MIT-ADT
UNIVERSITY
PUNE, INDIA
A leap towards World Class Education
(Established by MIT Art, Design and Technology University Act, 2015
(Maharashtra Act No. XXXIX of 2015)

MIT Art Design and Technology University

MIT School of Computing, Pune

Department of Computer Science and Engineering

Web Technology Programming Lab

Class - S.Y. (SEM-IV)

Divisions: SY-1, SY-2 , SY-21

(A.Y. 2024 – 2025- Term-II)

Team Members

1. Amit Uttarkar
2. Bhagyshree Shendkar
3. Pooja Ramesh Oza

Lab File Index

1	Course Structure & List of experiments
2	Cos, PEOs, POs, and PSO
3	Time Table
4	List of Hardware / Software Requirement
5	Practical Plan
6	Lab Manual
7	Continuous Assessment Record a. Practical Plan and Practical Attendance b. Practical Assignments c. PBA I, PBA II, End Semester Practical Exam. d. Attendance Record, Result and Analysis
8	Criteria to identify below average and above average students in practical/ Remedial actions record: Notice, time table, Attendance record, and Improvement details
9	PBA 1, PBA2, Ends Semester Practical Problem Statements
10	Unit wise Oral Question Bank
	Mapping of PEOs, POs, PSO, and COs

Prepared By – TTT Coordination Committee

Prof. Dr. Shraddha Phansalkar
Head - Dept. of CSE (Core)

Prof. Dr. Ganesh Pathak
Head - Dept. of CSE (ISA)

Prof. Dr. Rajneeshkaur Sachdeo
Dean School of Computing

Prof. Dr. Vipul Dalal
Director MIT School of Computing

COURSE STRUCTURE

Web Technology

SEMESTER – IV

Course Code: 23CSE3003		**Course Credits:**	**03**
Teaching Hours / Week (L:T:P):	**3:0:0**	**CA Marks:**	**40**
Total Number of Teaching Hours:	**03/week**	**END-SEM Marks:**	**60**

Course Prerequisites:

- Fundamentals of Programming and Networking

Course Description: It covers HTML and CSS for web design, client-side scripting with JavaScript and AJAX, server-side development using PHP, and advanced tools like Java Servlets, JSP, jQuery, and Bootstrap. The course aims to equip students with the skills needed to build dynamic and interactive web applications.

Course Learning Objectives: This course will enable the students to:

The main objectives of the course are to:

- To introduce HTML and CSS language for designing web pages
- Demonstrate the concepts of PHP language in server-side scripting
- Demonstrate proficiency in Server side programming with Java Servlets and JSP
- Manage Client-side scripting with JavaScript and AJAX.

Course Outcome:

On completion of the course, the learner will be able to–

CO1: Develop modern web pages using HTML and CSS features with different layouts as per the needs of applications.

CO2: Use JavaScript to develop dynamic web pages.

CO3: Use server-side scripting with PHP to generate the web pages dynamically using the database connectivity.

CO4: Develop modern Web applications using client and server-side technologies and web design fundamentals

UNIT – I	**Introduction To HTML and CSS**	**9 -- Hours**

Markup Language (HTML): Introduction to HTML - Formatting and Fonts - Commenting Code – Anchors – Backgrounds – Images – Hyperlinks – Lists – Tables – Frames - HTML Forms.

CSS: Introduction; Levels of style sheets; Style specification formats; Selector forms; Property value forms; Font properties; List properties; Color; Alignment of text; Background images.

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos**
	Self-study / Do it yourself /
	Experiential Learning Topics:
	Case Study / PBL - Project Based Learning

UNIT – II	**Introduction to JavaScript**	**9-- Hours**

Overview of JavaScript, using JS in an HTML (Embedded, External), Data types, Control Structures, Arrays, Functions and Scopes.

Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos**	
	Self-study / Do it yourself	
	Experiential Learning Topics:	
	Case Study / PBL - Project Based Learning	
UNIT – III	**Advanced JavaScript**	**9 -- Hours**
DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM. JavaScript Events, Objects in JS, Event Handling		
Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos**	
	Self-study / Do it yourself /	
	Experiential Learning Topics:	
	Case Study / PBL - Project Based Learning	
UNIT – IV	**PHP**	**9-- Hours**
PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, using MySQL with PHP.		
Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos**	
	Self-study / Do it yourself	
	Experiential Learning Topics:	
	Case Study / PBL - Project Based Learning	
UNIT – V	**Introduction to JQuery and Bootstrap**	**9-- Hours**
JQuery: Introduction to JQuery, Loading JQuery, Selecting elements, changing styles, creating elements, appending elements, removing elements, handling events. Bootstrap: Introduction of Bootstrap, Typography, Tables, Images, Buttons.		
Pedagogy	**ICT Teaching / PowerPoint Presentation and Videos**	
	Self-study / Do it yourself	
	Experiential Learning Topics:	
	Case Study / PBL - Project Based Learning	

Text Books:

1. Programming the World Wide Web – Robert W. Sebesta, 4th Edition, Pearson Education, 2008.
2. Harvey & Paul Deitel & Associates, Harvey Deitel and Abbey Deitel, "Internet and World Wide Web - How To Program", Fifth Edition, Pearson Education, 2011.

Reference Books:

1. Adam Bretz & Colin J Ihrig, —Full Stack Javascript Development with MEAN , SPD, ISBN-13: 978-0992461256
2. Giulio Zambon, — Beginning JSP, JSF and Tomcat , Apress Publication, ISBN-10: 1430246235; ISBN-13: 978-1430246237

URLs (Optional) - List of Online Courses

ASSESSMENT AND EVALUATION PATTERN		
WEIGHTAGE	Continuous Assessment (CA)	End Semester Assessment (ESA)
CA - Assignments (PBL / Case Study / Presentation / Poster Presentation / Seminar / Group Discussions / Quiz / Test / Open Book Examination / Book or Article Review / Jury / Concept Map /		

Mind Maps
ESA - Bloom's Taxonomy Levels: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating
Experiential Learning - Internship (Summer / Winter) / Industry Visit / Site Visit / Field Visit
Project Based Learning – Mini (Minor) Project / Major Project / Website Making / e-Portfolios

Course Articulation Matrix (CO-PO Mapping) [Subject Title – Subject Code]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2
CO-1	1	2	1	1		1	1		2	1	2	2		1
CO-2	2	1	2				2		2	1		2	2	
CO-3			2			2	1		1	1		2	2	
CO-4		2		2			1		2	1		2		

3 – HIGH, **2** – MEDIUM, **1** - LOW

	WT Lab Experiments	
1	Design a form for Company registration which includes all elements of form tags	
2	Create a webpage for any of the following and make use of CSS 1. Coffee Shop 2. Car Service Center 3. Flower Shop	
3	Write a JavaScript code to take inputs from user and display that inputs in following pattern "Hello …… Welcome To World of JavaScript".	
4	Write a JavaScript code to take color from user and apply that color to Div tag. Create one button on every click of button different colors should be applied to background	
5	Write a JavaScript code to add the button to screen on click on button. Write a JavaScript code to change the color of button with following events 1.Onmouseover 2.Onmouseout 3.Onclick	
6	Attach a click event listener to a button. When clicked, the button should change the text content of a target element to "Button Clicked!".	
7	Write a PHP script to take number from user and print the table of that number.	
8	Write a PHP script to print the data of specific user from database	
9	Write a PHP script to create cookie of color.	
10	Write a JQuery code to give animation to div on click of button Write a JQuery code to add dynamic element on page on click of button	
11	Display a popup on mouse over of link and add two text boxes and button to that popup.	

Program Outcome (PO's)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Lifelong learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1 To apply the acquired knowledge of Computer Science & Engineering to build sustainable solutions to real-world problems in society.

PEO2 Attain the ability to adapt quickly to new environments and technologies, assimilate new information, and work in multi-disciplinary areas with a strong focus on higher studies, innovation and entrepreneurship.

PEO3 To practice collaborative learning and team spirit as professionals with project-based learning and technical activities.

PEO4 To be able to effectively present, communicate and exhibit professionalism with ethical values and empathy for needs of society.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1 Apply fundamental knowledge in Computer Science and Engineering with advanced learning in specialized domains of Data Science, Artificial Intelligence, Networking, Security and high-performance computing using standard practices in Engineering and Sciences.

POS-2 Apply standard industry-based practices and techniques to provide solutions to real world problems.

List of Hardware and software requirement.

Hardware Requirements

Computers: Processor: Intel Core i3 or AMD Ryzen 3 (or higher) with multi-core support.

RAM: Minimum 8 GB (16 GB recommended).

Storage: SSD with at least 512 GB capacity for fast read/write operations. Additional HDD storage for data storage may be required.

Monitors: Monitors per workstation for increased productivity.

Other Peripherals:

High-quality keyboards and mice.

PS (Uninterruptible Power Supply) for power backup.

Adequate cooling solutions to maintain optimal operating temperatures.

Software Requirements

Operating Systems: Primary: Windows 10/11, macOS, or Linux (Ubuntu, CentOS, etc.) depending on the lab requirements.

Programming Languages and Compilers:

Xampp server, MySQL, html, php, Java sdk

Integrated Development Environments (IDEs):

Visual Studio Code

Notepad++

Topic: Introduction to WWW, HTML and CSS

Aim: Experiment No: 1, 2

1. Design a Company Registration Form with CSS styles that includes all of the form tag's features. Show the relevant Form output in a compatible browser.
2. a] Create an HTML script for Coffee: shop online web content utilizing HTML tags and CSS capabilities. Use a coffee cup and relevant images to enhance the presentation. Provide a coffee type selection option, as well as a rate for the coffee type pick. Tables should be used whenever possible.
   b] Create an HTML Script for a Car Service Center utilizing HTML tags and CSS features: Enhance content with numerous car workshop photographs and car accessories, as well as details on various car servicing packages.
   c] Create an HTML Script for a Flower Shop utilizing HTML tags and CSS features: CSS Selectors with Optional Beautification demonstrate output with a variety of floral photos and

the available flower stock in the floweriest shop. Provide animation using CSS

Objective:

To create a flower shop /Car Shop/Coffee shop webpage using HTML & CSS.

Theory:

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications. Learn more below about:

What is HTML?

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.

- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

With HTML, authors describe the structure of pages using markup. The elements of the language label pieces of content such as “paragraph,” “list,” “table,” and so on.

What is CSS?

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML- based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.

Some of the HTML tags are as follows:

Document structure tag:

- **HTML tag**: It is the root of the HTML document which is used to specify that the document is HTML.

  Syntax: <html> Statements... </html>

- **Head tag:** The head tag is used to contain all the head elements in the HTML file. It contains the title, style, meta, … etc tag.

  Syntax: <head> Statements... </head>

- **Body tag:** It is used to define the body of an HTML document. It contains images, tables, lists, … etc.

  Syntax: <body> Statements... </body>

- **Title tag:** It is used to define the title of an HTML document.

  Syntax: <title> Statements... </title>

Content container tag:

- **Heading tag:** It is used to define the heading of an HTML document.

Syntax:

<h1> Statements... </h>

<h2> Statements... </h2>

<h3> Statements... </h3>

<h4> Statements... </h4>

<h5> Statements... </h5>

<h6> Statements... </h6>

- **Paragraph tag:** It is used to define paragraph content in an HTML document.

  Syntax: Statements...

- **Emphasis tag:** It is used to render as emphasized text.

  Syntax: Statements...

- **Bold tag:** It is used to specify bold content in an HTML document.

Syntax**:** Statements...

- **Italic tag:** It is used to write the content in italic format.
  Syntax**:** Statements...
- **Small (text) tag:** It is used to set the small font size of the content.
  Syntax**:** <small> Statements... </small>
- **Underline tag:** It is used to set the content underline.
  Syntax**:** <u> Statements... </u>
- **Deleted text tag:** It is used to represent deleted text. It crosses the text content.
  Syntax**:** <strike> Statements... </strike>
- **Anchor tag:** It is used to link one page to another page.
  Syntax**:** Statements...
- **List tag:** It is used to list the content.
  Syntax**:** Statements...
- **Center tag:** It is used to set the content into the center.
  Syntax**:** Statements...

Transform Property:

The transform property in CSS is used to change the coordinate space of the visual formatting model. This is used to add effects like skew, rotate, translate, etc on elements. The transformations can be of 2-D or 3-D type.

Syntax:

transform: none|transform-functions|initial|inherit;

Translate Property:

The translate() function is an inbuilt function which is used to reposition an element in a horizontal and vertical direction.

Syntax:

translate(tx) or translate(tx, ty)

Parameters:

- tx: This parameter holds the length of translation corresponding to x-axis.
- ty: This parameter holds the length of translation corresponding to y-axis. If its value is not defined then it takes 0 as default value.

Few CSS properties are mentioned below:

1. **Display**: Specifies how an element is displayed on screen
2. **Width and Height**: Width and height properties are used closely with display: block and display: inline to set the width and height of HTML elements while creating a website.

3. **Font Family:** The **font-family** property allows us to change the particular font we are using.

You may select any font which is installed on the client's computer with this property.

4. **Font Weight:** The **font-weight** css property allows us to specify how thick the lines of the characters are.
5. **Font Size: Font-size** may be specified using a few different types but the easiest to work with is pixels (px).
6. **Color:** The **color** property allows us to specify the colour of the text.
7. **Background Color:** The **background-color** property allows us to specify the background colour for the element.
8. **Padding: padding** refers to the space around out content. We've included the **background-color** property here as well just to make it a bit easier to see what is happening.
9. **Margin: margin** is similar to padding in that is affects the spacing around the content. It is a second area outside the padding however. Compare the example below to the one above for padding and note where the background colour ends.
10. **Border:** The **border** is in between the padding and margin. I'll outline basic usage here but there are various properties available to control the look of the border with greater detail

Conclusion:

Output (Screenshots):

Frequently Asked Questions:

1. What is HTML stands for?
2. What is importance of HTML and CSS?
3. How to use HTML CSS together?
4. How to link HTML to CSS?
5. Is CSS Mandatory in HTML?

Topic: Introduction to JAVASCRIPT

Experiment No: 3,4,5,6

Aim:

3. a) Write a JavaScript code to take inputs from user and display that inputs in following pattern "Hello …… Welcome to the World of JavaScript".

b] Write a JavaScript code to perform arithmetic operation by taking values from users (Add, Sub, Mul, and Div) (1. 4 Button, 2.Dropdown)

4. a) Create one button on every click of button different colors should be applied to the background.

b) Create one button on every click of button different images should be applied.

c] Declare a JavaScript String array of colors say colors = ["Red", "Green", "Blue"] Accept a value from the user and add it to the array if the value is not present in the array.

d] Write a JavaScript code to apply font color and background color to Heading from dropdown, if font color and background color is same no changes should be reflected.

e] Create an array using JavaScript and display the occurrences of a specific character [For example; arr = ['a, 'b', 'a', 'c', 'z'] Output should be occurrences of a is 2].

5. a] Write a JavaScript code to append static row to the table on click of button

b] Write a JavaScript code to append anchor tag that redirects to the new page

c] Write a JavaScript code to change the color of button with following events

1. Onmouseover
2. Onmouseout
3. Onclick

d] Write a JavaScript code to change the background color of page onclick of array elements displayed as a list.

6. Write a JavaScript to create shopping applications which adds the books in cart, updates the existing books, delete the book and display the same. Create proper UI for the same.

Objective: To learn about concept and implementation of JAVASCRIPT

Theory:

What is Javascript? Why it is called Client side Scripting language

Client side: JavaScript is a client-side language, which means it gets executed at the client side

(i.e, user side). On contrary, PHP is a server-side scripting language, as it gets executed at the server. Whenever you browse the web, all the HTML, CSS & JS files are fetched from the server & then executed/interpreted at your side by your browser.

Scripting language: Since it is interpreted rather than compiled & are úsed to execute tasks one-by-one. More professionally,

A scripting or script language is a programming language that supports scripts: programs written for a special run-time environment that automate the execution of tasks that could alternatively be executed one-by-one by a human operator. Scripting languages are often interpreted

Explain Javascript functions used in above programs like with syntax and example

- **getElementByID()**

The getElementById() method returns an element with a specified value. The getElementById() method returns null if the element does not exist.
document.getElementById("demo").style.color
= "red";

- **innerHTML()**

The innerHTML property sets or returns the HTML content (inner HTML) of an element. let html = document.getElementById("myList").innerHTML;

- **value()**

The Object.values() method returns an array of a given object's own enumerable property values, in the same order as that provided by a for...in loop. (The only difference is that a for...in loop enumerates properties in the prototype chain as well.) console.log(Object.values(object1))

- **parseInt()**

The parseInt() function parses a string argument and returns an integer of the specified radix (the base in mathematical numeral systems). function roughScale(x, base) { const parsed = parseInt(x, base); if (isNaN(parsed)) { return 0; } return parsed * 100;
}

DOM

- It is an application programming interface.
- Dom represents a document as a hierarchical tree of nodes allowing dev to add, remove, and modify individual part of the page.
- Document node represent every document as a root. In this example the only child of the document node is html which is called a document element.
- Every piece of markup can be represented by a node in the tree: HTML elements are represented by element nodes, attributes are represented by attribute nodes, the document type is represented by a document type node, and comments are represented by comment nodes.
- In total, there are 12 node types, all of which inherit from a base type.
- DOM Level 1 describes an interface called Node that is to be implemented by all node types in the DOM.

- Every node has a nodeType property that indicates the type of node that it is. Node types are represented by one of the following 12 numeric constants on the Node type:

a. Node.ELEMENT_NODE (1)
b. Node.ATTRIBUTE_NODE (2)
c. Node.TEXT_NODE (3)
d. Node.CDATA_SECTION_NODE (4)
e. Node.ENTITY_REFERENCE_NODE (5)
f. Node.ENTITY_NODE (6)
g. Node.PROCESSING_INSTRUCTION_NODE (7)
h. Node.COMMENT_NODE (8)
i. Node.DOCUMENT_NODE (9)
j. Node.DOCUMENT_TYPE_NODE (10)
k. Node.DOCUMENT_FRAGMENT_NODE (11)
l. Node.NOTATION_NODE (12)

```
if (someNode.nodeType == Node.ELEMENT_NODE)
 {
   alert("Node is an element."); //won't work in IE < 9
  }
```

This example compares the someNode.nodeType to the Node.ELEMENT_NODE constant. If they're equal, it means someNode is actually an element. Unfortunately, since Internet Explorer 8 and earlier doesn't expose the Node type constructor, this code will cause an error. For cross- browser compatibility, it's best to compare the nodeType property against a numeric value, as in the following:

```
if (someNode.nodeType == 1)
    {
        alert("Node is an element."); //works in all browsers

 }
```

Properties of note type

- Two properties, nodeName and nodeValue, give specifi c information about the node. The values of these properties are completely dependent on the node type.
- nodeName is always equal to the element's tag name, and nodeValue is always null.
- Each node has a childNodes property containing a NodeList. A NodeList is an array-like object used to store an ordered list of nodes that are accessible by position. Keep in mind that a NodeList is not an instance of Array even though its values can be accessed using bracket notation and the length property is present.
- Node Relationship child parent

Conclusion:

Output (Screenshots):

Frequently Asked Questions:

1. What are all the looping structures in JavaScript?
2. What are the differences between Java and JavaScript?
3. What are JavaScript Data Types?
4. Is it possible to break JavaScript Code into several lines?
5. What are undeclared and undefined variables?
6. Write the errors shown in JavaScript?
7. What is the difference between JavaScript and Jscript?
8. How to convert the string of any base to integer in JavaScript?
9. What are the types of Pop up boxes available in JavaScript?
10. What is the difference between an alert box and a confirmation box?
11. What is the difference between call() and apply() methods?
12. How to target a particular frame from a hyperlink in JavaScript?
13. How many ways an HTML element can be accessed in JavaScript code?
14. What is JavaScript hoisting?
15. What is called Variable typing in JavaScript?

Topic: Introduction to PHP

Experiment No: 7, 8, 9

Aim:

7. Write a PHP script to take a number from the user and print the table of that number.

8. Design a shopping application form with following fields [itemID, itemName, itemQuantity] Write a PHP script to add and display the items.

9. Create the database table with the entries mentioned:
[rollNo, studName, studDept, passingYear, classGrades] Grades should be either {First Class, Second Class, Pass}

Design the UI as given below and show the appropriate result through PHP script

Objective: To learn about concept and implementation of JAVASCRIPT

Theory:

PHP stands for Hypertext Preprocessor. PHP is a very popular and widely-used open source server-side scripting language to write dynamically generated web pages. PHP was originally created by Rasmus Lerdorf in 1994. It was initially known as Personal Home Page.

PHP scripts are executed on the server and the result is sent to the web browser as plain HTML. PHP can be integrated with the number of popular databases, including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Sybase, and so on

Data Types in PHP

The values assigned to a PHP variable may be of different data types including simple string and numeric types to more complex data types like arrays and objects.

PHP supports a total eight primitive data types: Integer, Floating point number or Float, String, Booleans, Array, Object, resource and NULL. These data types are used to construct variables. Now let's discuss each one of them in detail.

What is PHP Arrays?

Arrays are complex variables that allow us to store more than one value or a group of values under a single variable name.

Types of Arrays in PHP

There are three types of arrays that you can create. These are:

- **Indexed array/ Numeric array** — An array with a numeric key.
- **Associative array** — An array where each key has its own specific value.
- **Multidimensional array** — An array containing one or more arrays within itself.

PHP Database connection

Requirements: XAMPP web server procedure:

Start XAMPP server by starting Apache and MySQL.

Write PHP script for connecting to XAMPP.

Run it in the local browser.

Database is successfully created which is based on the PHP code.

In PHP, we can connect to the database using XAMPP web server by using the following path. "localhost/phpmyadmin"

Conclusion:

Output (Screenshots):

Frequently Asked Questions:

1. What is GET and POST method in PHP?
2. What are the characteristics of PHP variables?
3. What is the difference between "echo" and "print" in PHP?
4. What did the acronym PHP originally stand for?
5. What are some of the common applications of PHP?
6. What is the recommended PHP version?
7. Which programming language is PHP similar to?

Topic:

Introduction to JQuery Expe

Aim:

r

i

m

e

n

t

N

o

:

1

0

,

1

1

10. Write a JQuery code to give animation to the div on click of button.
11. Write a JQuery code to add dynamic element on page on click of button

Objective: To learn about concept and implementation of Jquery

Theory:

What is jQuery?

JQuery is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, it simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

Why to use jQuery?

Some of the key points that support the answer for why to use jQuery:

- It helps us to manipulate HTML and CSS
- It helps us to manipulate DOM (Document Object Model) elements
- Provides event methods to trigger and respond to an events on a html page such as mouse click, keypress etc.
- Implements AJAX calls.
- Using jQuery (JavaScript library) on HTML page**:** There are several ways to start using it on your website.
- Use the Google-hosted/Microsoft-hosted content delivery network (CDN) to include a version. Or
- Download it from official website jQuery.com and host it on your server or local filesystem.

Advantages:

- Wide range of plug-ins that allows developers to create plug-ins on top of the JavaScript library.
- Large development community.
- It is a lot easier to use compared to standard javascript and other javascript libraries.
- It lets users develop Ajax templates with ease. Ajax enables a sleeker interface where actions can be performed on pages without requiring the entire page to be reloaded.
- Being Light weight and powerful chaining capabilities makes it stronger.

Disadvantages:

- While jQuery has an impressive library in terms of quantity, depending on how much customization you require on your website. The functionality may be limited thus using raw javascript may be inevitable in some cases.
- The JQuery javascript file is required to run the commands, while the size of this file is relatively small (25-100KB depending on the server). It is still a strain on the client computer and maybe your web server as well if you intend to host the script on your own web server.

Conclusion:

Output (Screenshots):

Frequently Asked Questions:

1. What is the purpose of jQuery?
2. How do I select an item using class or ID?
3. How do I select elements when I already have a DOM element?
4. How do I test whether an element has a particular class?
5. How do I test whether an element exists?
6. How do I determine the state of a toggled element?
7. How do I disable/enable a form element?
8. How do I check/uncheck a checkbox input or radio button?
9. How do I get the text value of a selected option?
10. How do I pull a native DOM element from a jQuery object

Topic: Introduction to Bootstrap

Experiment No: 12

Aim:

Display a popup on mouse over of link and add two text boxes and button to that popup.

Objective: To learn about concept of Bootstrap

Theory:

What is Twitter Bootstrap?

Bootstrap is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

Why use Bootstrap?

Mobile first approach: Since Bootstrap 3, the framework consists of Mobile first styles throughout the entire library instead of in separate files.

Browser Support: It is supported by all popular browsers.

Easy to get started: With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has good documentation.

Responsive design: Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about responsive design in the chapter Bootstrap Responsive Design. Provides a clean and uniform solution for building an interface for developers. It contains beautiful and functional built-in components which are easy to customize. It also provides web-based customization. And best of all it is open source.

What Bootstrap Package Includes?

Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, background.

CSS: Bootstrap comes with features of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system.

Components: Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, popovers, and much more.

JavaScript Plugins: Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one.

Customize: You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

Bootstrap Grid System

What is a Grid?

In graphic design, a grid is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal) lines used to structure content. It is widely used to design layout and content structure in print design. In web design, it is a very effective method to create a consistent layout rapidly and effectively using HTML and CSS. To put it simple words grids in web design organize and structure content, makes websites easy to scan and reduces cognitive load on users.

What is a Bootstrap Grid System?

As put by the official documentation of Bootstrap for grid system:

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes predefined classes for easy layout options, as well as powerful mixins for generating more semantic layouts.

Let us understand the above statement. Bootstrap 3 is mobile first in the sense that the code for Bootstrap now starts by targeting smaller screens like mobile devices, tablets, and then "expands" components and grids for larger screens such as laptops, desktops.

Working of Bootstrap Grid System

- Grid systems are used for creating page layouts through a series of rows and columns that house your content.

Here's how the Bootstrap grid system works:

- Rows must be placed within a .container class for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For
- Example, three equal columns would use three .col-xs-4.

Bootstrap Modal Plugin

- A model is a child window that is layered over its parent window. Typically, the purpose is to display content from a separate source that can have some interaction without leaving the parent window. Child windows can provide information, interaction, or more.

- You can toggle the modal plugin's hidden content:
  Via data attributes: Set attribute data-toggle="modal" on a controller element, like a button or link, along with a data-target="#identifier" or href="#identifier" to target a specific modal (with the id="identifier") to toggle.
- Via JavaScript: Using this technique you can call a modal with id="identifier" with a single

line of JavaScript:

- $('#identifier').modal(options)

Conclusion:

Output (Screenshots):

Frequently Asked Questions:

1. What is Bootstrap used for?
2. Can Bootstrap be used commercially?
3. What are the advantages of Bootstrap?
4. What is a Bootstrap Container, and how does it work?
5. What do you know about the Bootstrap Grid System?
6. In Bootstrap, what are the two codes for displaying code?

Topic:

Introduction to JQuery

Aim

Experiment No

o

:

1

3

Create a form that allows you to select a locale and displays the date in a manner appropriate to the locale using JSP.

Objective: To learn about concept of JSP

Theory:

JSP technology is used to create web applications just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet:
   JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP that makes JSP development easy.
2) Easy to maintain:
   JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.
3) Fast Development:
   No need to recompile and redeploy If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.
4) Less code than Servlet

   In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

   The Lifecycle of a JSP Page

- The JSP pages follow these phases:
- Translation of JSP Page

- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

Creating your first JSP page

```
<html>
<head>
<title>My first JSP page
</title>
</head>
<body>
<%@ page language=”java” %>

<% System.out.println(“Hello World”); %>
</body>
</html>
```

Type the code above into a text file. Name the file helloworld.jsp.
Place this in the correct directory on your JSP web server and call it via your browser.

Using JSP tags

There are five main tags:

1. Declaration tag
2. Expression tag
3. Directive Tag
4. Scriptlet tag
5. Action tag

Declaration tag (<%! %>)

This tag allows the developer to declare variables or methods. Before the declaration you must have <%!
At the end of the declaration, the developer must have %> Code place in this tag must end in a semicolon (;).
Declarations do not generate output so are used with JSP expressions or scriptlets.

For Example,

Expression tag (<%= %>)

This tag allows the developer to embed any Java expression and is short for out.println().

A semicolon (;) does not appear at the end of the code inside the tag. For example, to show the current date and time.

Directive tag (<%@ directive … %>)

A JSP directive gives special information about the page to the JSP Engine. There are three main types of directives:

1) Page – processing information for this page.
2) Include – files to be included.
3) Tag library – tag library to be used in this page.

Directives do not produce any visible output when the page is requested but change the way the JSP Engine processes the page.
For example, you can make session data unavailable to a page by setting a page directive (session) to false.

1. Page directive

This directive has 11 optional attributes that provide the JSP Engine with special processing information. The following table lists the 11 different attributes with a brief description:

language	Which language the file uses.	<%@ page language = "java" %>
extends	Superclass used by the JSP engine for the translated Servlet.	<%@ page extends = "com.taglib…" %>
import	Import all the classes in a java package into the current JSP page. This allows the JSP page to use other java classes. The following packages are implicitly imported. java.lang.* javax.servlet.* javax.servlet.jsp.* javax.servlet.http.*	<%@ page import = "java.util.*" %>
session	Does the page make use of sessions? By default all JSP pages have session data available. There are performance benefits to switching session to false.	Default is set to true.
buffer	Controls the use of buffered output for a JSP page. Default is 8kb	<%@ page buffer = "none" %>
autoFlush	Flush output buffer when full.	<%@ page autoFlush = "true" %>
isThreadSafe	Can the generated Servlet deal with multiple requests? If true a new thread is started so requests are handled simultaneously.	

Date : <%= new java.util.Date() %>

info	Developer uses info attribute to add information/document for a page. Typically used to add author, version, copyright and date info.	
errorPage	Different page to deal with	<%@ page errorPage =
	errors. Must be URL to error page.	"/error/error.jsp" %>
IsErrorPage	This flag is set to true to make a JSP page a special Error Page. This page has access to the implicit object exception (see later).	
contentType	Set the mime type and character set of the JSP.	

2. Include directive

Allows a JSP developer to include contents of a file inside another. Typically include files are used for navigation, tables, headers and footers that are common to multiple pages.

Two examples of using include files:

This includes the html from privacy.html found in the include current jsp page.

```
<%@ include file = "include/privacy.html %>
```

or to include a navigation menu (jsp file) found in the current

```
<%@ include file = "navigation.jsp %>
```

3. Tag Lib directive

A tag lib is a collection of custom tags that can be used by the page.

```
<%@ taglib uri = "tag library URI" prefix = "tag Prefix" %>
```

Custom tags were introduced in JSP 1.1 ar
side code from web designers.

Scriptlet tag (<% … %>)

Between <% and %> tags, any valid Java c
variable or bean declared.
For example, to print a variable.

Frequently Asked Questions:

1. List out some advantages of using JSP.
2. Give the syntax for JSP comments.
3. What is the difference between hide comment and output comment?
4. Is JSP technology extensible?
5. How can I implement a thread-safe JSP page? What are the advantages and Disadvantages of using it?
6. How can we handle the exceptions in JSP?
7. What are the two ways to include the result of another page?