frontend practical

Q2 - Using HTML Tags, design your own curriculum vitae.

```
#CODE
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Shubham Yadav - Curriculum Vitae</title>
 <style>
 body {
 font-family: Arial, sans-serif;
 margin: 0;
 padding: 0;
 background-color: #f9f9f9;
 }
 header {
 background-color: #333;
 color: #fff;
 padding: 20px;
 text-align: center;
 }
 section {
 margin: 20px;
 }
 h1, h2 {
 color: #333;
 }
 p {
 margin-bottom: 10px;
 }
 </style>
</head>
<body>
 <header>
 <p>SHUBHAM YADAV</p>
 <p>MCA (SE) - 2nd Semester</p>
 </header>
 <section>
 <h2>Personal Information</h2>
 <p>Name: Shubham Yadav</p>
 <p>Education: Master of Computer Applications (MCA)</p>
 <p>Semester: 2nd Semester</p>
 <p>Interest: Fullstack web Development</p>
 </section>
 <section>
 <h2>Education</h2>
 <p>MCA (SE),USICT, 2023 - Present</p>
 </section>
 <section>
 <h2>Skills</h2>
 <ul>
 <li>HTML , CSS , Javascript </li>
 <li>Reactjs</li>
 <li>Expressjs</li>
 <li>Problem-solving</li>
 </ul>
 </section>
 <section>
 <h2>Projects</h2>
 <p>Project Name: Online food delivery platform </p>
 <p>Description: Platform that makes life easier for foodies </p>
 </section>
 <section>
 <h2>Contact Information</h2>
 <p>Email: shub79908@gmail.com</p>
 <p>Phone: 9205537706</p>
 </section>
</body>
</html>
```

Q3

Write a XML Program to display Student profile having age like student roll no.,name age, semester, email id, phone number, department name, apply and validate using DTD.

```
<!ELEMENT student_profile (rollNo, name, age, semester, email, phoneNumber, department)>
<!ELEMENT rollNo (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT semester (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>
<!ELEMENT department (#PCDATA)>
<!DOCTYPE student_profile SYSTEM>
<student_profile>
<rollNo>00716404523</rollNo>
<name>Shubham Yadav</name>
<age>22</age>
<semester>2</semester>
<email>shub79908@gmail.com</email>
<phoneNumber>9205537706</phoneNumber>
<department>MCA</department>
</student_profile>


//ANOTHER APPROACH

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE studentProfile SYSTEM "student_profile.dtd">
<studentProfile>
    <student>
        <rollNo>101</rollNo>
        <name>John Doe</name>
        <age>21</age>
        <semester>4</semester>
        <email>john.doe@example.com</email>
        <phone>123-456-7890</phone>
        <department>Computer Science</department>
    </student>
    <student>
        <rollNo>102</rollNo>
        <name>Jane Smith</name>
        <age>22</age>
        <semester>6</semester>
        <email>jane.smith@example.com</email>
        <phone>987-654-3210</phone>
        <department>Electrical Engineering</department>
    </student>
</studentProfile>


//DTD FILE

<!ELEMENT studentProfile (student+)>
<!ELEMENT student (rollNo, name, age, semester, email, phone, department)>
<!ELEMENT rollNo (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT semester (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT department (#PCDATA)>
```

Q4

Write a javascript Program to check weather an input number is palindrome number or not.

```
function checkPalindrome(number) {
    // Convert the number to a string
    let strNumber = number.toString().trim();

    // Check if the input is a valid number
    if (isNaN(strNumber) || strNumber === "") {
        return "Please enter a valid number.";
    }

    // Reverse the string
    let reversedStr = strNumber.split('').reverse().join('');

    // Check if the original string and the reversed string are the same
    if (strNumber === reversedStr) {
        return number + " is a palindrome number.";
    } else {
        return number + " is not a palindrome number.";
    }
}

// Example usage:
console.log(checkPalindrome(121)); // Output: 121 is a palindrome number.
console.log(checkPalindrome(123)); // Output: 123 is not a palindrome number.
console.log(checkPalindrome("  1221  ")); // Output: 1221 is a palindrome number.
console.log(checkPalindrome("abc")); // Output: Please enter a valid number.
```

Q5   Write a JSP Program to auto refresh a page

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Auto Refresh Page</title>
    <!-- Set the page to refresh every 5 seconds -->
    <meta http-equiv="refresh" content="5">
</head>
<body>

<h2>Auto Refreshing Page</h2>
<p>The page will refresh automatically every 5 seconds.</p>

<%
    // Get the current date and time
    java.util.Date currentDate = new java.util.Date();
%>
<p>Current Date and Time: <%= currentDate.toString() %></p>

</body>
</html>
```

Q6 Write a JSP Program to upload file into server.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>File Upload</title>
</head>
<body>
    <h2>Upload File</h2>
    <form method="post" action="FileUploadServlet" enctype="multipart/form-data">
        <input type="file" name="file" /><br/><br/>
        <input type="submit" value="Upload" />
    </form>
</body>
</html>


//JAVA

import java.io.File;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;

@WebServlet("/FileUploadServlet")
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 2, // 2MB
                 maxFileSize = 1024 * 1024 * 10,      // 10MB
                 maxRequestSize = 1024 * 1024 * 50)   // 50MB
public class FileUploadServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throw
```

```
        // Get the file part from the request
        Part filePart = request.getPart("file");

        // Get the file name
        String fileName = getSubmittedFileName(filePart);

        // Define the directory where the file will be stored
        String uploadDir = getServletContext().getRealPath("") + File.separator + "uploa

        // Create the directory if it doesn't exist
        File directory = new File(uploadDir);
        if (!directory.exists()) {
            directory.mkdirs();
        }

        // Write the file to the specified directory
        String filePath = uploadDir + File.separator + fileName;
        filePart.write(filePath);

        // Redirect back to the upload page with a success message
        response.sendRedirect("upload.jsp?message=File uploaded successfully");
    }

    // Utility method to extract file name from HTTP header content-disposition
    private String getSubmittedFileName(Part part) {
        String header = part.getHeader("content-disposition");
        for (String headerPart : header.split(";")) {
            if (headerPart.trim().startsWith("filename")) {
                return headerPart.substring(headerPart.indexOf('=') + 1).trim()
                        .replace("\"", "");
            }
        }
        return null;
    }
}
```

Q7

Write a Generic Servlet program to display your own Enrollment number and Name using Apache Tomcat server.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyInfoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws S
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>My Info</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>My Enrollment Number and Name</h1>");
        out.println("<p>Enrollment Number: Your_Enrollment_Number</p>");
        out.println("<p>Name: Your_Name</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Q8

Write a HTTP Servlet program to display all the HTTP Request Header parameters.

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/DisplayHeaders")
public class DisplayHeaders extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>HTTP Request Headers</title></head><body>");
        out.println("<h1>HTTP Request Headers:</h1>");
        out.println("<ul>");

        Enumeration<String> headerNames = request.getHeaderNames();
        while (headerNames.hasMoreElements()) {
            String headerName = headerNames.nextElement();
            String headerValue = request.getHeader(headerName);
            out.println("<li><strong>" + headerName + ":</strong> " + headerValue + "</li
        }

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

Q9 Write a HTTP Servlet program to create a Cookie.

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

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

@WebServlet("/CreateCookieServlet")
public class CreateCookieServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        // Create a new cookie
        Cookie cookie = new Cookie("username", "JohnDoe");

        // Set the maximum age of the cookie to 24 hours (86400 seconds)
        cookie.setMaxAge(24 * 60 * 60);

        // Add the cookie to the response
        response.addCookie(cookie);

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

        // Write the response
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h2>Cookie Created Successfully</h2>");
        out.println("</body></html>");
    }
}
```

Q 10

Write a JDBC Program to fetch the employees records from the Employee table designed in MS Access. The Table should have fields like : Employee ID, Name, DOB, Address, Department, DOJ, Position etc.

```
import java.sql.*;

public class FetchEmployeeRecords {
    // JDBC URL for MS Access database
    static final String JDBC_URL = "jdbc:ucanaccess://path_to_your_database.accdb";

    public static void main(String[] args) {
        try {
            // Establishing a connection to the database
            Connection connection = DriverManager.getConnection(JDBC_URL);

            // Creating a SQL statement
            Statement statement = connection.createStatement();

            // SQL query to fetch employee records
            String sqlQuery = "SELECT * FROM Employee";

            // Executing the query
            ResultSet resultSet = statement.executeQuery(sqlQuery);

            // Displaying the employee records
            while (resultSet.next()) {
                int employeeID = resultSet.getInt("EmployeeID");
                String name = resultSet.getString("Name");
                Date dob = resultSet.getDate("DOB");
                String address = resultSet.getString("Address");
                String department = resultSet.getString("Department");
                Date doj = resultSet.getDate("DOJ");
                String position = resultSet.getString("Position");
                // Displaying the employee information
                System.out.println("Employee ID: " + employeeID);
                System.out.println("Name: " + name);
                System.out.println("Date of Birth: " + dob);
                System.out.println("Address: " + address);
                System.out.println("Department: " + department);
                System.out.println("Date of Joining: " + doj);
                System.out.println("Position: " + position);
                System.out.println("-----------------------------");
            }

            // Closing the resources
            resultSet.close();
            statement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```