Experiment 1:

A web page is created using HTML, consists of various divisions which include, head, body and footer parts. Write a program which include various tags used inside the header tag.

Aim:

To create a web page using HTML that demonstrates the use of various tags inside the <head> tags.

procedure:

-> start with a basic HTML 5 template structure.
-> inside the <head> tag.
  - <title> for page title
  - <meta> tags for description.
-> save the file with a .html extension.
-> open the file in web browser.

program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="description" content
                    ="head tag demo">
  <meta name="author" content =
                    "student">
```

```
<meta name = "viewport" content =
        "width = device-width, initial>
<title> Sample page </title>
<style>
body {font-family : Arial; background
                    : #f0f8ff}
</style>
<script> console.log ("page loaded!")
</head>
<body>
<h1> welcome to my web page </h1>
</body>
</html>
```

output :
=

Welcome to MY web page
This page demonstrates the use of head section tags.
© 2025 student Name.
All rights reserved.

28/9/25

Result :
=

A web page is created using HTML consists of various division is executed succesfully.

Experiment 2

write a Javascript program to get the numbers from the user and display only the nearest odd numbers.

Aim:

To write a Javascript program that takes a number input from the user.

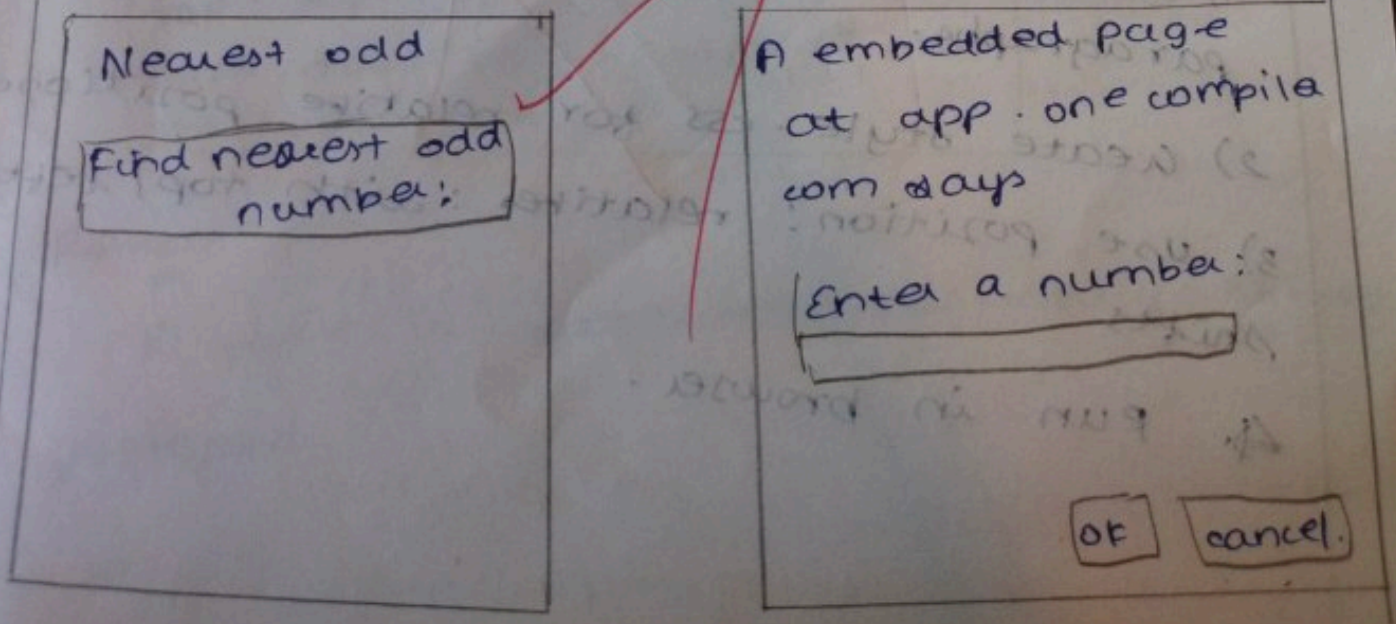
procedure:

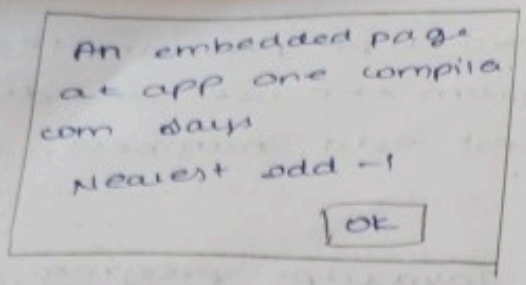
-> prompt the user to enter a number
-> convert the input to an integer.
-> use the modulus operator %. to check if the number.
-> display the result using alert().

program:

```
<html> <head><title>odd</title>
<script>
function odd(){} { let n = +prompt ("Enter numb
</script></head>
<body>
<h1> Nearest odd finder</h1>
</body></html>
```

output:





Result:

A code is executed succesfully through Notepad.

Experiment 3:

Create a HTML web page / change the position of the element relative to the parent element and relative to itself using relative positioning with a CSS file.

Aim:

To create an HTML page that changes element positions using relative positioning with CSS.

procedure:

1) create index.html with headings / paragraphs.

2) Create style.css for relative positioning

3) Use position: relative; with top / left shifts

4. Run in browser.

program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Relative position</title>
<link rel = "stylesheet" href = "style.css">
</head>
<body>
<div class = "box">
<h1 class = "more"> Welcome </h1>
<p> pargraph stays normal. </p>
</div>
</body>
</html>.
```

output:

Welcome to css poisitioning.

This paragraph remains normally positioned.

style.css. palent {border: 2px solid blue; padding : 20px; width: 400 px: height : 200 px: position: relative; } child {position: relatiw top : 30px; /* more the h1 element 30 px down */ left : 20 px: /* more the h1 element.

23/9/2025

Result:

A code is excuted sucessfully through Notepad.

Experiment 4:

Aim:
To create a Java servlet that collects student registration details via a form and display them on the same page.

procedure:
1. create a servlet.
2) Inside doGet(), show a form
3. on submit, fetch values with request.getParameter().
4. Display details using printWriter.
5. Deploy on Tomcat.

program:

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

public class studentRegistrationServlet
extends HttpServlet {
  public void doGet(HttpServletRequest q,
HttpServletResponse s)
  throws IOException, ServletException {

s: setContentType("text/html");
  PrintWriter o = s.getWriter();

  String n = q.getParameter("name");
      r = q.getParameter("regno");
      c = q.getParameter("course");
```

```
o. print ("<html><body>"
    + "Name : <input name ='name'/h2>
    + "RegNO : <input name ='regno'
                <br>
else
  o. print ("<h2>Details</h2>"
        + "Name:" + n + "<br>"
        + "Regno:" + r + "<br>"
        + "course:" + c + "<br>"
        + "Email:" + e).

    }
}.
```

Output:

```
Student registration.
      Details
");  out.println"
Name:"+ name+"
");  out.println("
Register num:" +regno"
");out.println("
course:" +course"
") out.println("
");  out.println(".
```

21/9/2023

Result:

The program is executed successfully.

Experiment 5

Make a registration form and help the organization to collect their employee's details like, employee's name, id, address, department, mobile number and select their city from the drop down list box using HTML and CSS.

Aim:

Create an HTML + CSS employee registration like form to collect details (name, ID, address, department, mobile, city).

procedure:

→ make HTML file with form elements
→ Add input, textarea, and select tags
→ create CSS for styling
→ Run in browser.

program:

```
<!DOCTYPE html>
<html>
<head>
<title> Employee Registration </title>
<link rel = "stylesheet" href = "style.css">
</head>
<body>
<div class = "container">
```

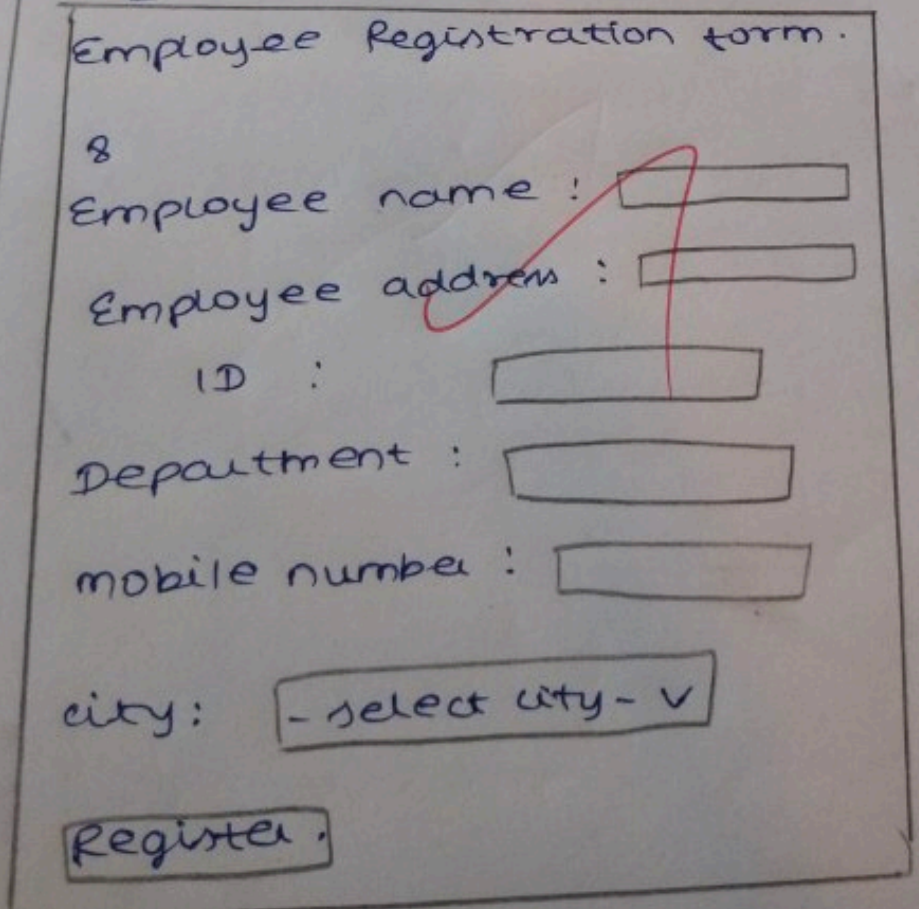
```
<h2> Employee registration
<form>

<label> Name : </label>
<input type = "text" name = "name">
<label> ID : </label>
<select name = "city" required>
<option value = " "> --- select --</option>
<option> chennai </option> <option>
                          Bangalore </option>

</select>

</form>
</div>
</body>
</html>
```

Output :



23/9/2025

Result :

=

program is executed successfully.