <% } %>
      </li>
    <% } %>
  </ul>
</div>
<%
  qIndex++;
}>
}
%>
</div>
</body>
</html>

```

Output :

Welcome to the Quiz App!

Test your knowledge across various topics. Click below to begin!

Start Quiz

Available Quizzes

Java Quiz

Questions: 10

1. What is the size of int in Java?

- ☐ 2 bytes
- ☒ 4 bytes
- ☐ 8 bytes
- ☐ Depends on OS

2. Which keyword is used to inherit a class in Java?

- ☐ this
- ☐ import
- ☒ extends
- ☐ instanceof

3. What is the purpose of the final keyword in Java?

- ☒ To define a constant
- ☐ To inherit a class
- ☐ To start a loop
- ☐ To throw an exception

Quiz Results

Score: 90 %

Total Questions: 10

Correct: 9

Q1: What is the size of int in Java?

- A. 2 bytes
- B. 4 bytes - Correct**
- C. 8 bytes
- D. Depends on OS

Q2: Which keyword is used to inherit a class in Java?

- A. this
- B. import
- C. extends - Correct**
- D. instanceof

Q3: What is the purpose of the final keyword in Java?

- A. To define a constant - Correct**
- B. To inherit a class
- C. To start a loop
- D. To throw an exception