

Set A

1. Write a servlet to display the current date and time on the browser.

```
import java.io.*;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.util.*;
public class DateTimeServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Date date = new Date();
        out.println("<HTML><BODY><h2>Current Date and
Time:</h2>");
        out.println("<BR> <BR> <h1>" + date.toString());
        out.println("</h1></BODY></HTML>");
    }
}
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
    version="4.0">
    <servlet>
        <servlet-name>MyDateTime</servlet-name>
        <servlet-class>DateTimeServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>MyDateTime</servlet-name>
        <url-pattern>/DateTimeServlet</url-pattern>
    </servlet-mapping>
</web-app>
```

- 2. Design a servlet that provides information about a http request from a client, such as IP address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of the currently loaded servlet.**

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.ServletContext;
import javax.servlet.annotation.WebServlet;

@WebServlet("/getInfo")
public class RequestInfoServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Set the content type for the response
        response.setContentType("text/html");

        // Get a writer to output the response body
        PrintWriter out = response.getWriter();

        // Get the ServletContext to access server-specific information
        ServletContext context = getServletContext();

        // --- Information about the Client Request ---
        String clientIP = request.getRemoteAddr();
        // The User-Agent header typically contains the browser and OS information
        String userAgent = request.getHeader("User-Agent");

        // --- Information about the Server ---
        // Get the name of the operating system
        String osName = System.getProperty("os.name");
        // Get the name of the currently loaded servlet (its class name)
        String servletName = this.getClass().getName();
```

```
// --- Generate HTML Output ---
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head><title>Request and Server Information</title></head>");
out.println("<body>");

out.println("<h1>Client Request Information</h1>");
out.println("<table border='1'>");
out.println("<tr><th>Information</th><th>Value</th></tr>");
out.println("<tr><td>Client IP Address</td><td>" + clientIP + "</td></tr>");
out.println("<tr><td>Browser/User Agent Type</td><td>" + userAgent + "</td></tr>");
out.println("</table>");

out.println("<h1>Server Information</h1>");
out.println("<table border='1'>");
out.println("<tr><th>Information</th><th>Value</th></tr>");
out.println("<tr><td>Operating System Type</td><td>" + osName + "</td></tr>");
out.println("<tr><td>Current Servlet Name</td><td>" + servletName + "</td></tr>");
out.println("</table>");

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

out.close();
}

}
```

Set B

1. Create a JSP page to accept a number from the user and display it in words.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Number to Words Converter</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="items-center justify-center min-h-screen">

<%!
    // Arrays for converting digits to words
    private final String[] units = {
        "", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten",
        "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen",
        "Eighteen", "Nineteen"
    };

    private final String[] tens = {
        "", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"
    };
    private String convertLessThanOneThousand(int n) {
        String current;

        if (n % 100 < 20) {
            // Handles 0-19 (e.g., 5, 12, 19)
            current = units[n % 100];
            n /= 100;
        } else {
            // Handles 20-99 (e.g., 25, 99)
            current = units[n % 10];
            n /= 10;

            // Handles 20, 30, ..., 90
            current = tens[n % 10] + (current.isEmpty() ? "" : " ") + current;
            n /= 10;
        }
    }
}
```

```

if (n == 0) return current;

// Handles 100-999
return units[n] + " Hundred" + (current.isEmpty() ? "" : " " + current);
}

// Converts a non-negative integer up to 999,999 to words.
public String convert(long number) {
    if (number == 0) { return "Zero"; }
    if (number < 0) { return "Negative numbers are not supported."; }
    if (number > 999999) { return "Number is too large (max 999,999)."; }

    String words = "";
        // --- 1. Process Thousands ---
    int thousands = (int) (number / 1000);
    if (thousands > 0) {
        words += convertLessThanOneThousand(thousands) + " Thousand";
    }

    // --- 2. Process Hundreds and Units ---
    int remainder = (int) (number % 1000);
    if (remainder > 0) {
        if (!words.isEmpty()) {
            words += " "; // Add space between Thousand and Hundred parts
        }
        words += convertLessThanOneThousand(remainder);
    }

    return words.trim();
}
%>

<div class="bg-white p-6 md:p-10 rounded-xl shadow-2xl w-full max-w-md">
    <h1 class="text-3xl font-bold text-gray-800 mb-6 text-center">
        The Word Machine
    </h1>
    <p class="text-sm text-gray-500 mb-6 text-center">
        Enter a positive whole number (up to 999,999) to see it written out!
    </p>

    <!-- Input Form -->
    <form method="post" action="numberToWords.jsp" class="space-y-4">
        <div>
```

```

<label for="inputNumber" class="block text-sm font-medium text-gray-700
mb-1">Enter Number:</label>
<input type="number"
       id="inputNumber"
       name="inputNumber"
       min="0"
       max="999999"
       placeholder="e.g., 42500"
       required
       class="mt-1 block w-full px-4 py-2 border border-gray-300 rounded-lg shadow-sm
focus:ring-indigo-500 focus:border-indigo-500 transition duration-150 ease-in-out">
</div>
<button type="submit"
        class="w-full flex justify-center py-2 px-4 border border-transparent rounded-lg
shadow-md text-sm font-medium text-white bg-indigo-600 hover:bg-indigo-700
focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition duration-150
ease-in-out">
    Convert to Words
</button>
</form>

<%
// --- JSP Scriptlet (Execution Block) ---
String numberStr = request.getParameter("inputNumber");

if (numberStr != null && !numberStr.trim().isEmpty()) {
    long number = 0;
    String resultWords = "";
    String errorMessage = null;

    try {
        number = Long.parseLong(numberStr.trim());
        if (number < 0 || number > 999999) {
            errorMessage = "Please enter a positive number between 0 and 999,999.";
        } else {
            // Call the Java function defined in the declaration block
            resultWords = convert(number);
        }
    } catch (NumberFormatException e) {
        errorMessage = "That doesn't look like a valid whole number!";
    }
}
%>
<!-- Display Area -->

```

```
<div class="mt-8 p-4 rounded-xl <%= errorMessage != null ? "bg-red-100 border border-red-400" : "bg-green-50 border border-green-300" %>">
  <% if (errorMessage != null) { %>
    <p class="text-lg font-semibold text-red-700"><%= errorMessage %></p>
  <% } else { %>
    <p class="text-xs font-medium text-gray-600">Your Number:</p>
    <p class="text-2xl font-bold text-indigo-700 mb-3"><%= number %></p>
    <p class="text-xs font-medium text-gray-600">In Words:</p>
    <p class="text-xl font-bold text-green-800"><%= resultWords %></p>
  <% } %>
</div>
<%
  }
%>
</div>
</body>
</html>
```

2. Create a jsp page, which accepts user name in a text box and greets the user according to the time on server side

Input.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>User Greeting</title>
    </head>
    <body>
        <h1>Enter Your Name</h1>
        <form action="greeting.jsp" method="post">
            Enter your name: <input type="text" name="userName"><br><br>
            <input type="submit" value="Greet Me">
        </form>
    </body>
</html>
```

Greeting.jsp

```
<%@page import="java.util.Calendar"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Greeting</title>
    </head>
    <body>
        <%
```

```
// Retrieve the username from the request parameter
String userName = request.getParameter("userName");

// Get the current time on the server
Calendar rightNow = Calendar.getInstance();
int hour = rightNow.get(Calendar.HOUR_OF_DAY);

// Determine the appropriate greeting based on the hour
String greeting;
if (hour >= 5 && hour < 12) {
    greeting = "Good Morning";
} else if (hour >= 12 && hour < 17) {
    greeting = "Good Afternoon";
} else if (hour >= 17 && hour < 22) {
    greeting = "Good Evening";
} else {
    greeting = "Good Night";
}
%>

<h1>
    <%= greeting %>, <%= userName %>!
</h1>
</body>
</html>
```

Set B.

Q 1. Create a JSP page to accept a number from the user and display it in words.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Number to Words Converter</title>
</head>
<body>

<div class="container">
<h1>Number to Words</h1>

<form method="get" action="numberToWords.jsp">
<label for="inputNumber">Enter a Number (0 to 999):</label>
<input type="number" id="inputNumber" name="number" required min="0"
max="999" step="1">
<input type="submit" value="Convert to Words">
</form>

<%
// --- JSP Scriptlet Logic ---
String numberStr = request.getParameter("number");

if (numberStr != null && !numberStr.trim().isEmpty()) {
try {
// Parse the number
int number = Integer.parseInt(numberStr);
String words = "";

```

```
String errorMessage = null;
// Check for reasonable limits for this simple implementation
if (number < 0) {
    errorMessage = "Please enter a non-negative number.";
} else if (number > 999) {
    errorMessage = "Number is too large for this converter (Max: 999).";
} else if (number == 0) {
    words = "Zero";
} else {
    // Call the conversion function
    words = convertBelowThousand(number);
}

out.println("<div class='result'>");
if (errorMessage != null) {
    out.println("<p class='error'>Error: " + errorMessage + "</p>");
} else {
    out.println("<h3>Result:</h3>");
    out.println("<p>The number <b>" + numberStr + "</b> in words is:</p>");
    out.println("<p><b>" + words + "</b></p>");
}
out.println("</div>");

} catch (NumberFormatException e) {
    out.println("<p class='error'>Error: Invalid number format entered.</p>");
}
}

%>
</div>
</body>
</html>
```

```

<%!
// --- JSP Declaration: Java Logic for Conversion ---
// This section contains the actual Java methods, executed only once per JSP
compilation.

private final String[] units = {
    "", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten",
    "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen",
    "Eighteen", "Nineteen"
};

private final String[] tens = {
    "",      // 0
    "",      // 1 (handled by units for 10-19)
    "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"
}

// * Converts a number (0 to 999) into words.
private String convertBelowThousand(int number) {
    String current;

    if (number % 100 < 20){
        current = units[number % 100];
        number /= 100;
    } else {
        current = units[number % 10];
        number /= 10;
        if (number % 10 > 0) {
            current = tens[number % 10] + " " + current;
        }
        number /= 10;
    }
}

```

```
}

if (number == 0) return current.trim();

if (!current.isEmpty()) {

    return units[number] + " Hundred and " + current.trim();

} else {

    return units[number] + " Hundred";

}

%>
```

2. Create a JSP page, which accepts user name in a text box and greets the user according to the time on server side.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.util.Calendar" %>
<%@ page import="java.util.Date" %>
<!DOCTYPE html>
<html>
<head> </head>
<body>
<p>
<%
// =====
// J S P   S C R I P T L E T   B L O C K (<% )
// This runs when the page is requested.
// =====

// 1. Get the username from the request parameters
String userName = request.getParameter("userName");

// Use a default name if the user hasn't submitted the form yet
if (userName == null || userName.trim().isEmpty()) {
    userName = "Guest";
}

// 2. Determine the time and greeting
String greetingMessage = "Hello";

// Get the current server time
Calendar cal = Calendar.getInstance();
int hourOfDay = cal.get(Calendar.HOUR_OF_DAY); // 0 (midnight) to 23 (11 PM)

// Logic for time-based greeting
if (hourOfDay >= 5 && hourOfDay < 12) {
    greetingMessage = "Good Morning"; // 5 AM to 11:59 AM
} else if (hourOfDay >= 12 && hourOfDay < 17) {
    greetingMessage = "Good Afternoon"; // 12 PM to 4:59 PM
} else if (hourOfDay >= 17 && hourOfDay < 22) {
    greetingMessage = "Good Evening"; // 5 PM to 9:59 PM
} else {
```

```
greetingMessage = "Hello (It's late)"; // 10 PM to 4:59 AM
}

// Display the full date/time for context
String serverTime = new Date().toString();
%>

<!-- HTML / Presentation Layer -->
<div>
<!-- Input Form -->
<form method="GET">
<div>
    <label for="userName">
        What is your name?
    </label>
    <input type="text" id="userName" name="userName" value="<%=
"Guest".equals(userName) ? "" : userName %>" >
</div>

    <button type="submit"> Greet Me </button>
</form>

<!-- Greeting Result Display -->
<p>
    <!-- J S P   E X P R E S S I O N   B L O C K   -->
    <%= greetingMessage %>, <%= userName %>!
</p>
</div>
</body>
</html>
```