Orchid International College

Bijaychowk, Gaushala

Department Of Information Technology

LAB MANUAL

WEB TECHNOLOGY

BIM , BSc.CSIT , BCA

WEB TECHNOLOGY I

Objective

1. To gain knowledge on designing static web pages.
2. Able to validate web pages at client-side.
3. Design and validate XML documents.
4. Understand, analyze and create XML documents and XML Schema.
5. To develop a functioning website with all the concepts of HTML , CSS and JavaScript.
6. Use JavaScript & PHP to validate form input entry.

Hardware and Software Required:

- A working computer system with any operating system.
- A web browser.
- An XML editor .

Outcomes:

- Create web pages using HTML and Cascading Styles sheets
- Analyze a web page and identify its elements and attributes
- Create XML documents and XML Schema
- Design with good aesthetic sense of designing .

Markup Language

1. Introduction To HTML
2. Basic HTML Structure
3. Working with Text and Headings
4. Formatting Text
5. Creating Lists
6. Working with Links
7. Adding Images
8. Creating Tables
9. Creating Forms
10. Working with Frames
11. Adding Multimedia
12. Sectioning Elements
13. Attributes

1. **Introduction to HTML:**
   - **Objective**: Understand the purpose and basics of HTML.
   - **Activities:**
     - Explore various websites and identify HTML elements.
     - Discuss the role of HTML in web development.
     - Briefly introduce tags, attributes, and the document structure.
2. **Basic HTML Structure:**
   - **Objective**: Learn the basic structure of an HTML document.
   - **Activities**:
     - Create an HTML document using a text editor.
     - Understand the DOCTYPE declaration, HTML, head, and body tags.
     - Add a title to the document.
3. **Working with Text and Headings:**
   - **Objective**: Use HTML tags to display text and headings.
   - **Activities**:
     - Add paragraphs and headings to the HTML document.
     - Experiment with different heading tags (h1, h2, h3, etc.).
     - Apply basic text formatting such as bold and italic.
4. **Formatting Text:**
   1. **Objective**: Learn how to format text in HTML.
   2. **Activities**:
      - Apply text formatting using tags like , , <u>, etc.
      - Create superscript and subscript text.
      - Experiment with line breaks and horizontal rules.
5. **Creating Lists:**
   - **Objective**: Create ordered and unordered lists..
   - **Activities**:
     - Create an unordered list with bullet points.
     - Create an ordered list with different numbering styles.
     - Nest lists within each other.

6. **Working with Links:**
   - **Objective**: Add hyperlinks to web pages.
   - **Activities**:
     - Create a basic hyperlink to an external website.
     - Add internal links within the same HTML document.
     - Explore various link attributes such as target and title.
7. **Adding Images:**
   - **Objective**: Insert images into an HTML document.
   - **Activities**:
     - Download an image from the web.
     - Add the image using the <img> tag and specify attributes like alt, width, and height.
     - Experiment with aligning and linking images.
     - Experiment with image map.
8. **Creating Tables:**
   - **Objective**: Construct tables to organize data.
   - **Activities**:
     - Create a simple table structure with rows and columns.
     - Add headers and data cells to the table.
     - Apply attributes like rowspan, colspan, and table borders.
9. **Creating Forms:**
   - **Objective**: Build interactive forms using HTML.
   - **Activities**:
     - Create a basic form with input fields, checkboxes, and radio buttons.
     - Use labels and form elements to improve accessibility.
     - Understand the form action attribute and submit button.
10. **Working with Frames:**
    - **Objective**: Understand the concept of frames in HTML.
    - **Activities**:
      - Create a frameset with multiple frames.
      - Define the source of each frame.

- Experiment with frame attributes such as scrolling and resizing.

11. Adding Multimedia:

- **Objective**: Embed multimedia content in an HTML document.
- **Activities**:
  - Add videos using the <video> tag and specify attributes like controls and autoplay.
  - Embed audio files using the <audio> tag.
  - Include external multimedia content, such as YouTube videos.

12. Sectioning Elements:

- **Objective**: Understand the concept of HTML5 sectioning elements.
- **Activities**:
  - Add HTML5 sectioning elements like <article> , <aside> , <section>, <nav> , <header> and <footer>.

13. Global Attributes:

- **Objective**: Learn about the global attributes applicable to most HTML elements.
- **Activities**:
  - Introduce global attributes such as id, class, style, and title.
  - Create HTML elements and apply global attributes to modify their behavior.
  - Discuss the importance of using semantic and meaningful values for attributes.

.

Cascading Style Sheet

Objective:

The objective of this lab is to familiarize students with Cascading Style Sheets (CSS) and Responsive Web Design concepts. Students will learn about CSS syntax, different methods of inserting CSS, selectors, colors, backgrounds, borders, margins, paddings, box model, text formatting, links, lists, tables, positioning, responsive web design techniques, media queries, and popular responsive web design frameworks.

Lab Setup:

To complete this lab, you will need a text editor to write HTML and CSS code. You can use any text editor of your choice, such as Notepad++, Sublime Text, Visual Studio Code, or any online HTML/CSS editor.

Lab Tasks:

Task 1: CSS Syntax

1. Create a new HTML file and link a CSS file to it using an external CSS file.

2. Add some text content to the HTML file.

3. In the CSS file, apply the following styles to the HTML content:

   a. Set the font size to 18 pixels.

   b. Change the text color to red.

   c. Apply a background color of light gray.

Task 2: Inserting CSS

1. Modify the HTML file created in Task 1 to include internal CSS.

2. Inside the `<head>` section of the HTML file, add a `<style>` tag and write CSS rules to change the font family and text alignment.

Task 3: Comments and Selectors

1. Add comments in your CSS file to describe the purpose of the styles.

2. Create a new HTML file and apply the following styles using various selectors:

   a. Apply a background color of yellow to an element with the ID "myId".

   b. Change the font color to blue for all elements with the class "myClass".

   c. Apply bold font weight to all `<h1>` headings.

Task 4: Combinators , Pseudo-classes, Pseudo-elements, and Attribute Selectors

1. Create a new HTML file and style the following elements using CSS:

   a. Select all `<p>` elements that are direct children of `<div>` elements.

   b. Style the first letter of each paragraph with a larger font size and a different color.

   c. Apply a custom style to all links with a `target="_blank"` attribute.

   d. Apply the concept of Universal Selector, The Type Selector, The Child Selector, The Descendant Selector, The Adjacent Sibling Selector.

Task 5: Colors, Backgrounds, Borders, Margins, and Paddings

1. Create a new HTML file and apply different styles to demonstrate the following CSS properties:

   a. Set a background color for an element.

   b. Add a border with a specific width, style, and color.

   c. Set margins and paddings for an element.

Task 6: Height/Width, Box Model, and Text Formatting

1. Create a new HTML file and demonstrate the following CSS properties:

   a. Set a fixed height and width for an element.

   b. Understand the box model and apply different styles to the content, padding, border, and margin of an element.

   c. Modify the text formatting using CSS properties such as text-align, text-decoration, and text-transform.

Task 7: Box Shadows , Text Effects and Shadows

1. Create a new HTML file and demonstrate the following CSS properties:

   a. Apply the concept of CSS Box Model in div .

   b. Apply the concept of Text-shadow in paragraph.

Task 8: Links, Lists, and Tables

1. Create a new HTML file and apply different styles to links, lists, and tables using CSS.

   a. Customize link styles (normal, hover, visited, active).

   b. Modify the appearance of unordered and ordered lists.

   c. Style table elements, including table headers, rows, and cells.

Task 9: Position, Overflow, Float, and Align

1. Create a new HTML file and demonstrate the following CSS properties:

   a. Position elements using different values of the "position" property (static, relative, absolute, fixed).

   b. Apply overflow and float properties to control the content and layout of elements.

c. Align elements using CSS properties like text-align, vertical-align, and margin.

Task 10: Forms and Responsive Web Design

1. Create a new HTML file with a form containing various form elements.

2. Style the form elements using CSS to enhance the visual appearance.

3. Implement responsive web design techniques to make the form adapt to different screen sizes.

Task 11: Media Queries and Responsive Web Design Frameworks

1. Create a new HTML file and incorporate media queries to make the web page responsive.

2. Research and choose a responsive web design framework (e.g., Bootstrap, Foundation) and apply it to the HTML file.

3. Customize the responsive framework's components to match the design requirements of your web page.

Client-Side Scripting

Table of Contents:

- Assignment Operators
- Comparison Operators
- Logical Operators

7. Control Statements
   - Conditional Statements (if, else if, else)
   - Switch Statement
   - Looping Statements (for, while, do-while)
   - Break and Continue Statements

8. Functions
   - Function Declaration
   - Function Expressions
   - Function Parameters and Return Values
   - Anonymous Functions
   - Arrow Functions

9. Arrays
   - Creating Arrays
   - Accessing Array Elements
   - Modifying Arrays
   - Array Methods
   - Iterating over Arrays

10. Classes and Objects
    - Object-Oriented Programming (OOP) in JavaScript
    - Creating Classes and Objects
    - Constructor Functions
    - Prototypes and Inheritance

11. Built-in Objects
    - String Object
    - Math Object
    - Date Object
    - Array Object
    - Regular Expression Object
12. Event Handling and Form Validation
    - Handling Events in JavaScript
    - Event Listeners
    - Form Validation Techniques
    - Common Form Validation Patterns
13. Error Handling
    - Error Types in JavaScript
    - try...catch Statement
    - Throwing Custom Errors
    - Error Handling Best Practices
14. Handling Cookies
    - Introduction to Cookies
    - Creating and Reading Cookies
    - Modifying and Deleting Cookies
    - Cookie Best Practices
15. Document Object Model (DOM)
    - Introduction to DOM
    - DOM Manipulation
    - Accessing DOM Elements

- Modifying DOM Elements
- Creating and Removing Elements

16. Browser Object Model (BOM)
    - Introduction to BOM
    - Window Object
    - Location Object
    - Navigator Object
    - History Object
17. Basics of jQuery, React, and AngularJS
    - Overview of jQuery
    - Introduction to React
    - Introduction to AngularJS
18. AJAX and JSON
    - Asynchronous JavaScript and XML (AJAX)
    - Making AJAX Requests
    - JSON (JavaScript Object Notation)
    - Parsing and Generating JSON

Objective:

The objective of this lab is to familiarize students with JavaScript , jQuery , React and AngularJS , AJAX and JSON.

Lab Setup:

To complete this lab, you will need a text editor to write HTML and JavaScript code. You can use any text editor of your choice, such as Notepad++, Sublime Text, Visual Studio Code, or any editor of your choice.

Lab 1: Introduction to Client-Side Scripting

Objective: Understand the basics of client-side scripting and its advantages.

1. Read the provided material on client-side scripting.
2. Discuss the advantages of client-side scripting compared to server-side scripting.
3. Write a short summary highlighting the key points of client-side scripting.

Lab 2: Adding JavaScript to a Page

Objective: Learn different methods to add JavaScript code to an HTML page.

1. Create an HTML file.
2. Add an inline JavaScript code block within the HTML file.
3. Create an external JavaScript file and link it to the HTML file.
4. Use the "defer" and "async" attributes when linking an external JavaScript file and observe the differences.

Lab 3: Output in JavaScript

Objective: Explore different ways to generate output using JavaScript.

1. Use console.log() to output messages to the browser console.
2. Display alert boxes with appropriate messages.
3. Write JavaScript code to dynamically update the content of HTML elements.

Lab 4: Comments in JavaScript

Objective: Understand the importance of comments and their usage in JavaScript.

1. Add single-line comments to your JavaScript code, explaining the purpose of each line.
2. Add multi-line comments to describe the functionality of a specific block of code.

3. Review your code and ensure that comments follow best practices for readability and clarity.

Lab 5: Variables and Data Types

Objective: Learn how to declare variables and understand different data types in JavaScript.

1. Declare variables using various data types: string, number, boolean, array, and object.
2. Experiment with variable scoping (global vs. local variables).
3. Perform type conversions using built-in JavaScript functions.

Lab 6: Operators

Objective: Practice using different operators in JavaScript.

1. Create variables and perform arithmetic operations (addition, subtraction, multiplication, division) using arithmetic operators.
2. Use assignment operators to update the value of variables.
3. Compare values using comparison operators and combine conditions using logical operators.

Lab 7: Control Statements

Objective: Understand how to control the flow of execution using control statements.

1. Write if statements to perform conditional branching based on a given condition.
2. Implement a switch statement to handle different cases.
3. Use loop statements (for, while, do-while) to iterate over arrays or perform repeated actions.

Lab 8: Functions

Objective: Learn how to define and use functions in JavaScript.

1. Create a function that takes parameters and returns a value.
2. Implement an anonymous function and assign it to a variable.
3. Explore arrow functions and their concise syntax.

Lab 9: Arrays

Objective: Understand the concepts related to arrays and practice using array methods.

1. Create an array and access its elements using indexes.
2. Use array methods (push, pop, shift, unshift, slice, splice) to modify the array.
3. Iterate over the array using different loop constructs.

Lab 10: Classes and Objects

Objective: Explore object-oriented programming (OOP) concepts in JavaScript.

1. Create a class with properties and methods.
2. Instantiate objects from the class and manipulate their properties and methods.
3. Implement inheritance using prototypes.

Lab 11 : Built-in objects

Objective: Get familiar with the built-in objects in JavaScript and their usage.

1. Use string object methods to manipulate strings.
2. Perform mathematical calculations using the Math object.
3. Work with dates using the Date object.
4. Explore array methods to manipulate arrays efficiently.
5. Apply regular expressions using the RegExp object.

Lab 12: Event Handling and Form Validation

Objective: Learn how to handle events and perform form validation using JavaScript.

1. Create an HTML form and handle its submission event.
2. Implement form validation using JavaScript to ensure correct user input.
3. Display appropriate error messages for invalid form entries.

Lab 13: Error Handling

Objective: Understand the importance of error handling and practice handling exceptions.

1. Write JavaScript code that intentionally generates errors.
2. Implement try...catch statements to handle exceptions gracefully.
3. Throw custom errors and handle them appropriately.

Lab 14: Handling Cookies

Objective: Learn how to work with cookies in JavaScript.

1. Understand the concept of cookies and their usage.
2. Write JavaScript code to create, read, modify, and delete cookies.
3. Implement best practices for handling cookies, such as setting expiration dates and secure flags.

Lab 15: Document Object Model (DOM)

Objective: Manipulate the Document Object Model (DOM) using JavaScript.

1. Access and modify HTML elements using JavaScript.

2. Create new elements and append them to the DOM.
3. Remove elements from the DOM dynamically.

Lab 16: Browser Object Model (BOM)

Objective: Understand the Browser Object Model (BOM) and its usage.

1. Access and manipulate the window object properties.
2. Use the location object to get information about the URL.
3. Explore the navigator object to retrieve browser-related information.
4. Manipulate the browser's history using the history object.

Lab 17: Basics of jQuery, React, and AngularJS

Objective: Get an overview of popular JavaScript libraries and frameworks.

1. Study the basics of jQuery and its features.
2. jQuery selectors: element, id, class
3. jQuery events: mouse, keyboard, form, document/window
4. jQuery effects: hide/show, fade, slide, animate, stop, callback, chaining
5. Explore the fundamental concepts of React and its component-based architecture.
6. Understand the key concepts of AngularJS and its two-way data binding.

Lab 18: AJAX and JSON

Objective: Learn about AJAX and JSON and how to work with them.

1. Make asynchronous requests to a server using AJAX.
2. Retrieve data from a server and update the HTML page dynamically.
3. Parse JSON data and generate JSON objects using JavaScript.

XML and XML Technologies

Objective: The objective of this lab is to introduce you to XML (eXtensible Markup Language) and various XML technologies such as XPath, XSLT, and XQuery. By the end of this lab, you should be able to understand XML syntax, create XML documents, define elements and attributes, work with namespaces, validate XML using DTD and Schema, and perform basic operations using XPath, XSLT, and XQuery.

Prerequisites:

- Basic knowledge of HTML and CSS.

- A text editor and a web browser.

Lab Setup:

1. Open your preferred text editor.

2. Create a new folder for the lab exercises.

3. Within the folder, create separate files for each exercise and save them with the appropriate file extensions (.xml, .xsl, .xq, etc.).

Lab Exercises:

Exercise 1: Introduction to XML

1. Create a new XML file and save it with a ".xml" extension.
2. Define the XML declaration at the beginning of the file:
   <?xml version="1.0" encoding="UTF-8"?>
3. Create an XML document structure with root and nested elements, including attributes.
4. Save the XML file and open it in a web browser or XML viewer to verify its structure.

Exercise 2: Elements and Attributes

1. Open the XML file you created in the previous exercise.
2. Add elements and attributes to the XML document, representing different data types and their corresponding values.
3. Ensure proper nesting and hierarchy of elements.
4. Save the XML file and validate its structure by opening it in a web browser or XML editor.

Exercise 3: Namespace

1. Open the XML file you created in the previous exercise.
2. Introduce a namespace to the XML document by declaring a namespace prefix and using it in element names.
3. Define a namespace URI and associate it with the namespace prefix.
4. Add elements within the namespace and ensure they are properly qualified.
5. Save the XML file and validate its structure.

Exercise 4: DTD and Schema

1. Create a new XML file and save it with a ".xml" extension.
2. Define a Document Type Definition (DTD) or XML Schema to specify the structure and data types of the XML document.
3. Create XML schema definitions for simple types.
4. Define complex types and their relationships in XML schema
5. Reference the DTD or Schema in the XML file using the appropriate declaration.
6. Add elements and attributes to the XML document, ensuring they comply with the defined DTD or Schema.
7. Save the XML file and validate its structure against the DTD or Schema.

Exercise 5: XSD Attributes, Default and Fixed Values, Facets:

1. Define attributes in XML schema.
2. Set default and fixed values for attributes.
3. Apply facets to constrain XML data.

Exercise 6 : Use of Patterns, Order Indicators (All, Choice, Sequences), Occurrence Indicators (MaxOccurs, MinOccurs):

1. Define patterns in XML schema for validating data.
2. Experiment with different order indicators for elements.
3. Use occurrence indicators to control element occurrence.

Exercise 7: Introduction to XPath

1. Create an XML file with multiple elements and save it with a ".xml" extension.
2. Open the XML file and write XPath expressions to select specific elements or attributes within the XML document.
3. Use different XPath axes and predicates to refine the XPath expressions.
4. Test the XPath expressions using an XPath evaluator or XPath query tool.

Exercise 8: Introduction to XSLT

1. Create an XML file and a separate XSLT file, both saved with the appropriate file extensions (.xml and .xsl).

2. Define an XSLT template to transform the XML document into a different format or structure.
3. Use XSLT instructions and functions to apply transformations, create new elements, modify values, or remove elements.
4. Apply the XSLT transformation to the XML file using a web browser or XSLT processor.
5. Verify that the XML document is transformed according to the XSLT instructions.

Exercise 9: Introduction to XQuery

1. Create an XML file and a separate XQuery file, both saved with the appropriate file extensions (.xml and .xq).
2. Write XQuery expressions to query and retrieve data from the XML document based on specific criteria.
3. Use XQuery functions and operators to perform data manipulation or filtering.
4. Execute the XQuery file using an XQuery processor or integrated development environment (IDE).
5. Verify that the desired data is extracted from the XML document based on the XQuery expressions.

Exercise 10: SAX: DOM, Creating XML Parser

1. Compare SAX and DOM parsing approaches.
2. Implement an XML parser using SAX.
3. Parse XML documents and extract relevant data using SAX.

SERVER SIDE SCRIPTING LANGUAGE

Objective

1. To gain knowledge on developing dynamic web pages.
2. Able to validate web pages at server-side.
3. To be able to communicate with the database.
4. To prepare a fully functional project at the end of the course

Hardware and Software Requirements:

- A working computer system with either Windows or Linux
- A web browser .
- Apache web server.
- A database either Mysql or Oracle.

Introduction

Lab 1: Setting Up PHP Environment

- Installing PHP and a web server (e.g., Apache)

- Verifying PHP installation

- Running a simple PHP script

Lab 2: Introduction to PHP

- Exploring PHP's role in web development

- Writing and executing basic PHP programs

- Understanding PHP tags and syntax

- Printing output with echo and print

Lab 3: PHP in Action

- Building a simple dynamic webpage with PHP

- Embedding PHP code within HTML

- Manipulating HTML elements using PHP

Lab 4: Basic Rules of PHP Programs

- Understanding PHP variables and data types

- Declaring and initializing variables
- Applying naming conventions and best practices
- Writing comments in PHP code

Working with Text and Numbers

Lab 5: Text Manipulation in PHP

- Concatenating strings
- Using string functions (e.g., strlen, substr)
- Formatting strings and output

Lab 6: Number Operations in PHP

- Performing arithmetic operations
- Using math functions (e.g., round, rand)
- Formatting numbers and output

Lab 7: Variables in PHP

- Creating and assigning values to variables
- Variable scope and lifetime
- Variable interpolation and concatenation

Making Decisions and Repeating Yourself

Lab 8: Understanding true and false

- Using boolean data type
- Comparing values with comparison operators
- Applying logical operators (e.g., AND, OR, NOT)

Lab 9: Making Decisions with if...else Statements

- Writing conditional statements
- Executing different code paths based on conditions

Lab 10: Building Complicated Decisions

- Using nested if statements

- Implementing switch...case statements

- Choosing appropriate decision structures

Lab 11: Repeating Yourself with Loops

- Working with while, do...while, and for loops

- Controlling loop execution (e.g., break, continue)

- Looping through arrays and performing operations

Working with Arrays

Lab 12: Array Basics

- Creating and initializing arrays

- Accessing array elements

- Manipulating arrays (e.g., adding, modifying, deleting elements)

Lab 13: Looping Through Arrays

- Iterating over arrays using for and foreach loops

- Performing operations on array elements

Lab 14: Modifying Arrays

- Modifying array elements and structures

- Using array functions (e.g., array_push, array_merge)

Lab 15: Sorting Arrays

- Sorting arrays in ascending and descending order

- Custom sorting with user-defined functions

Lab 16: Using Multidimensional Arrays

- Creating and working with multidimensional arrays

- Accessing and modifying nested array elements

Functions

Lab 17: Declaring and Calling Functions

- Creating user-defined functions
- Passing arguments to functions
- Invoking functions with different argument types

Lab 18: Returning Values from Functions

- Defining return values in functions
- Accessing and utilizing function return values

Lab 19: Understanding Variable Scope

- Understanding local and global variables
- Scope hierarchy and variable visibility
- Utilizing static variables in functions

Web Forms

Lab 20: Server Variables

- Accessing and utilizing server variables in PHP
- Understanding their role in form processing

Lab 21: Accessing Form Parameters

- Retrieving form input using the $_POST and $_GET superglobals
- Handling form data securely

Lab 22: Form Processing with Functions

- Implementing form validation and processing functions
- Sanitizing and validating user input

Lab 23: Displaying Default Values

- Setting default values for form fields
- Populating form fields with data

Lab 24: Putting It All Together

- Building a complete web form with form processing logic

Working with Databases

Lab 25: Organizing Data in a Database

- Understanding database concepts (tables, records, fields)

- Designing a database schema for a specific scenario

Lab 26: Connecting to a Database Program

- Establishing a connection to a MySQL database using PHP

- Handling connection errors

Lab 27: Creating a Table

- Creating database tables using PHP and SQL statements

Lab 28: Putting Data into the Database

- Inserting data into database tables

- Handling database insertion errors

Lab 29: Retrieving Data from the Database

- Querying and fetching data from database tables

- Displaying retrieved data

Lab 30: Changing the Format of Retrieved Rows

- Manipulating and formatting database query results

Lab 31: Retrieving Form Data Safely

- Implementing secure database queries with parameter binding

Lab 32: A Complete Data Retrieval Form

- Creating a web form to retrieve and display data from the database

Lab 33: MySQL Aggregate Functions

- Creating a query which includes aggregate functions like sum , avg ,count etc.

- Create a query to implement the concept of Order By and Group By Clause

- Create a query to implement the concept of MySQL Subqueries and MySQL Joins.

Cookies and Sessions

Lab 34: Working with Cookies

- Setting and retrieving cookies in PHP

- Implementing cookie-based functionality

Lab 35: Activating Sessions

- Starting and managing PHP sessions

- Configuring session-related settings

Lab 36: Storing and Retrieving Information

- Storing and retrieving data in PHP sessions

- Utilizing session variables

Lab 37: Configuring Sessions

- Modifying session-related configurations and behavior

Lab 38: Login and User Identification

- Implementing a simple login system using sessions

Lab 39: Why setcookie( ) and session_start( ) Want to Be at the Top of the Page

- Understanding the importance of proper cookie and session handling in PHP

Dates and Times

Lab 40: Displaying the Date or Time

- Retrieving and displaying the current date and time

- Formatting date and time strings

Lab 41: Parsing a Date or Time

- Converting date and time strings to PHP DateTime objects

- Manipulating and formatting DateTime objects

Lab 42: Dates and Times in Forms

- Handling date and time input from web forms

- Validating and processing date and time data

Lab 43: Displaying a Calendar

- Building a simple calendar application using PHP and HTML

Working with Files

Lab 44: Understanding File Permissions

- Understanding file permissions and their significance

- Setting file permissions in PHP

Lab 45: Reading and Writing Entire Files

- Reading file contents into

Lab 46: Reading and Writing Parts of Files

- Reading specific sections of files

- Appending data to existing files

Lab 47: Working with CSV Files

- Reading and parsing CSV files in PHP

- Writing data to CSV files

Lab 48: Inspecting File Permissions

- Checking file permissions and handling errors

Lab 49: Sanitizing Externally Supplied Filenames

- Validating and sanitizing filenames to prevent security vulnerabilities

Advanced Server Side Scripting Language

Lab 50: Object-Oriented Programming in PHP

1. Introduction to Object-Oriented Programming (OOP)

- Basic concepts of OOP

2. Classes and Objects

- Defining a class
- Creating objects
- Accessing properties and methods

3. Properties and Methods

- Defining properties
- Access modifiers (public, private, protected)
- Defining methods

4. **Constructors and Destructors**

- Using constructors to initialize objects
- Destructor and its purpose

5. **Method Overriding**

- Overriding methods in subclasses
- Understanding inheritance and polymorphism

6. **Encapsulation**

- Accessors and mutators (getters and setters)
- Encapsulating data within classes

7. **Inheritance**

- Creating subclasses
- Extending functionality from a parent class

8. **Polymorphism**

- Polymorphic behavior in PHP
- Using interfaces and abstract classes

9. **Static Members**

- Defining static properties and methods

- Accessing static members

10. Exception Handling

- Handling exceptions in PHP
- Using try-catch blocks

Lab 51: AJAX (Asynchronous JavaScript and XML)

1. Introduction to AJAX

- AJAX request/response model

2. Using PHP with AJAX

- Sending AJAX requests to a PHP script
- Handling AJAX responses in PHP

3. Using PHP with MySQL

- Connecting to a MySQL database using PHP
- Performing CRUD operations with AJAX

Lab 52: jQuery

1. Playing With Elements

- Selecting elements using jQuery selectors
- Manipulating element content, attributes, and styles

2. Hiding and Unhiding Images

- Showing and hiding elements with animations
- Creating image sliders with jQuery

3. jQuery UI

- Introduction to jQuery UI library
- Implementing UI components (e.g., datepicker, tabs)

Lab 53: Joomla

1. Installation

- Installing Joomla on a local server or web host
- Configuring the database and site settings

2. Handling Joomla Back End

- Navigating the Joomla administrator interface
- Managing articles, categories, and modules

3. Customization in Joomla

- Changing the site's appearance with templates
- Installing and configuring extensions (plugins, modules)

4. Template Development in Joomla

- Creating a custom Joomla template
- Modifying template files and layout

5. Artisteer (IDE)

- Using Artisteer for visual Joomla template design
- Exporting and installing Artisteer templates

6. Module Development in Joomla

- Creating custom modules in Joomla
- Implementing module parameters and functionality

7. Component Development in Joomla

- Creating custom components in Joomla
- Creating backend and frontend functionality

8. Introduction to MVC (Model, View, Controller)

- Understanding the MVC architectural pattern
- Applying MVC in Joomla component development

Lab 5: WordPress Administrator Level

1. Theme Integration

- Installing and activating WordPress themes
- Customizing theme settings and appearance

2. Creating Pages

- Creating and managing static pages in WordPress
- Setting up page templates and hierarchy

3. Managing Widgets

- Adding and managing widgets in WordPress
- Customizing widget areas and layouts