**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY**

**UNIVERSITY SCHOOL OF INFORMATION, COMMUNICATION & TECHNOLOGY**

**SECTOR-16C, DWARKA, NEW DELHI**



## PRACTICAL FILE

## (IT - 664)

# FRONT END DESIGN TECHNIQUES LAB

|  |  |  |
| --- | --- | --- |
| **­­­­SUBMITTED TO** |  | **SUBMITTED BY** |
|  |  |  |
| **Mr. NITENDRA SINGH**  **(Research Scholar, USICT)** |  | **Mr. RAJ KHATRI**  **MCA (Software Engineering)**  Enrolment No.: - **04316404523** |

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Program | Date | Signature |
| **1** | What is Frontend? Write a note on HTML, CSS and JavaScript |  |  |
| **2** | Using HTML Tags, design your own curriculum vitae |  |  |
| **3** | Using HTML Tags, redesign the home page of GGSIP university |  |  |
| **4** | Write an 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 |  |  |
| **5** | Write a JavaScript Program to check whether an input number is palindrome number or not |  |  |
| **6** | Write a JSP Program to auto refresh a page. |  |  |
| **7** | Write a JSP Program to upload file into server. |  |  |
| **8** | Write a Generic Servlet program to display your own Enrolment number and Name using Apache Tomcat server. |  |  |
| **9** | Write a HTTP Servlet program to display all the HTTP Request Header parameters. |  |  |
| **10** | Write a HTTP Servlet program to create a Cookie. |  |  |
| **11** | 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 |  |  |
| **12** | Create your portfolio website using LocalWP and WordPress |  |  |

**Practical – 01**

**Aim: What is Frontend? Write a note on HTML, CSS and JavaScript**

The Frontend of a website refers to the part users directly interact with, encompassing its visual elements, layout, and interactive functionality. It’s the user interface (UI), the first impression that grabs attention and determines a website usability and accessibility. Think of it as the store front of a store, exciting customers and facilitating their browsing experience.

**HTML**

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

An HTML element is defined by a start tag, some content, and an end tag:

**<tag name>** Content goes here... **</tag name>**

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>My First Heading</h1>

<p>My first paragraph. </p>

</body>

</html>

**Example Explained**

* The <!DOCTYPE html> declaration defines that this document is an HTML5 document
* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading
* The <p> element defines a paragraph

**HTML Headings**

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important

**Example**

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

**HTML Paragraphs**

HTML paragraphs are defined with the <p> tag.

**Example**

<p>This is a paragraph. </p>

<p>This is another paragraph. </p>

**HTML Button**

HTML buttons are created with the <button> tag.

**Example**

<button>Click me</button>

**HTML Links**

HTML links are defined with the <a> tag:

**Example**

<a href="https://www.usictipu.com">This is a link</a>

* The link's destination is specified in the href attribute.
* Attributes are used to provide additional information about HTML elements.
* You will learn more about attributes in a later chapter.

**HTML Images**

HTML images are defined with the <img> tag.

The source file (src), alternative text (alt), width, and height are provided as attributes:

**Example**

<img src="images/random.jpg" alt="image " width="164" height="142">

**HTML List**

HTML lists can be unordered (<ul>), ordered (<ol>), or definition lists (<dl>). Here's an example of an unordered list:

**Example**

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

**CSS**

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files

**CSS Example**

body {

background-color: lightblue;

}

h1 {

color: white;

text-align: center;

}

p {

font-family: verdana;

font-size: 20px;

}

**CSS Syntax**

H1{color:blue; font-size:12px}

* The selector points to the HTML element you want to style.
* The declaration block contains one or more declarations separated by semicolons.
* Each declaration includes a CSS property name and a value, separated by a colon.
* Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

**Three Ways to Insert CSS**

There are three ways of inserting a style sheet:

* External CSS
* Internal CSS
* Inline CSS

**JAVASCRIPT**

* JavaScript is a programming language that enables you to create dynamically updating content, control multimedia, animate images, and much more.
* JavaScript runs on the client side of the web, which enables interactions with the user and dynamic content.
* JavaScript is an essential part of web development, used to enhance user interfaces and add functionality to web pages.

**JavaScript Comments**

* Single-line comments start with `//` and are used for short comments.

**Example**

// This is a single-line comment

* Multi-line comments start with `/\*` and end with `\*/` and are used for longer comments.

**Example**

/\* This is a

multi-line comment \*/

**JavaScript Variables**

* Variables are containers for storing data values.
* Use the `var`, `let`, or `const` keyword to declare variables.

**Example**

var x = 5; // Declaring a variable with the value of 5

var y = 6; // Declaring another variable with the value of 6

var z = x + y; // Adding x and y, and assigning the result to z

**JavaScript Functions:**

* Functions are blocks of code that perform a specific task.
* They are defined using the `function` keyword, followed by a name and parentheses containing parameters.

**Example**

function myFunction(p1, p2) { // Defining a function called myFunction with two parameters

return p1 \* p2; // Returning the product of p1 and p2

}

**JavaScript Events:**

* Events are actions that occur on web pages, such as when a user clicks a button.
* You can use JavaScript to respond to these events and perform actions.

**Example**

<button onclick="myFunction()">Click me</button>// Calling myFunction when the button is clicked

**JavaScript Output:**

* You can change the content of HTML elements using JavaScript with the innerHTML property.

**Example**

document.getElementById("demo").innerHTML = "Hello JavaScript!";

// Changing the content of an HTML element with the id "demo"

**Practical – 02**

**Aim - Using HTML Tags, design your own curriculum vitae**

**Program Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Resume</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

box-sizing: border-box;

background-color: #f8f8f8; /\* Light gray background color \*/

background-size: cover;

background-position: center;

color: #333;

}

.container {

max-width: 800px;

margin: 20px auto;

background-color: #fff; /\* White background for content \*/

padding: 20px;

border-radius: 10px; /\* Rounded corners \*/

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Shadow for depth \*/

display: grid;

grid-template-columns: 1fr 1fr;

grid-gap: 20px;

}

.top {

grid-column: span 2; /\* Span across both columns \*/

text-align: center;

margin-bottom: 20px;

}

.top .heading1 {

font-size: 32px;

margin-bottom: 10px;

color: #333;

}

.top .para {

margin: 0;

font-size: 16px;

}

.section {

margin-bottom: 20px;

}

.section h2 {

font-size: 24px;

margin-bottom: 10px;

color: #333;

}

.section p {

margin: 0;

font-size: 16px;

}

@media print {

body {

margin: 0;

padding: 0;

background-color: #fff;

color: #333;

}

.container {

max-width: 800px;

margin: 20px auto;

display: grid;

grid-template-columns: 1fr 1fr;

grid-gap: 20px;

}

.section {

margin-bottom: 20px;

}

.section h2 {

font-size: 24px;

margin-bottom: 10px;

color: #333;

}

.section p {

margin: 0;

font-size: 16px;

}

}

</style>

</head>

<body>

<div class="container">

<div class="top">

<h1 class="heading1">Raj Khatri</h1>

<p class="para">

khatriraj030@gmail.com | +91 9650607524 | New Delhi, India

</p>

</div>

<div class="section">

<h2>Education</h2>

<p>

B.Sc. (Hons.) Computer Science,<br />

University of Delhi, <br />

Graduated in 2023 <br />

CGPA: 8.1

</p>

<br />

<p>

MCA (Software Engineering)<br />

University of Delhi, <br />

Graduated in 2023 <br />

CGPA: 8.9

</p>

</div>

<div class="section">

<h2>Projects</h2>

<p>

1. E-Commerce Website - Designed and developed a fully functional

e-commerce platform using MERN stack.

</p>

<br />

<p>2. JavaMovie - Built a android app for learning c++ java.</p>

<br />

<p>

3. FoMo Movie Ticketing - Designed UI/UX of a movie ticketing app on

Figma.

</p>

</div>

<div class="section">

<h2>Tech Stack</h2>

<p>JavaScript, React, Node.js, Express, MongoDB</p>

</div>

<div class="section">

<h2>Languages</h2>

<p>Hindi (Native), English (Proficient)</p>

</div>

<div class="section">

<h2>Coursework</h2>

<p>

Data Structures and Algorithms, Web Development, Database Management Systems

</p>

</div>

<div class="section">

<h2>Certifications</h2>

<p>1. Meta Frontend Development - Coursera</p>

<p>2. Full Stack Development Bootcamp - Udemy</p>

<p>3. Google UX Design Specialization - Coursera</p>

</div>

<div class="section">

<h2>Portfolio Links</h2>

<p>GitHub: github.com/khatrijiraj</p>

<p>Leetcode: leetcode.com/khatrijiraj</p>

<p>Behance: behance.net/khatrijiraj</p>

<p>Twitter: twitter.com/khatrijiraj</p>

</div>

<div class="section">

<h2>Experience</h2>

<p>Software Engineer at XYZ Tech (2020 - Present)</p>

<p>

Responsible for designing and implementing scalable web applications

</p>

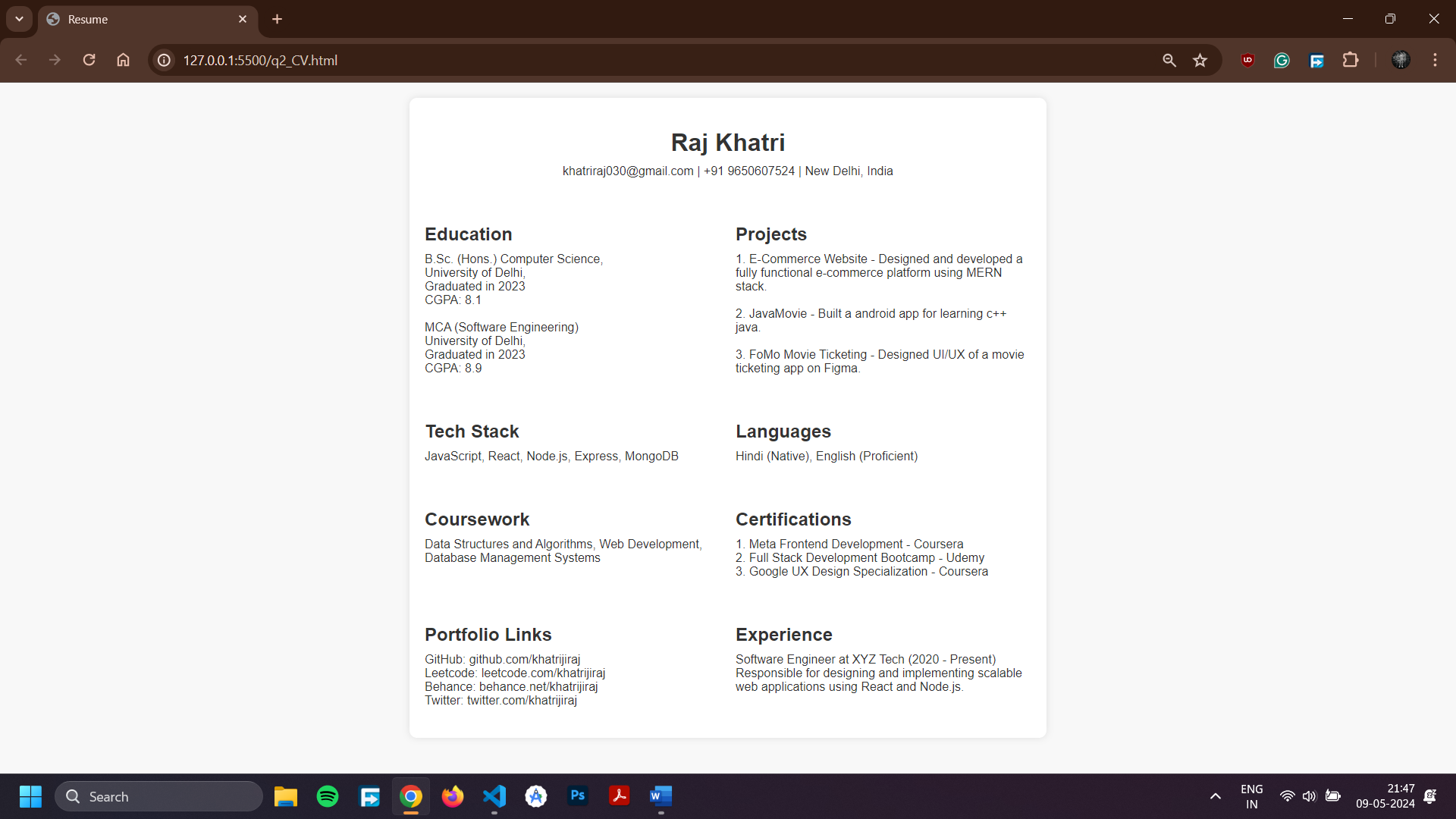
</div>

</div>

</body>

</html>

**Output**



**Practical - 03**

**Aim: Using HTML Tags, redesign the home page of GGSIP university**

**Program Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>GGSIP University</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

box-sizing: border-box;

}

header {

background-color: #005cb6;

color: #fff;

padding: 10px 20px;

text-align: center;

}

nav {

background-color: #f8f9fa;

padding: 10px 20px;

text-align: center;

}

nav a {

margin: 0 10px;

color: #005cb6;

text-decoration: none;

font-weight: bold;

}

nav a:hover {

text-decoration: underline;

}

main {

padding: 20px;

}

footer {

background-color: #005cb6;

color: #fff;

padding: 10px 20px;

text-align: center;

}

.section {

margin-bottom: 20px;

}

.section h2 {

color: #005cb6;

font-size: 24px;

margin-bottom: 10px;

}

.section p {

font-size: 16px;

line-height: 1.5;

}

.section img {

max-width: 100%;

height: auto;

}

</style>

</head>

<body>

<header>

<h1>Guru Gobind Singh Indraprastha University</h1>

<p>Knowledge is Supreme Goal</p>

</header>

<nav>

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Academics</a>

<a href="#">Admissions</a>

<a href="#">Research</a>

<a href="#">Campus Life</a>

<a href="#">Alumni</a>

<a href="#">Contact</a>

</nav>

<main>

<section class="section">

<h2>Welcome to GGSIP University</h2>

<p>

Guru Gobind Singh Indraprastha University (GGSIPU or IP or IPU) is a

state university located in Delhi, India. It was established in 1998

by the Government of NCT of Delhi. The University is recognized by the

University Grants Commission (UGC), India under section 12B of UGC

</p>

<img

style="height: 300px"

src="http://ipu.ac.in/data0/images/1.jpg"

alt="University Campus" />

</section>

<section class="section">

<h2>Latest News</h2>

<ul>

<li><a href="#">Admission Notice for B.Tech Programs - 2022</a></li>

<li>

<a href="#">Online Registration for Ph.D. Entrance Exam - 2022</a>

</li>

<li>

<a href="#">Seminar on Emerging Technologies in Computer </a>

</li></ul>

</section>

<section class="section">

<h2>Academic Programs</h2>

<ul>

<li>B.Tech in Computer Science</li>

<li>BA (Hons) in Economics</li>

<li>Master of Business Administration (MBA)</li>

<li>Ph.D. in Biotechnology</li>

</ul>

</section>

<section class="section">

<h2>Events</h2>

<ul>

<li><a href="#">Tech Fest 2022 - Register Now!</a></li>

<li><a href="#">Cultural Night - Celebrating Diversity</a></li>

</ul>

</section>

</main>

<footer>

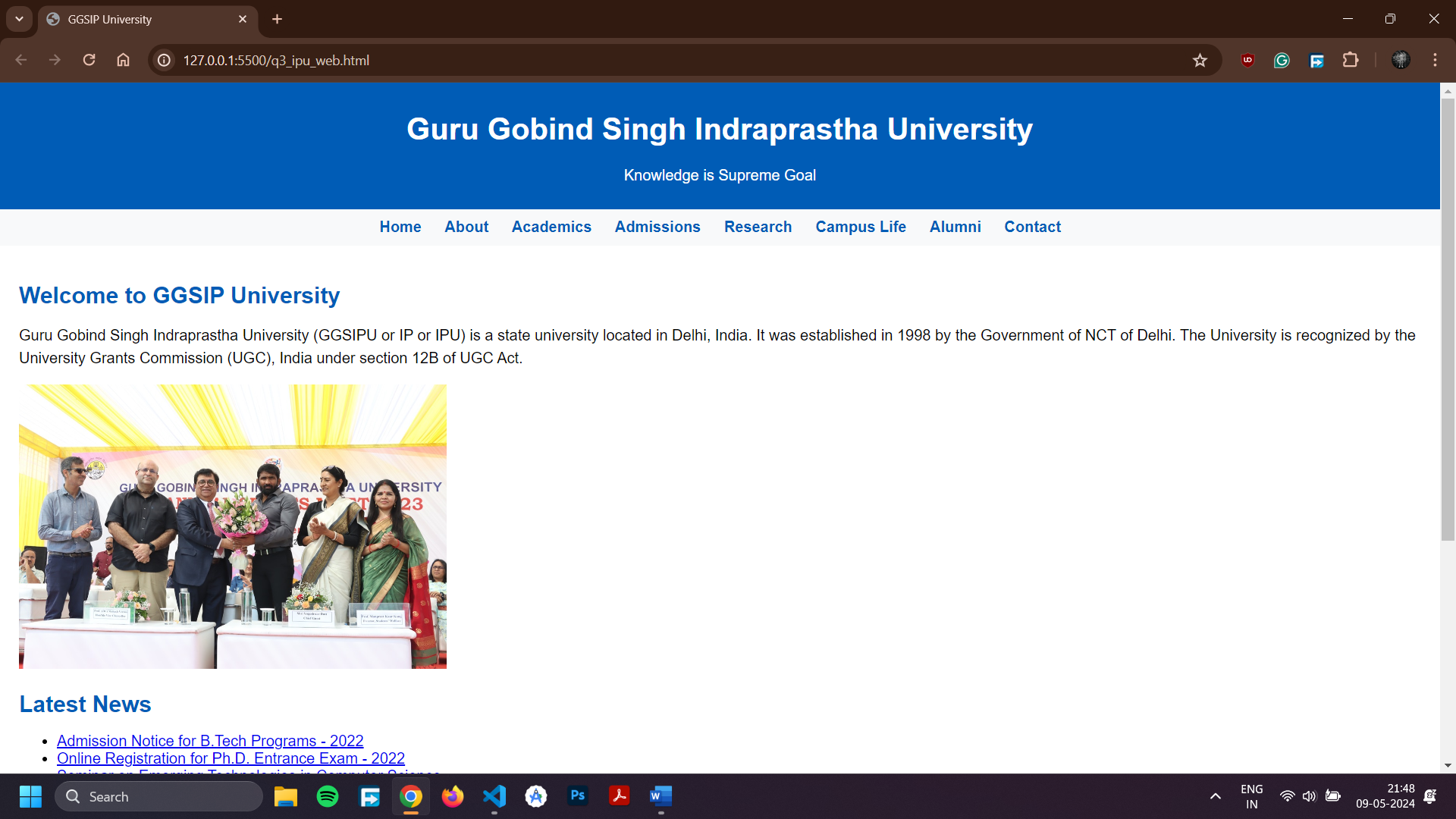
<p>&copy; 2022 GGSIP University. All rights reserved.</p>

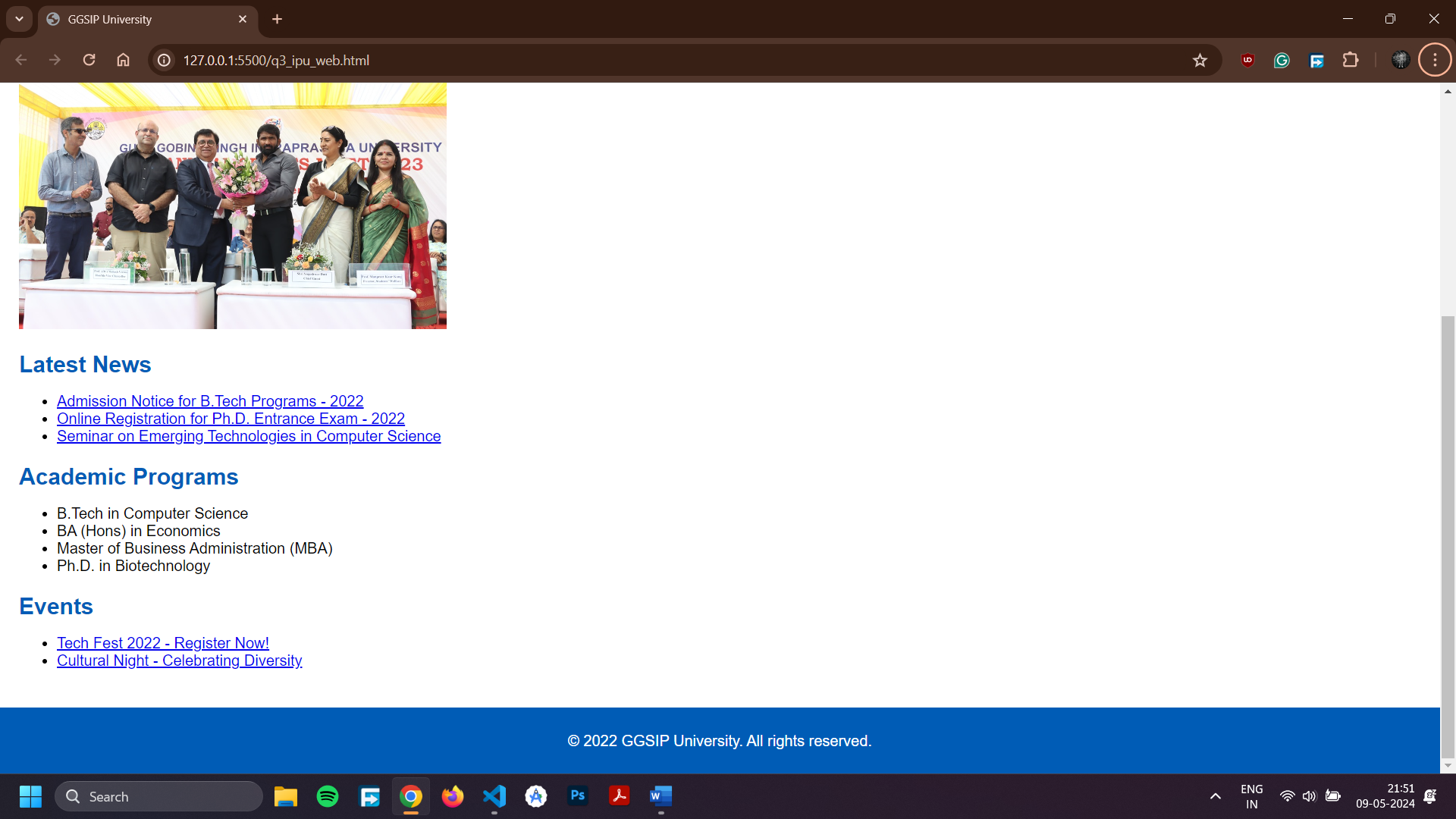
</footer>

</body>

</html>

**Output**





**Practical - 04**

**Aim: Write an 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**

**Program Code**

<!-- student\_profile.xml -->

<!DOCTYPE student [

<!ELEMENT student (roll, name, age, semester, email, phone, department, apply)>

<!ELEMENT roll (#PCDATA)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT age (#PCDATA)>

<!ELEMENT semester (#PCDATA)>

<!ELEMENT email (#PCDATA)>

<!ELEMENT phone (#PCDATA)>

<!ELEMENT department (#PCDATA)>

<!ELEMENT apply (#PCDATA)>

]>

<student>

<roll>1</roll>

<name>John Doe</name>

<age>20</age>

<semester>3</semester>

<email>john.doe@example.com</email>

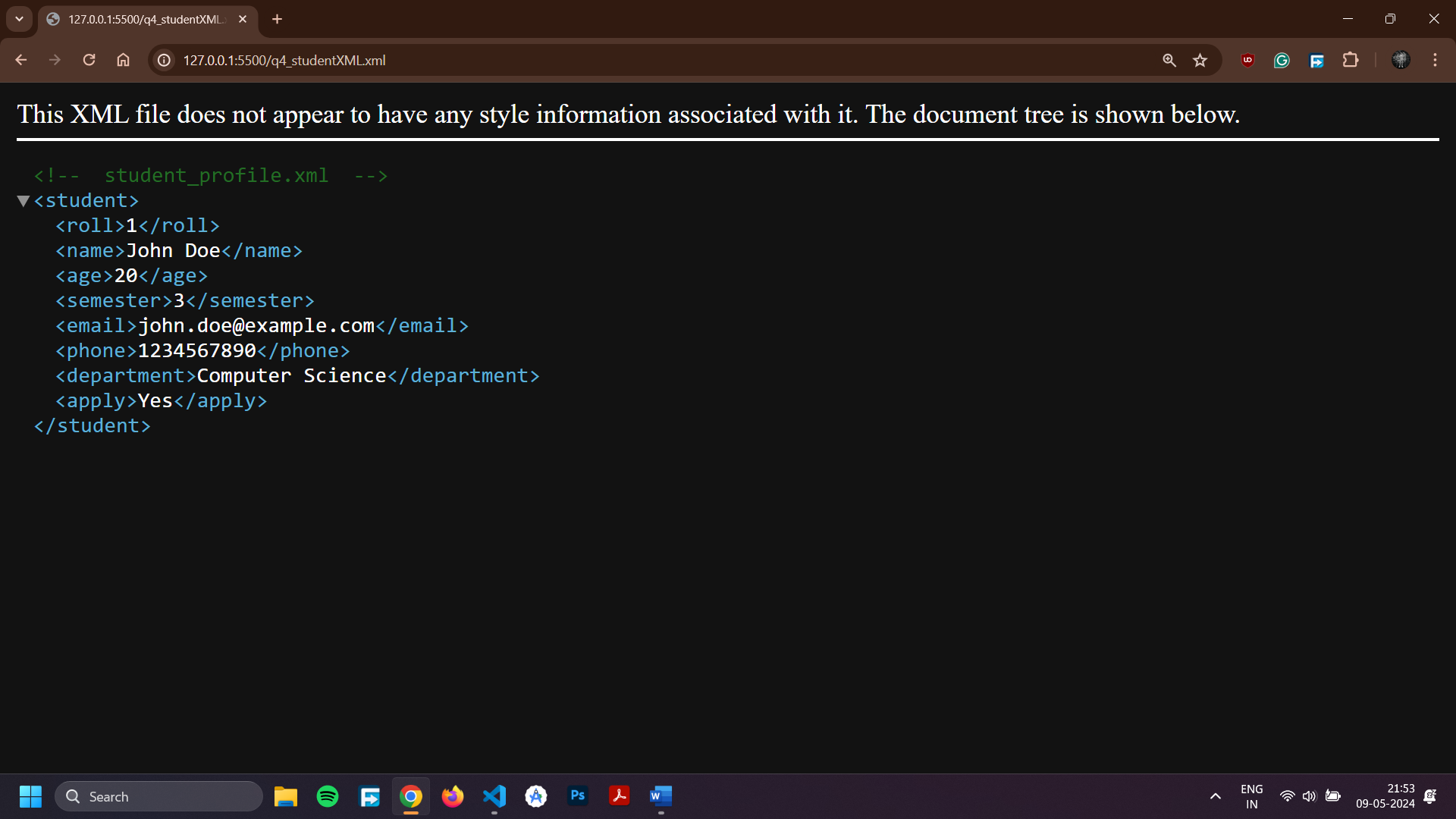
<phone>1234567890</phone>

<department>Computer Science</department>

<apply>Yes</apply>

</student>

**Output**



**Practical - 05**

**Aim: Write a JavaScript Program to check whether an input number is palindrome number or not**

**Program Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Palindrome Checker</title>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

}

input[type="text"],

input[type="submit"] {

margin: 10px;

padding: 5px;

font-size: 16px;

}

</style>

</head>

<body>

<h1>Palindrome Checker</h1>

<form id="palindromeForm">

<label for="number">Enter a number:</label>

<input type="text" id="number" name="number" />

<input type="submit" value="Check" />

</form>

<div id="result"></div>

<script>

function isPalindrome(number) {

const strNumber = String(number);

const reversedNumber = strNumber.split("").reverse().join("");

return strNumber === reversedNumber;

}

document.getElementById("palindromeForm")

.addEventListener("submit", function (event) {

event.preventDefault();

const number = document.getElementById("number").value;

if (!isNaN(number) && number !== "") {

if (isPalindrome(number)) {

document.getElementById(

"result"

).innerText = `${number} is a palindrome number.`;

} else {

document.getElementById(

"result"

).innerText = `${number} is not a palindrome number.`;

}

} else {

document.getElementById("result").innerText =

"Please enter a valid number.";

}

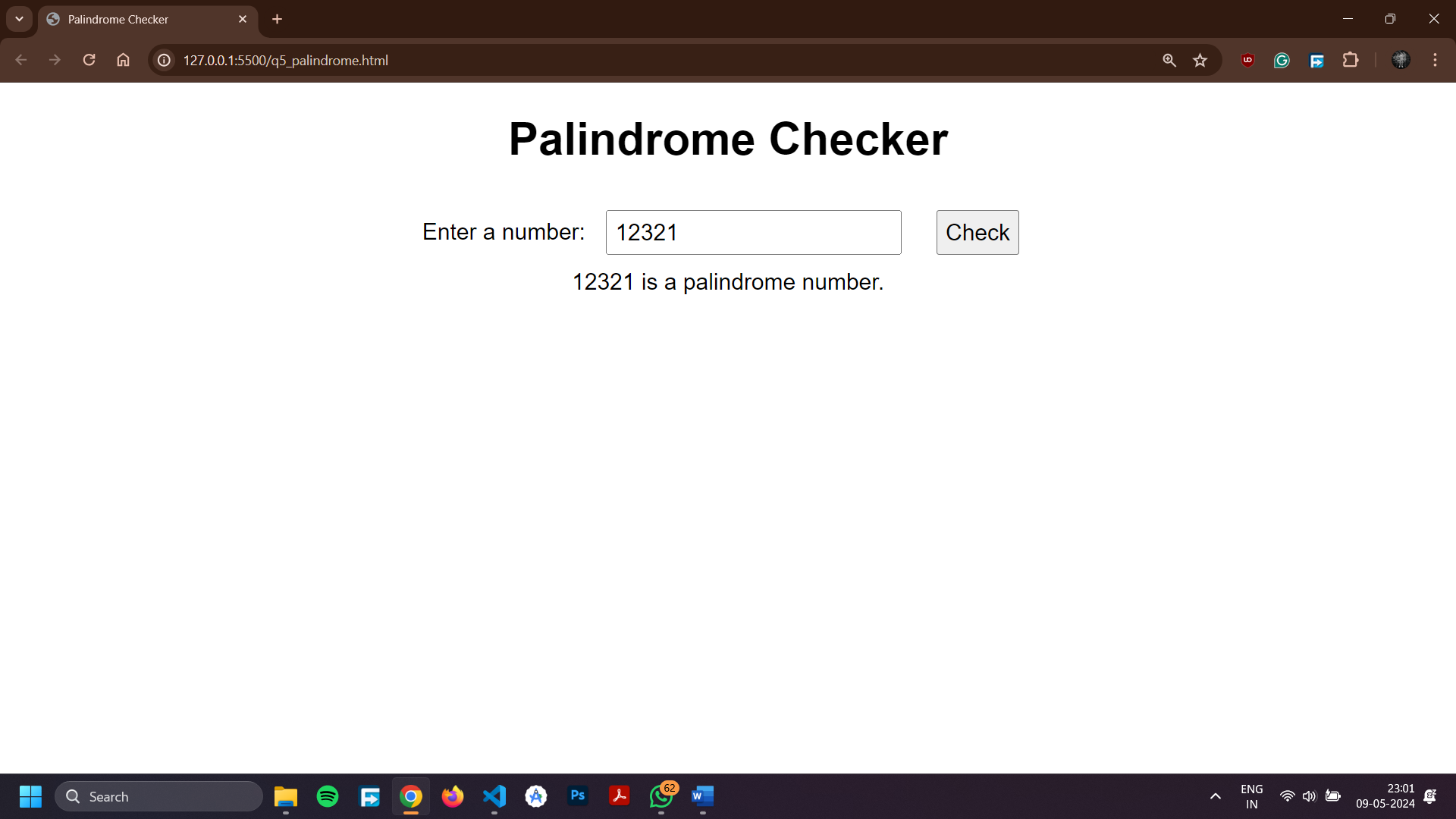
});

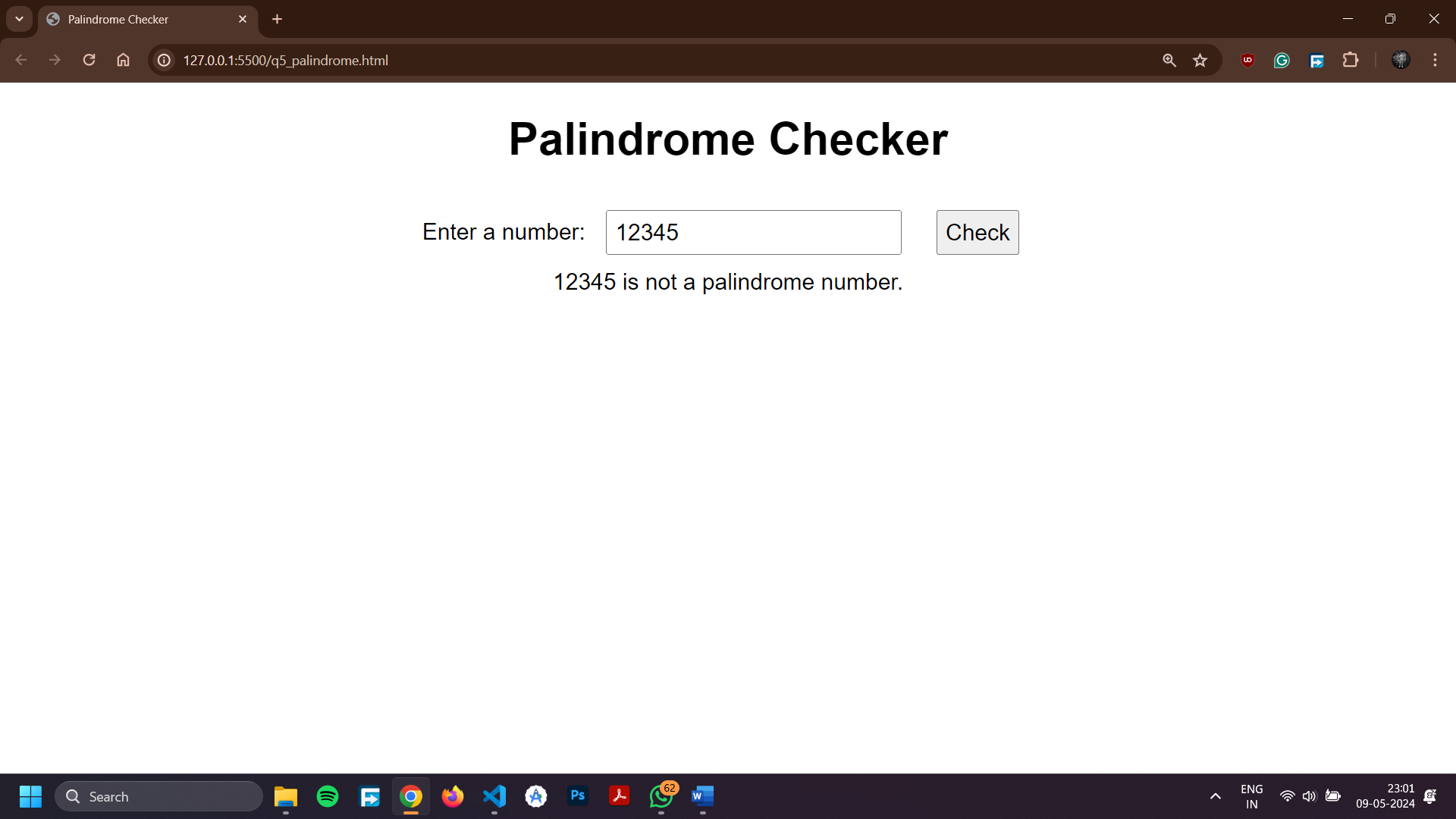
</script>

</body>

</html>

**Output**





**Practical – 06**

**Aim: Write a JSP Program to auto refresh a page.**

**Program Code**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta http-equiv="refresh" content="5" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Auto Refresh Page</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

text-align: center;

padding: 20px;

border-radius: 10px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

color: #333;

}

p {

color: #666;

}

.countdown {

font-size: 2em;

margin-top: 20px;

color: #333;

}

</style>

<script>

let countdown = 5;

setInterval(() => {

countdown--;

document.getElementById("countdown").innerText = countdown;

}, 1000);

</script>

</head>

<body>

<div class="container">

<h1>Welcome to the Auto-Refreshing Page!</h1>

<p>

This page will automatically refresh in

<span id="countdown">5</span> seconds.

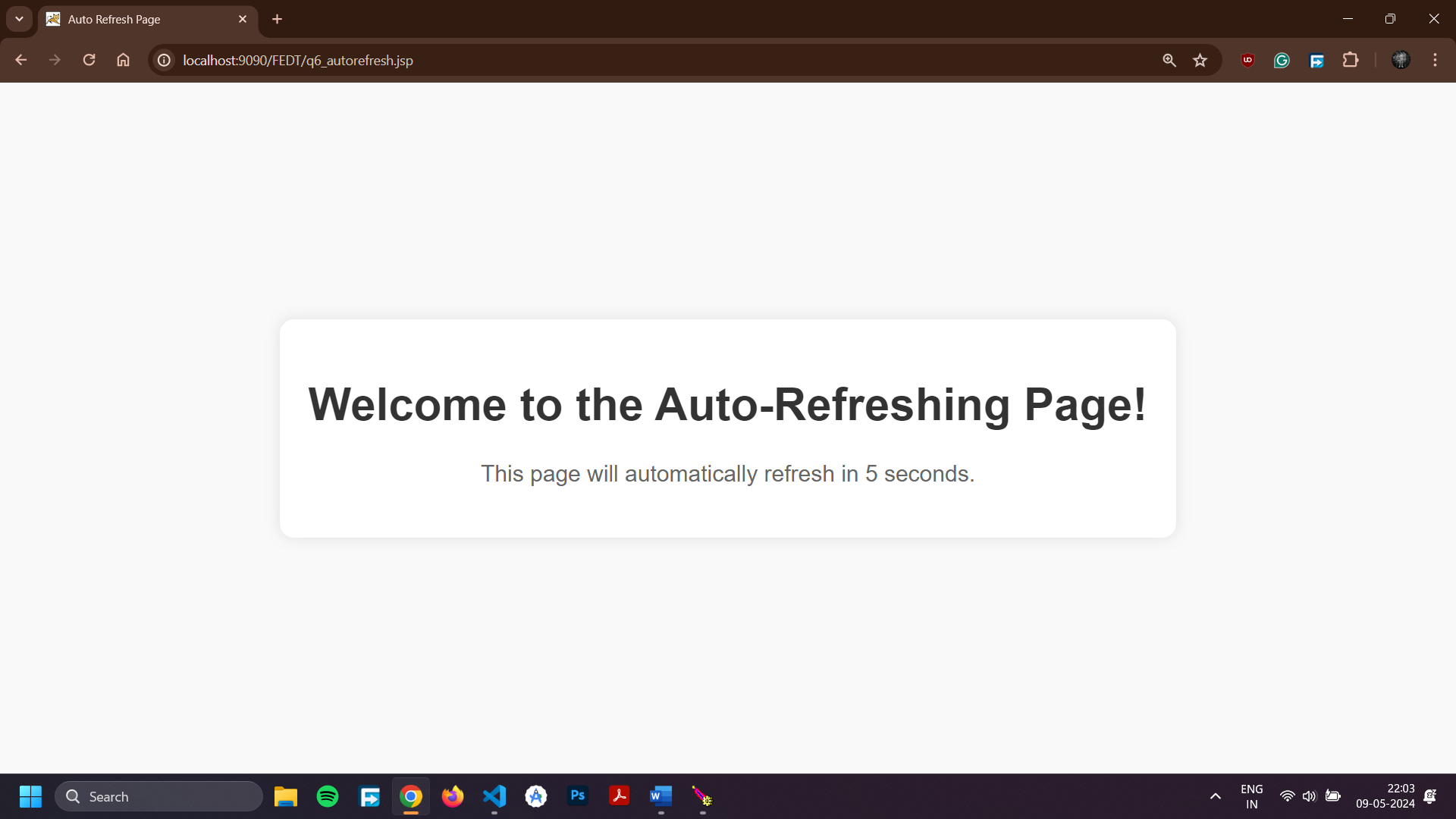
</p>

</div>

</body>

</html>

**Output**



**Practical – 07**

**Aim: Write a JSP Program to upload file into server.**

**Program Code**

**Index.html**

<!DOCTYPE html>

<html>

    <head>

        <title>Upload File</title>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

    </head>

    <body>

        <h1>Upload File</h1>

        <form action="FileUploadServlet" method="post" enctype="multipart/form-data">

            <input type="file" name="file" />

            <br/><br/>

            <input type="submit" value="Upload" />

        </form>

    </body>

</html>

**FileUploadServlet.java**

import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.MultipartConfig;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.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 String UPLOAD\_DIR = "D:/Documents/USICT-MCA/Semester-2-USICT-MCA/IT620 - Front End Design Techniques/IT664 - Front End Design Techniques Lab/q7\_upload/upload";

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

File uploadDir = new File(UPLOAD\_DIR);

if (!uploadDir.exists()) {

uploadDir.mkdirs();

}

for (Part part : request.getParts()) {

String fileName = extractFileName(part);

part.write(uploadDir + File.separator + fileName);

}

PrintWriter out = response.getWriter();

out.println("Files uploaded successfully!");

}

private String extractFileName(Part part) {

String contentDisp = part.getHeader("content-disposition");

String[] items = contentDisp.split(";");

for (String s : items) {

if (s.trim().startsWith("filename")) {

return s.substring(s.indexOf("=") + 2, s.length() - 1);

}

}

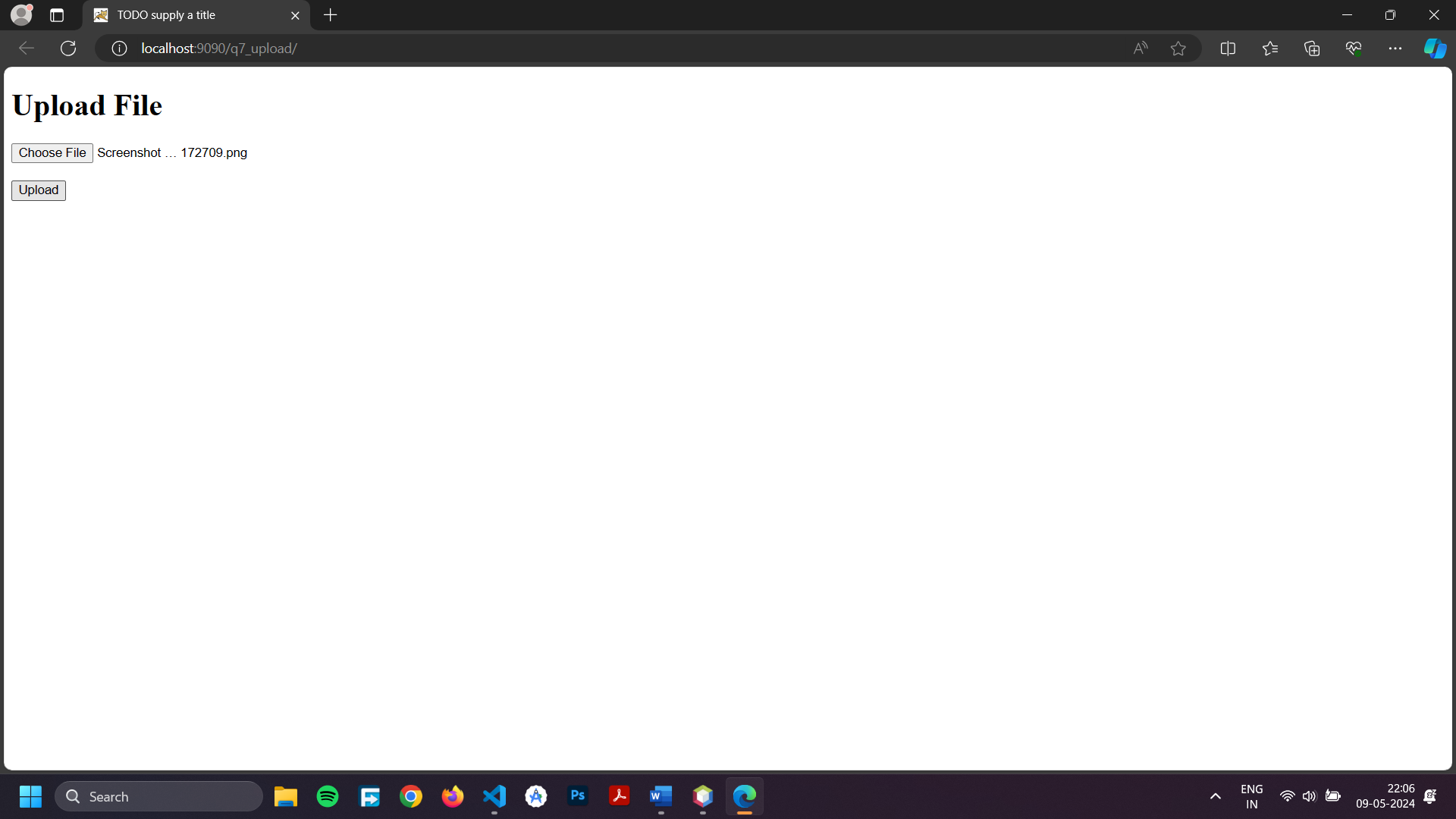
return "";

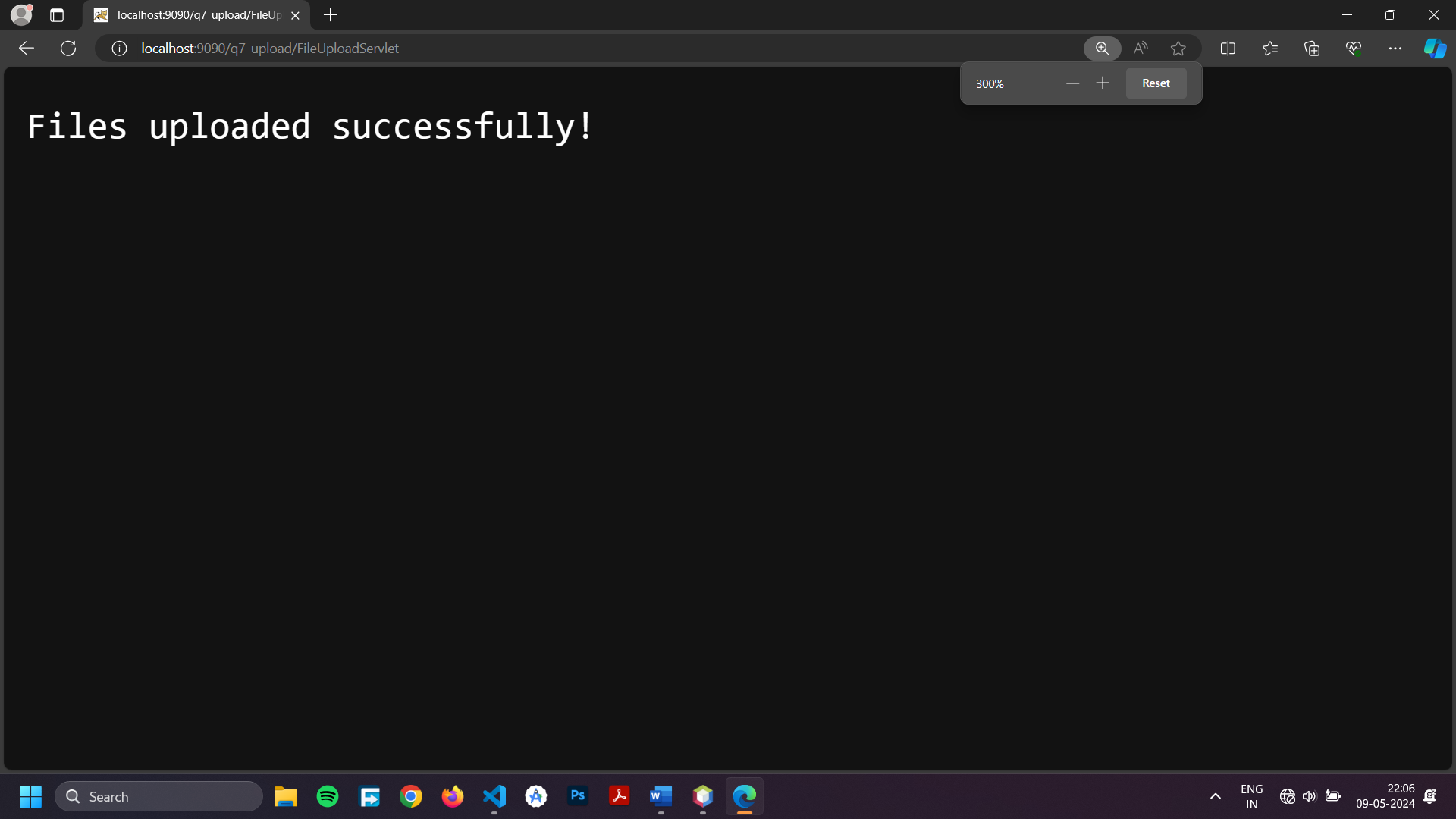
}

}

}

**Output**





**Practical – 08**

**Aim: Write a Generic Servlet program to display your own Enrolment number and Name using Apache Tomcat server.**

**Program Code**

**Myinfo.jsp**

<!DOCTYPE html>

<html>

<head>

<title>My Info</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h1>My Enrollment Number: ${enrollmentNumber}</h1>

<h1>My Name: ${name}</h1>

</body>

</html>

**MyInfoServlet.java**

import java.io.IOException;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "MyInfoServlet", urlPatterns = {"/myinfo"})

public class MyInfoServlet extends HttpServlet {

@Override

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

String enrollmentNumber = "04316404523";

String name = "Raj Khatri";

System.out.println("Setting attributes - Enrollment Number: " + enrollmentNumber + ", Name: " + name);

request.setAttribute("enrollmentNumber", enrollmentNumber);

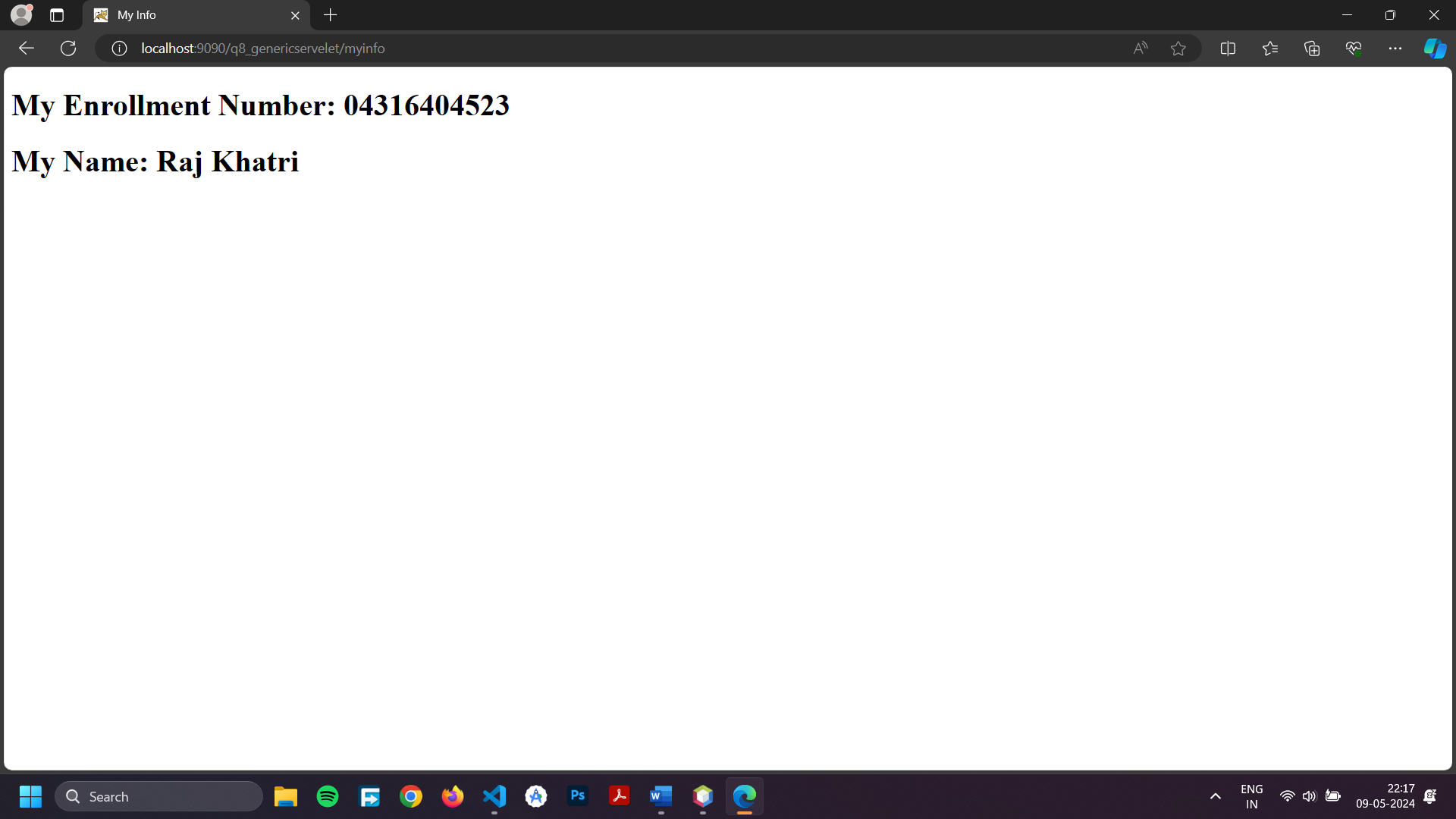
request.setAttribute("name", name);

request.getRequestDispatcher("/myinfo.jsp").forward(request, response);

}

}

**Output**



**Practical – 09**

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

**Program Code**

import java.io.\*;

import java.util.\*;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

public class RequestHeader extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException

{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

Enumeration e = request.getHeaderNames();

while (e.hasMoreElements()) {

String name = (String)e.nextElement();

String value = request.getHeader(name);

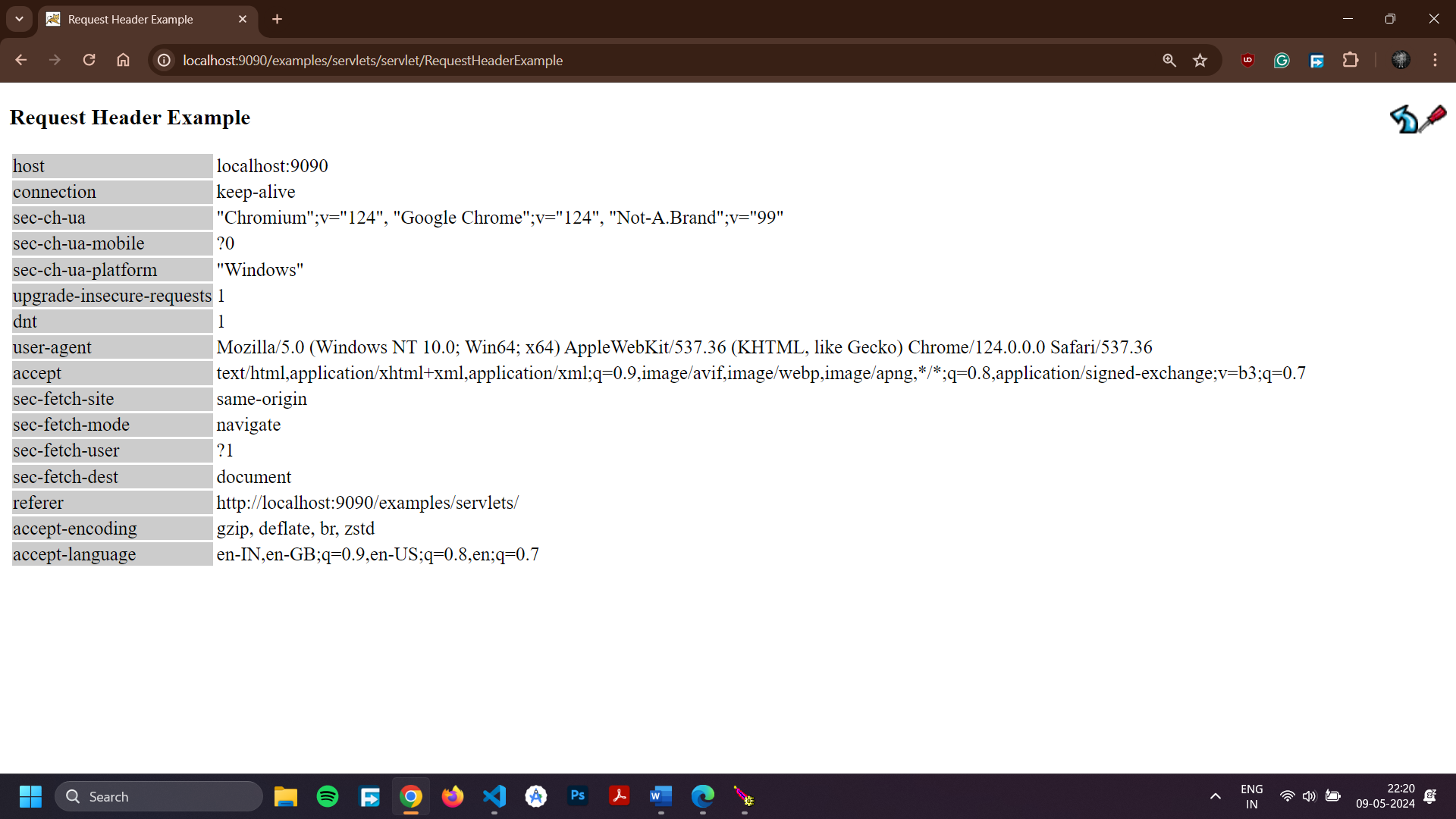
out.println(name + " = " + value);

}

}

}

**Output**



**Practical – 10**

**Aim: Write a HTTP Servlet program to create a Cookie.**

**Program Code**

import java.io.\*;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

public class CookieExample extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException

{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// print out cookies

Cookie[] cookies = request.getCookies();

for (int i = 0; i < cookies.length; i++) {

Cookie c = cookies[i];

String name = c.getName();

String value = c.getValue();

out.println(name + " = " + value);

}

// set a cookie

String name = request.getParameter("cookieName");

if (name != null && name.length() > 0) {

String value = request.getParameter("cookieValue");

Cookie c = new Cookie(name, value);

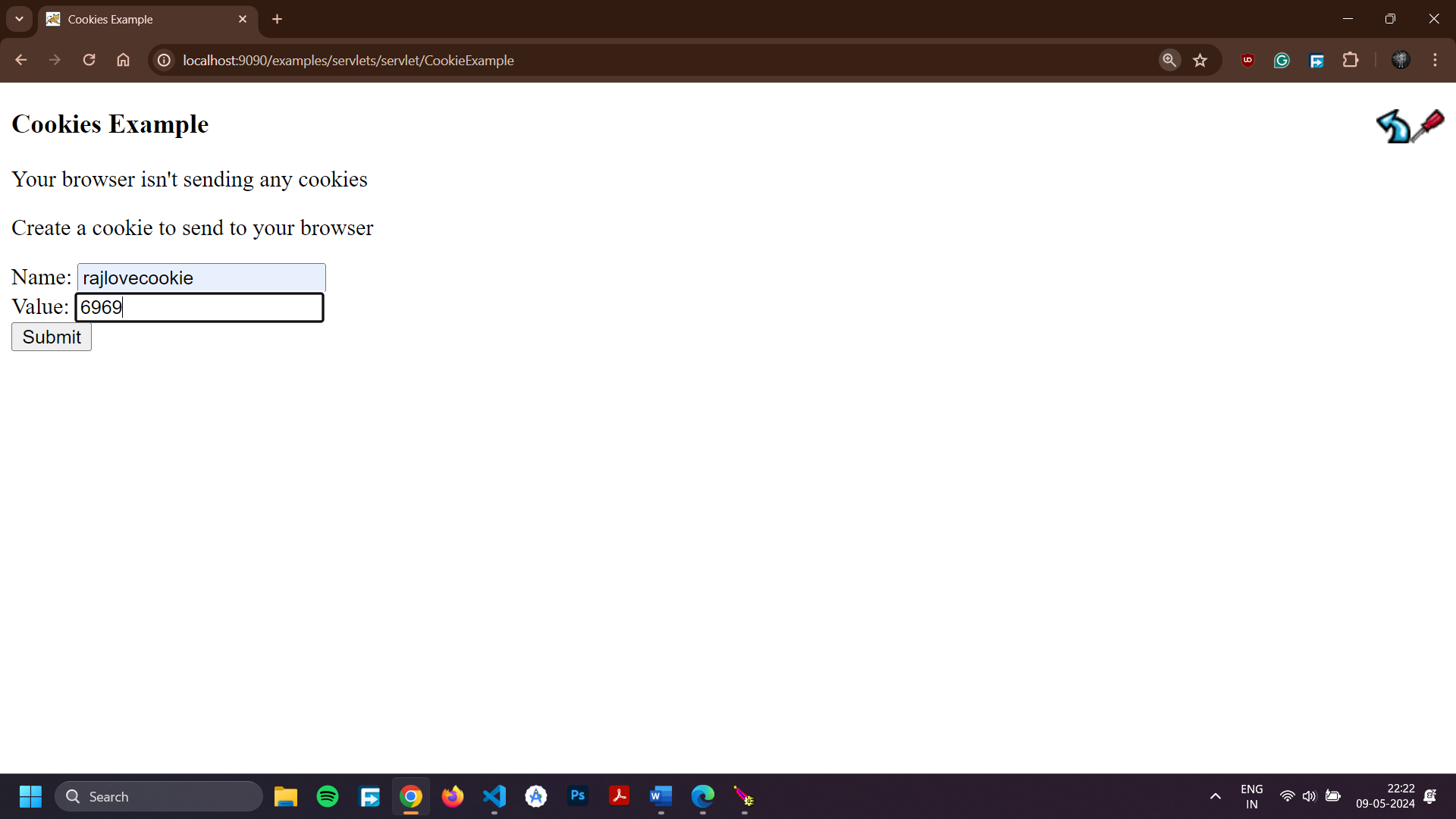
response.addCookie(c);

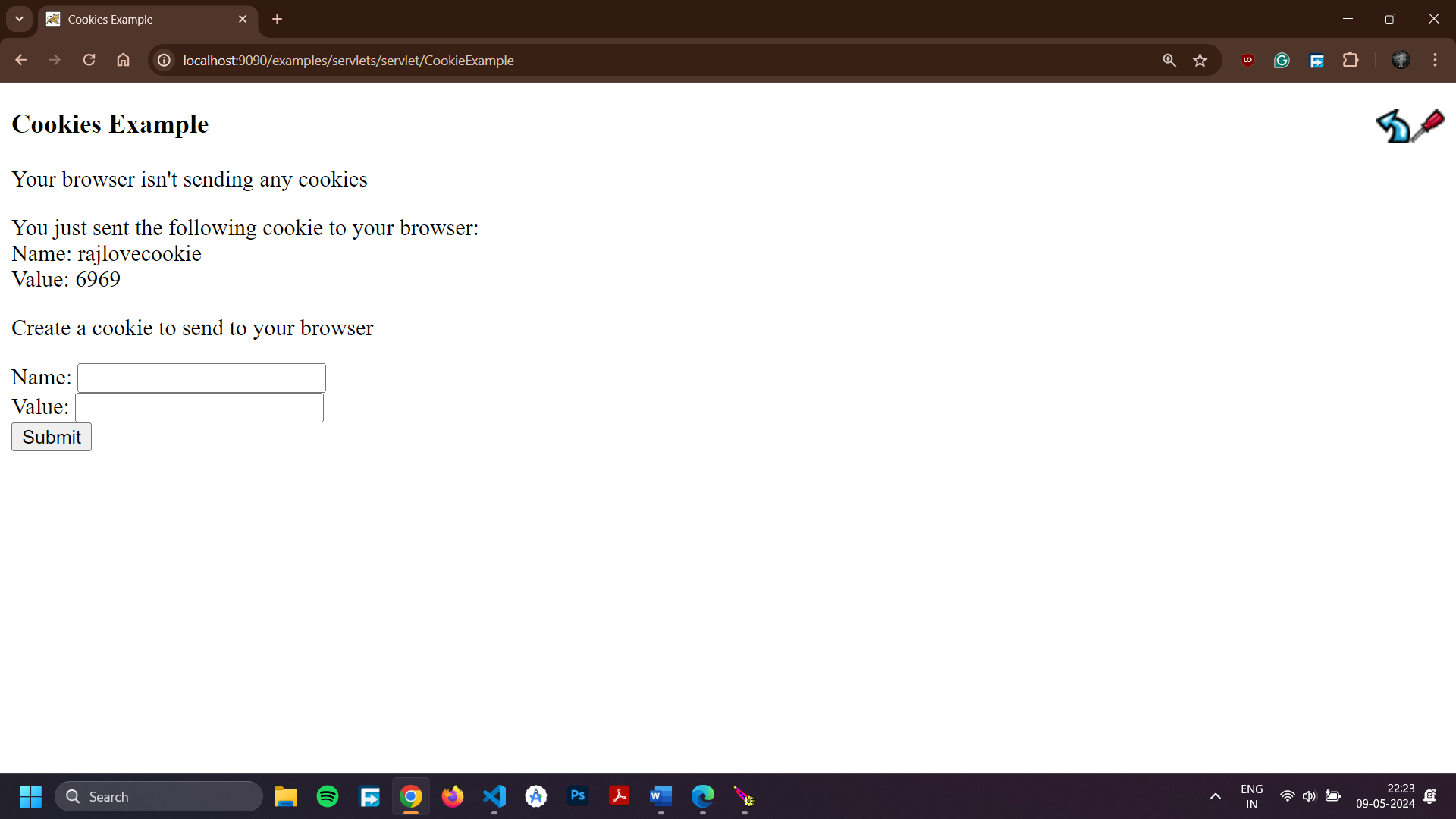
}

}

}

**Output**





**Practical – 11**

**Aim: 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.**

**Program Code**

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/FetchDataServlet")

public class FetchDataServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

try {

Class.forName("com.mysql.cj.jdbc.Driver");

try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/employee", "root", "admin")) {

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT EmployeeID, EmployeeName, DOB, Address, Department, DOJ, Position FROM employees");

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

out.println("<h2>Employee Records</h2>");

out.println("<table border='1'><tr><th>Employee ID</th><th>Name</th><th>DOB</th><th>Address</th><th>Department</th><th>DOJ</th><th>Position</th></tr>");

while (rs.next()) {

out.println("<tr><td>" + rs.getInt("EmployeeID") + "</td><td>" + rs.getString("EmployeeName") + "</td><td>" + rs.getString("DOB") + "</td><td>" + rs.getString("Address") + "</td><td>" + rs.getString("Department") + "</td><td>" + rs.getString("DOJ") + "</td><td>" + rs.getString("Position") + "</td></tr>");

}

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

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

}

} catch (ClassNotFoundException | SQLException e) {

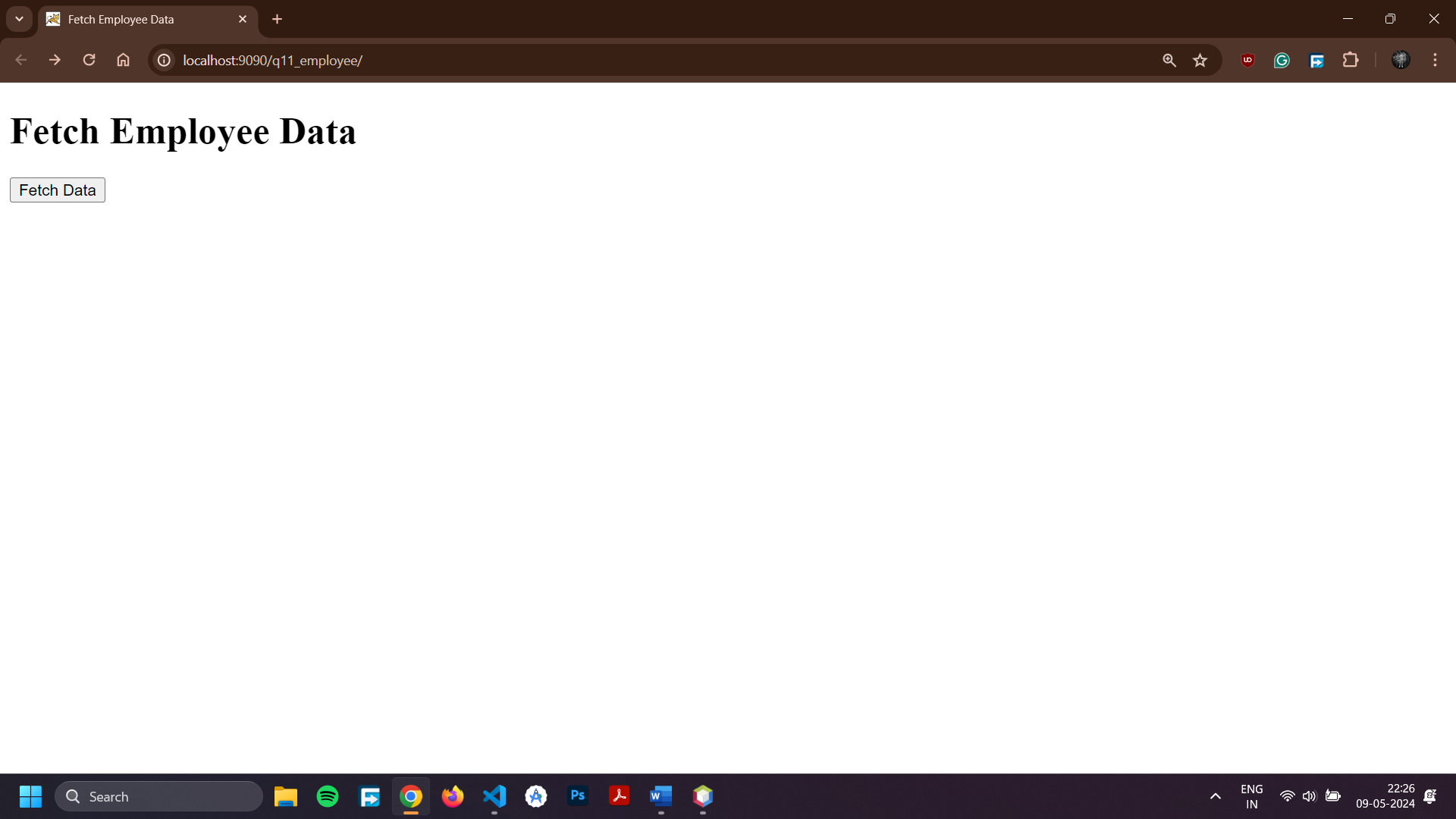
out.println("Error: " + e.getMessage());

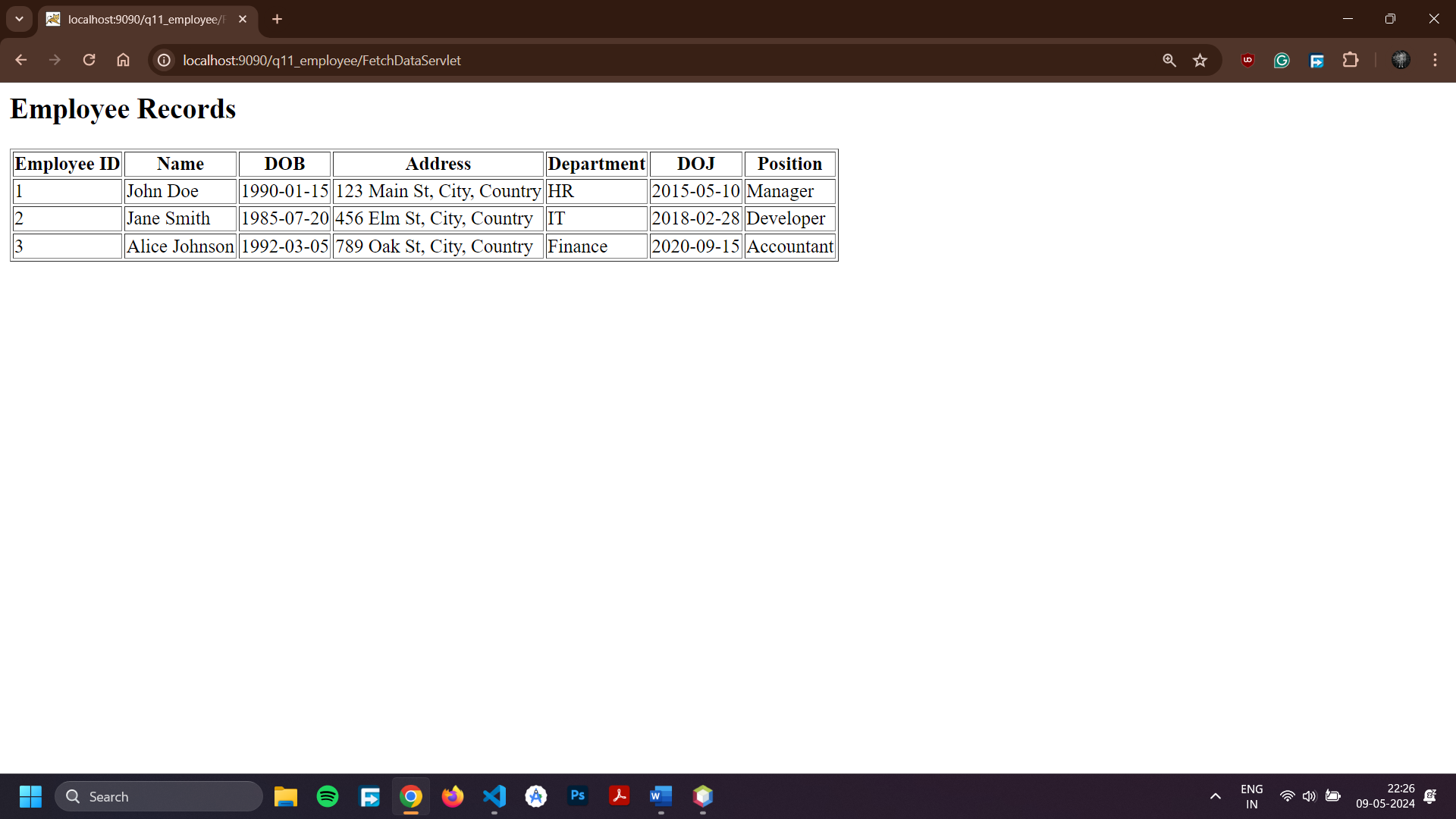
}

}

}

**Output**

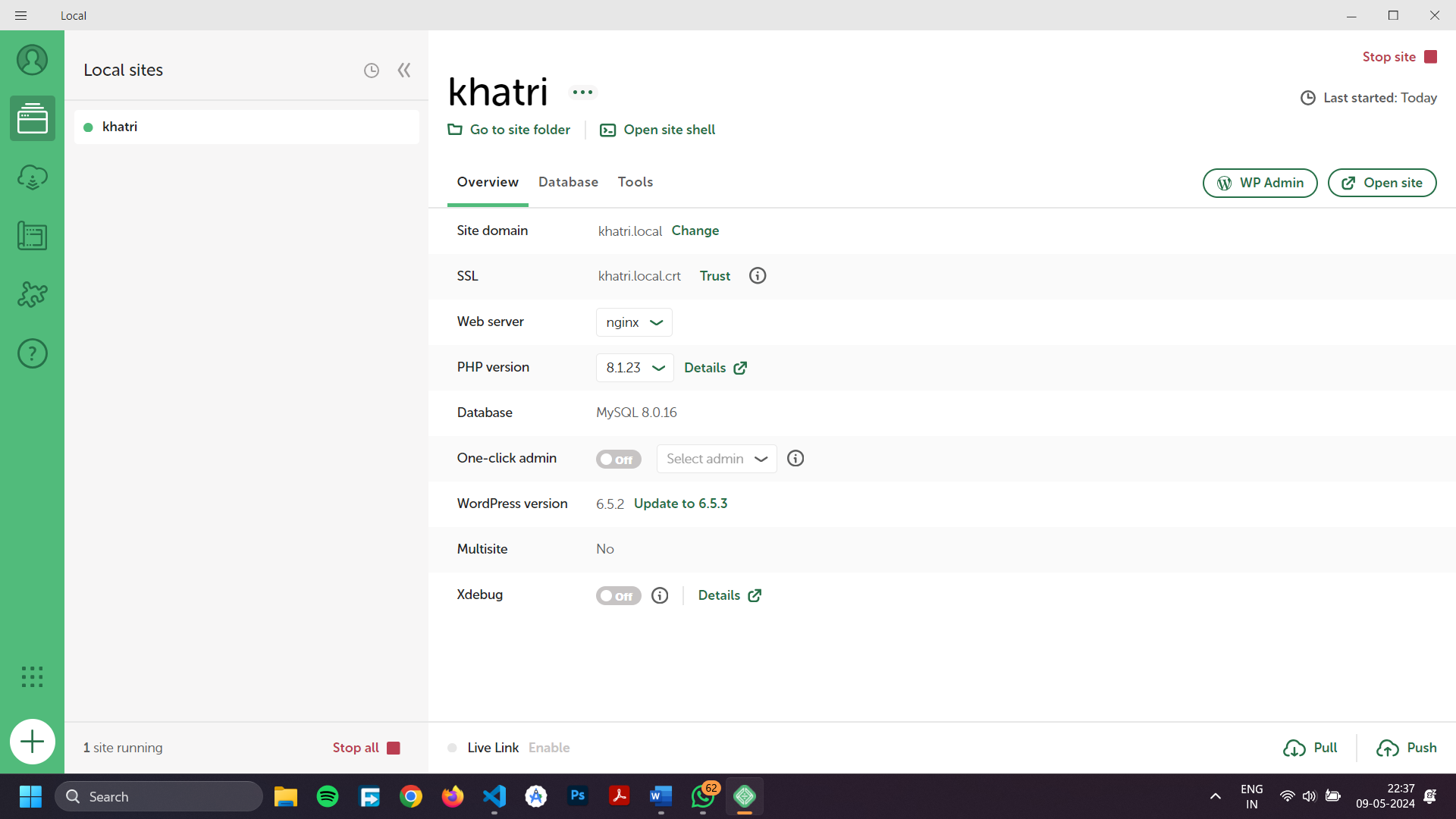




**Practical – 12**

**Aim: Create your portfolio website using LocalWP and Wordpress**

**LocalWP Dashboard**



**Output**

