SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

MODERN WEB APPLICATIONS

Course Code: INT316

Semester: V

Lab Manual

2024

SHANMUGHA ARTS, SCIENCE, TECHNOLOGY AND RESEARCH ACADEMY
(SASTRA Deemed to be) University
Tirumalaisamudram, Thanjavur-613 401
School of Computing

Course Objective:

This course will help the learner to be familiar with client server architecture and to develop a web based application by selecting appropriate technologies for client and server side programming along with database connectivity.

Course Learning Outcomes:

Upon successful completion of this course, the learner will be able to

- Understand the internet and client server architecture.
- Develop Interactive web pages by the use of HTML and CSS.
- Design dynamic web pages by integrating HTML with CSS and JavaScript.
- Implement programs in PHP to demonstrate basics of PHP.
- Apply PHP Form controls and manipulate the forms in web sites.
- Analyse to use appropriate client-side or server-side applications.
- Demonstrate the connection between a PHP programs to a MYSQL Database and perform insert, update and delete operations on the table.

List of Experiments:

1. **Basic HTML tags, Text formatting, table, link, list and images tags.**
2. **Forms and its components..**
3. **HTML with CSS (inline, internal and external style sheets).**
4. **CSS positional parameters and drop down styles to add aesthetic effects for your web page.**
5. **HTML Form components and validate the content using Java script (looping and conditional constructs and function).**
6. **Java Script components like JS alert, JS button, JS popover to understand the usage.**
7. **XML document with validation using internal and external Document Type Definition and transformation using XSLT**
8. **Design a PHP webpage for the purpose of validating the content of various page components.**
9. **Web site creation using PHP and provide necessary navigation among the web pages and validate the form content.**
10. **Design a Web Page and manipulating data in MySQL using PHP. Apply CRUD operations and provide necessary user authentication.**

Exercise No. 1 Basic HTML tags, Text formatting, table, link, list and images tags.

a) Generate an HTML file showcasing your timetable and incorporate the rowspan and colspan attributes of the table tag to effectively structure the layout. Familiarize yourself with the font tag (refer w3schools), and then experiment with various font styles and colors to enhance the visual appeal of the content in the HTML file.

b) Design a HTML file to present your resume effectively. Implement all text formatting tags, tables, and lists to organize and enhance the content within the document.

c) Develop a wildlife-themed website showcasing various animals and their details. Begin by crafting an index.html file with appropriate headings and image links leading to their respective pages. Generate three additional pages (e.g., tiger.html, lion.html, wolf.html) dedicated to specific animals. When users click on an image on the index.html page, the respective animal page should open. Utilize text formatting tags, font tags, list tags, table tags, and colors throughout the website to enhance the visual appeal and content presentation.

d) Design a banner using HTML tags to show an advertisement for DAKSH.

Objectives

The following set of exercises will help the learner to expertise the syntax of various HTML tags like text formatting, table and link, list and image tags and help the learner to develop a simple web page to understand how to improve the aesthetic effects of a web page.

Procedure / Algorithm

1. Use DOCTYPE to specify the html coding.
2. Use appropriate tags to add content on the web page.
3. Save the document with .html extension.
4. Open the HTML document on the browser by specifying complete path.
5. Do changes on the existing HTML file and refresh the browser to view the updated content.

Sample Input

The learner is expected to open the specified HTML document on the browser and change the existing content and or apply various possible tags in the HTML document. During execution, no inputs are required for the above exercise as it is a static web page creation.

Output

Designed web page will be displayed on the browser. Refresh the browser for every changes made in the HTML document.

Exercise No. 2 Forms and its components.

Design an online job portal with the following web pages : welcome page, application page and success page.

welcome page: display about job consultancy

application page: provision for entering applicant's details like first name, last name, age, date of birth, email etc., (refer the following sample web form screen shot) with submit button.

success page: On clicking the submit button in the form page, this success page should be displayed.

First Name: *

Last Name: *

Age: *

Date of Birth: 01 Jan 2015 *

Sex: Male Female *

Email: *

Mobile Number: *

Address1: *

Address2:

State: Tamilnadu * Country India *

Religion: Hindu Muslim Christian

No photo

Qualification

Examination Passed	Board/University	Year Of Completion	Mode Of Education	Percentage/Grade
SSLC				
Higher Secondary				
U.G	B.Sc		Full Time	
P.G	M.Sc		Full Time	

Industrial Preference

IT Mechanical Automobile Marketing R&D Chemical Education Bio-Technology

Location Preference

User Name: *

Password: *

Submit

Objectives

The learner will be able understand how to design a web form with various components like text box, radio button, check box, buttons, drop-down box etc. The learner can design various pages and navigate between the web pages.

Procedure / Algorithm

1. Use DOCTYPE to specify the html coding.
2. Use appropriate FORM component tags to design the welcome page, application page and success page.
3. Save all the documents with appropriate filename with .html extension.
4. Open the **welcome.html** document on the browser by specifying complete path.
5. Enter all the details in the input fields, selection box and option buttons.
6. Clicks submit button and check whether it moves on to success.html page.

Sample Input

Open the *welcome.html* page on the browser – no input required for this page

Enter the applicant's details on the appropriate fields in the *application.html* and click submit button

No input is required for success.html page

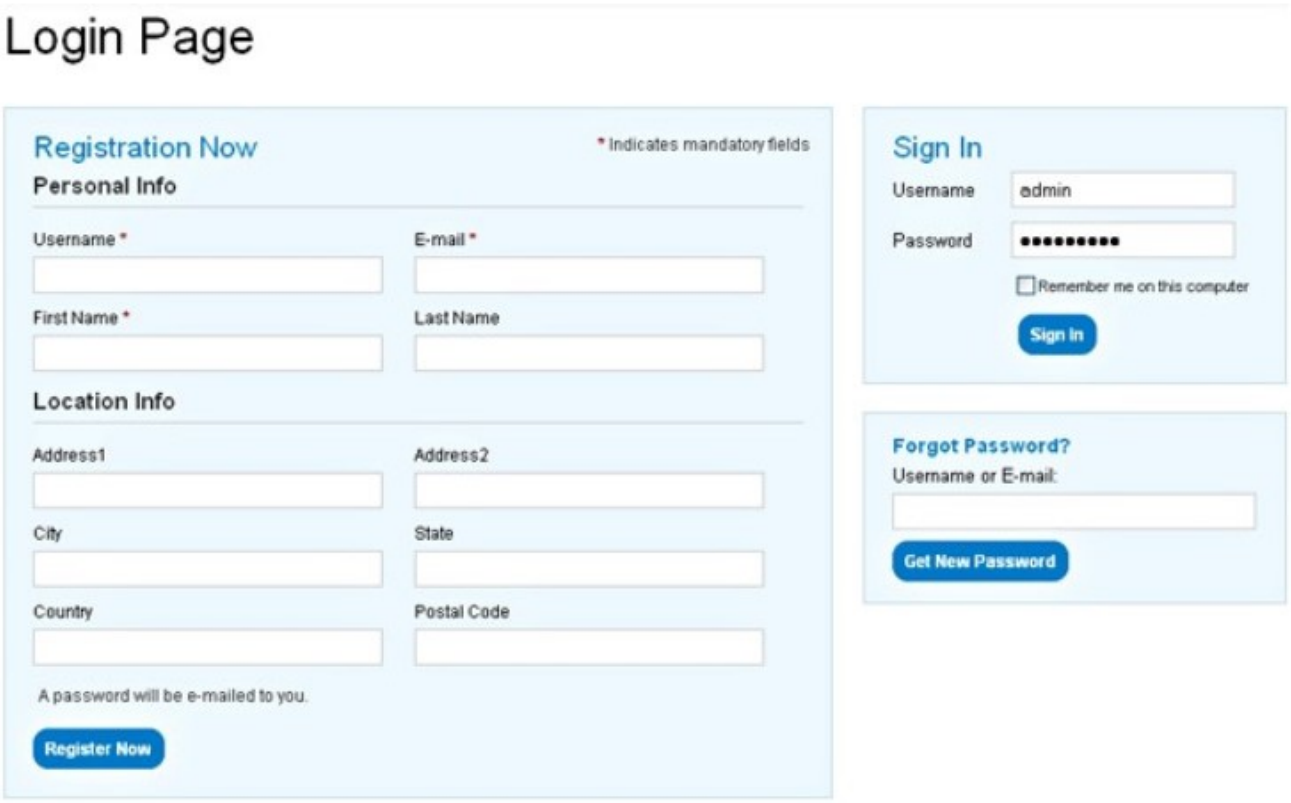
Output

The three pages are shown on the browser based on the user's input on the application.html page. For the valid values entered in the application.html page, it redirects the user to success.html page.

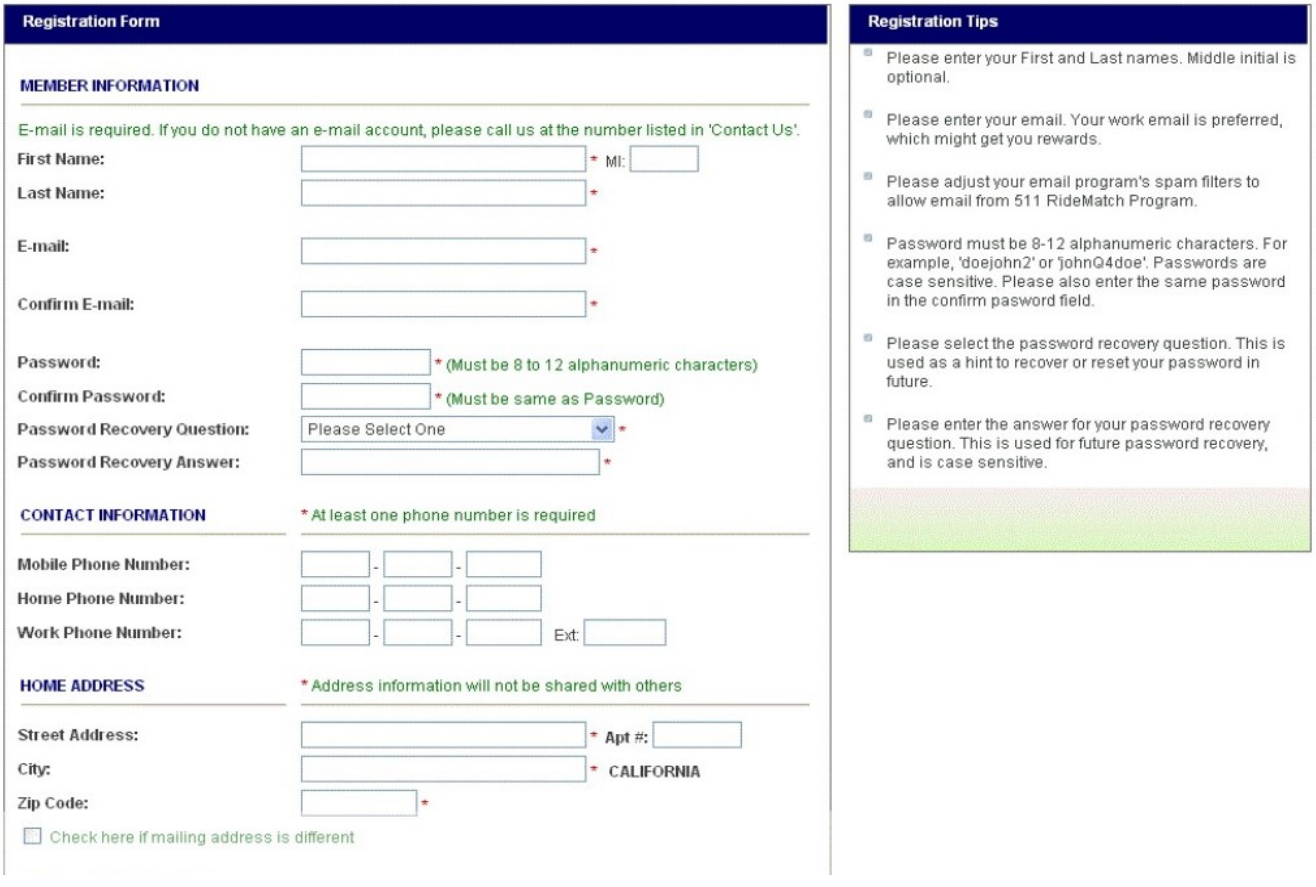
Exercise No. 3 HTML with CSS (inline, internal and external style sheets).

Design the following 3 web pages using HTML
Use External CSS to apply style for text boxes, buttons, headings, tables, paragraph etc., Alignment should be perfect and should be more attractive. Provide unique ID for the controls. Refer the following sample web pages.

Registration Page-1

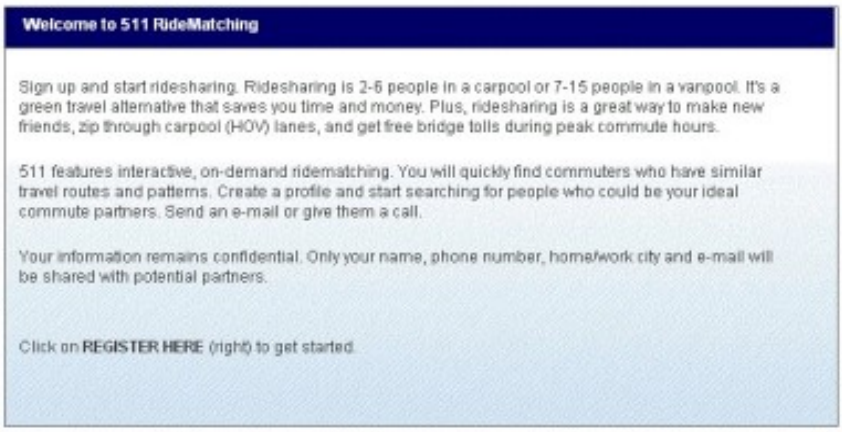


Registration Page-2




COMMUTE INFORMATION
Current Commute Mode:
Please select one
How did you hear about us?
Please select one
Work Schedule:
8 : 00 : AM To: 5 : 00 : PM
Flexible to Arrive or Leave:
30 Minutes
Commute Days:
Please select the days of your commute in a week
Mon. Tue. Wed. Thu. Fri. Sat. Sun.
DISCLAIMER
Terms of Use and Privacy Policy of 511.org.
* Required Fields
<< Clear
Continue >>

Home Page


Welcome to 511 RideMatching
Sign up and start ridesharing. Ridesharing is 2-6 people in a carpool or 7-15 people in a vanpool. It's a green travel alternative that saves you time and money. Plus, ridesharing is a great way to make new friends, zip through carpool (HOV) lanes, and get free bridge tolls during peak commute hours.
511 features interactive, on-demand ridematching. You will quickly find commuters who have similar travel routes and patterns. Create a profile and start searching for people who could be your ideal commute partners. Send an e-mail or give them a call.
Your information remains confidential. Only your name, phone number, home/work city and e-mail will be shared with potential partners.
Click on REGISTER HERE (right) to get started.


RideMatch Log On
First Time User?
Register Here >>
Returning User? Please log on using your registered e-mail and password below.
E-mail:
Password:
Forgot Password?
Log On >>


Tips
Ridesharing is flexible. Once a week or every day.
Contact everyone on your Matchlist. Call or send an e-mail.
Before starting, meet potential partners and discuss details.
Talk about who will alternate driving.
Determine your route and schedule.
Discuss insurance, parking, emergencies, illness and vacations.
Establish rules for cell phones, perfume, eating, etc.

Objectives

The learner will be able to design a web page using forms and frameset (iframe) with inline, internal and external style sheets. Improving the aesthetic effects of the website by adding various styles on the existing HTML tags.

Procedure / Algorithm

1. Create the above HTML documents with necessary form components. Use DOCTYPE to specify the html coding.
2. Use appropriate FORM component tags to design the Home page, Login page, sign in , Forgot password page, and tips page.
3. Save all the documents with appropriate filename with .html extension.
4. Create a HTML document with frameset (iframe in HTML5). Include necessary frame to load relevant html file (eg: registration page, login page, sign in, forgot password etc.,)
5. Enter all the details in the input fields, selection box and option buttons.
6. Clicks submit button and navigate to next page.

Sample Input

Open the *registration.html* page on the browser – personal, location information on the login registration form, and fill member information, contact information and home address information then click submit button and navigate to next page.

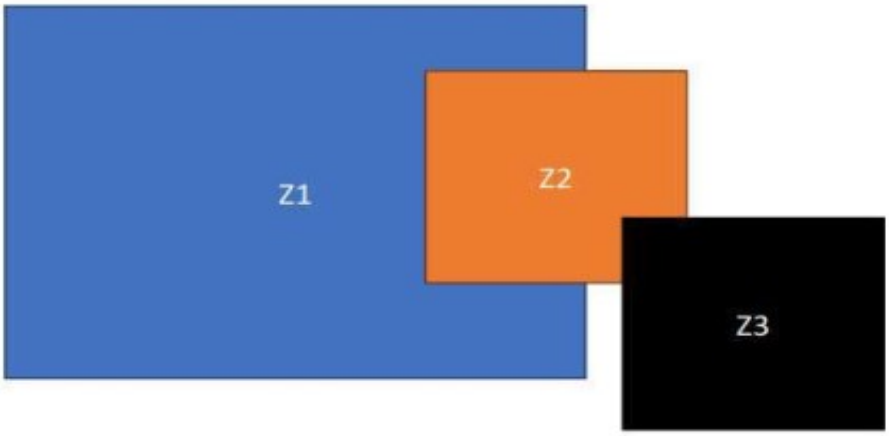
Output

All the pages are shown on the browser based on the user's input. All the tags and styles effects can be shown on the browser.

Exercise No.4 CSS positional parameters and drop down styles to add aesthetic effects for your web page.

1) These 3 boxes are to be arranged in various positional parameters such as static, relative, absolute

2) Use the z-index for the below design



3) Use padding, borders, shadows for the below design

Objectives

The learner will be able understand positioning elements and drop-down menus to provide navigation links without using a lot of screen space.

Procedure / Algorithm

1. Setting up the HTML Structure: Ensure your HTML elements are properly structured. For instance:
2. Setting up the CSS: Create a CSS file (styles.css) and link it to your HTML file.
3. Define the styles for the elements, using position: relative; for relative positioning.
4. The position: relative; property allows you to offset the element from its normal position. It keeps the element in the document flow but shifts its position based on the specified top, right, bottom, and left properties.
    - .box { position: relative; /* Set relative positioning */ top: 20px; /* Adjust the position */ left: 20px; }

    Drop-down menu using CSS:

5. Creating a drop-down menu using CSS involves a combination of HTML and CSS to structure the menu and style it

Input

The various positional parameters can be given while designing CSS

Output

For the above, static, relative and absolute positional parameters are shown. The padding, shadow and border can be viewed.

Exercise No. 5 HTML Form components and validate the content using Java script (looping and conditional constructs and function).

a) Practice basic javascript codes using embedded script tag, and external script js files.

b) Write a Java script code to check, given number is prime or not.

c) Write a Java script code to print fibonacci series.
d) Swapping of 2 variables
e) Find out which number is bigger among the array of numbers
f) Check whether an object is a string or not.
g) Create a JavaScript function that will take a string and extract a given number of characters from it.
h) Create a JavaScript function that will capitalize the initial letter of a string.
i) Create a JavaScript function that capitalizes the first letter of each string's words.
j) Create a JavaScript function that concatenates a given string n (default: 1) times.
k) Create a JavaScript function that divides a string into segments of a specified size

Objective

The learner will be able to practice and enhance JavaScript coding skills through various tasks, including both embedded script tags and external JavaScript files and creating functions to aimed at building a solid foundation in JavaScript programming.

Procedure / Algorithm

1. Create a basic HTML structure.
2. Include placeholders for form elements and other HTML elements that will be manipulated.
3. Add a form to the HTML document and write a JavaScript function to validate the input.
4. Use JavaScript to manipulate HTML elements based on various conditions or actions.
5. Use both embedded <script> tags and external JavaScript files.
6. Ensure the external JavaScript file is linked properly in the HTML <head> or before the closing </body> tag.

Input

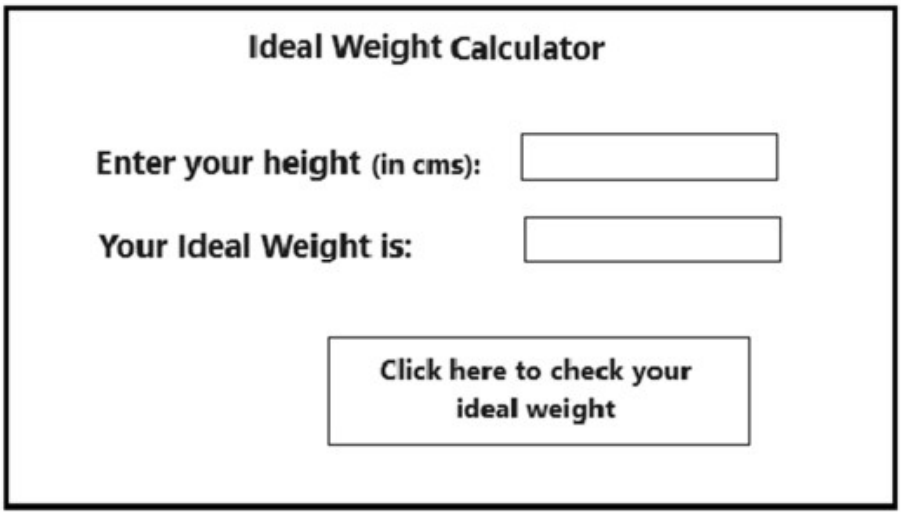
Proper input values should be given for the numeric valued programs and string values can be entered for string manipulation programs. Learner is expected to feed valid and invalid values for the form fields to check whether the code is showing proper messages and showing expected output without any error.

Output

The program is expected to show specified output for the valid expressions and show error messages for the invalid inputs.

Exercise No.6. Java Script components like JS alert, JS button, JS popover to understand the usage.

a) In JavaScript, create a method to add rows to a table when a user clicks the add row button.
b) Try to eliminate items from a drop-down list by writing a JavaScript application.
c) Create a JavaScript function to obtain the First and Last names' values from the registration form and show them in the alert window.
d) Create a JavaScript application to highlight the text in italics in the next paragraph when the mouse hovers over a certain link.
e) Create a JavaScript function that takes three parameters: a row, a column (to specify a specific cell), and a string to alter the cell's contents.
f) Write a JavaScript program to set paragraph background color: get the color option from the user.
g) Write a JavaScript program to count and display dropdown list items (same) in an alert.
h) Write a JavaScript program to calculate your ideal weight. Note the formula to calculate: **Ideal Weight = Height – 100.**



Objective

The objective is to gain proficiency in JavaScript by learning how to effectively implement and utilize key components such as JS alerts, JS buttons, and JS popovers. This includes understanding the syntax and behaviour of these components, mastering event handling, and applying them to create interactive and dynamic web pages. Through hands-on practice, the goal is to develop the ability to use these JavaScript features to enhance user experience and interface functionality in web development projects.

Procedure

Create a basic HTML structure to host the JavaScript components.

JavaScript Alert:

Alerts are simple messages displayed in a dialog box. They are useful for providing information or warnings to users.

1. Create a button in HTML.
2. Write a JavaScript function that uses the alert() method to display a message.
3. Link the button to the function using the onclick attribute.

JavaScript Button:

Buttons can trigger various JavaScript functions. You can define the actions to be taken when a button is clicked.

4. Define a button in HTML and associate it with a JavaScript function.
5. Write the JavaScript function to be executed when the button is clicked.

JavaScript Popover:

A popover is a small overlay that displays additional information. You can create a simple popover using JavaScript and CSS.

6. Create a button and a hidden div for the popover content in HTML.
7. Use CSS to style the popover and initially hide it.
8. Write JavaScript to toggle the display of the popover when the button is clicked.

Input

HTML button, CSS and script programs should be written properly and connected with the button actions (events)

Output

Rendered button and hidden popover div on the web page, Popover toggles visibility when button is clicked.

Exercise No.7: XML document with validation using internal and external Document Type Definition and transformation using XSLT

a) Design an xml document with an internal DTD for the following:

Create a root element as SASTRA, has a child element "schools" which is the parent of elements "school". SOC, SOM & BL are the TEXT of each "school". Value of an attribute "category" of the first "school" element is "computing". Second "school" element contains the following attributes (category='management', campus='main') and 3rd 'school' element consists the attributes (category='law', degree= either 'ug' or 'pg' but default is 'ug').

b) 2. Create an XML document and external DTD for the following

Root <Software>. ***Child tags*** <languages>, <application> and <system>. Every child element should have minimum 1 child element for them with meaningful attributes. Document should have at least one element with no. of occurrences is 2 and attributes with choice based values required.

c) ***Apply the basic styles of XSLT for the "books.xml" document. Transform the content on the browser using the following styles:***

<xsl:for-each ..>

<xsl:sort .. >

<xsl:if..>

<xsl:choice ..>

<xsl:apply-templates..>

Note: for the above ex no 3, kindly refer the XML document from the following url: https://drive.google.com/file/d/1yB6eNaxEtnzhHnEfJ8T6U_6shwlv0b4p/view?usp=sharing

Objective

The learner will be able to create an XML document to create text based light weight data store. The learner will be able to validate the XML content using DTD and transform them using XSLT to view the content on the browser.

Procedure

Creating an XML Document:

1. Define the purpose and structure of your XML document.
2. Identify the root element and the nested elements.
3. Use an XML editor or a text editor to create the XML document.
4. Ensure the XML document is well-formed, adhering to XML syntax rules.

Validation using DTD

1. Link the XML document to the DTD.
2. Use an XML validator tool or a programming language that supports XML validation to check if the XML document conforms to the DTD rules.
3. Ensure all elements, attributes, and the overall structure match the definitions in the DTD.

Transform the XML content on browser

1. Determine the output format (e.g., HTML, text, another XML format).
2. Identify the elements in the XML document to be transformed.
3. Write the XSLT document specifying the transformation rules.
4. Use <xsl:template> elements to match XML elements and define the transformation output.
5. Use an XSLT processor to apply the XSLT document to the XML document.
6. Generate the final output document based on the transformation rules defined in the XSLT.

Input

XML document is created based on syntactic rules that ensure document well-formedness. The content of the XML document can be validated using DTD, hence proper DTD should be associated with an XML document. Finally, transforming the content on the browser using XSLT.

Output

Well structured XML document is created and validated using DTD ruleset. The expected output for the DTD is, to show an error for the elements which are not obeying the rules given in the DTD. XSLT applied XML document will be shown on the browser.

Exercise No. 8 Design a PHP webpage for the purpose of validating the content of various page components.

Develop a Matrimony web site for registering new members before they start using the services. There should be a Master page with the header comprise of the site banner and the footer consist of links to About Us & contact us.

There should be three content pages as follows:

- **Login page**– this page collects the User Id and password and checks them against actual values. Assume that the actual user id is admin@mymatrimony.com and the password is "password123". If login succeeds show that successfully logged in or show that login failed. In the login page there should be a hyper link to "Register new user" page.
- **Registration page** –This page collects the following information from the user
    - Name
    - Gender(use radio buttons)
    - Date of birth(use dropdown lists for DD,MM and Year.
    - Religion(drop down list)
    - Mother tongue
    - Mobile number
    - E-mailId
    - Preferred password
    - Photograph(tobedisplayedattherightsideofthepage&providebrowseoptiontoletuserload his/her photo from system)
- Do necessary validations for Mobile number & E-mail id using HTML Validation controls.
- Calculate the age of the person and if it is less than 18 years in case of a female and less than 20 in case of a male, show invalid age message. Use java script based validation.
- Password should be minimum 8 characters, must contain minimum one capital letter, lowercase letter, numeric digit and a special character. Apply code behind validation using PHP.
- When the submit button is pressed, load the page of registration summary.
- **Registration summary** – In this page present the profile of the user collected

from the registration page in a neat & tabularized manner.

Objective

The learner will be able to develop a Matrimony website that enables the registration of new members before they access the services, featuring a master page with a header containing the site banner and footer with links to About Us and Contact Us pages.

Procedure

This procedure outlines the steps required to create a basic PHP website with a login page, registration page, and registration summary page. It covers setting up the environment, creating the necessary database and tables, and writing the PHP and HTML code for each part of the system, including form handling and user authentication.

1. Install a local server environment such as XAMPP or WAMP.
2. Start the Apache and MySQL services.
3. Design a PHP page for **Login page,** User Id and password and checks them against actual values.
4. If login succeeds show that successfully logged in or show that login failed.
5. In the login page there should be a hyper link to “Register new user” page.
6. Design **Registration** page , which collects name, gender, date of birth and other necessary fields.
7. Write a code to validate mobile and email and other mandatory fields.
8. When the submit button is pressed, load the page of **registration** summary.
9. Display the profile of the user collected from the registration page in a neat & tabularized manner.

Input

User input in the registration page, username, email, password etc to complete registration process.

Output

A success message if login is successful.

An error message if login fails due to incorrect username or password

Exercise no. 9 Web site creation using PHP and provide necessary navigation among the web pages and validate the form content.

Airline Reservation System

Develop a web application using PHP to represent the online Airline reservation system. Please glance at the attached image files and develop the Home page and different content pages that resemble the pages depicted in the images. The Home page should have the navigation bar with menus for navigating to different pages.

There will be one Home page and four content pages in the application

1. Login page
2. Register page
3. Flight Booking Page
4. Confirmation page

Home page

- Should have the navigation bar with menus for navigating to different pages.
- Should have some content like “About Company”.

Login page

- Observe the login.jpg file and implement the login page.
- Create a database “dbAirline” and a table named “tblUsers” with three columns “userid”, “password” and “fullName” under the database. Insert 3 records with some values manually to the table.
- On clicking “login” button, the input username and password needs to be checked in database “dbAirline”.
- On success of login (if user found), should show the message “login success” and should be navigated to “Home Page”. If user is not found in the database, show the proper error message.

Register page

- The register page should include all the fields that are available in the register.jpg image.
- To implement the validations required for the various controls refer the image files with the name ending with _val.
- Implement required field validation for User_id, password, password confirmation and first name fields using the html validation control.

- Implement JavaScript validation for the remaining validations such as mobile number etc.
- On Success of validations, should show the message “registration success” and should be navigated to “Flight Booking Page”

Flight Booking Page

- Provide a menu using hyperlinks controls for Book, Cancel and Ticket status.
- When Book option is clicked redirect to the page titled Confirmation page

Confirmation page

- Show the details of the customer that were entered by him/her in the Register page.

Authentication

- Provide authentication to all the pages except registration page. If any page is requested by any unauthenticated user directly with page url, it needs to be redirected to login page.

Objective

The learner will be able to develop a web application using PHP that represents an online Airline Reservation System. The application should include a Home page with a navigation bar and menus for navigating to different content pages, which should resemble the pages depicted in the attached image files. The application will consist of one Home page and four content pages:

Procedure

1. Install a local server environment such as XAMPP or WAMP.
2. Start the Apache and MySQL services.
3. Create Database and create tables like
    - Eg:- user table, flights, bookings
4. **Project Structure** Create a new folder for your project (e.g., airline_reservation).
5. Index.php login.php, register.php and booking.php
6. Include CSS files with necessary formatting and structuring elements.
7. Collected information will be stored in the data base.

Input

User input in the airline reservation, registration form and searching details for booking flights.

Output

A success message if login is successful.

An error message if login fails due to incorrect username or password

Booking confirmation and reserved ticked details will be shown on the browser.

Exercise No. 10. Design a Web Page and manipulating data in MySQL using PHP. Apply CRUD operations and provide necessary user authentication.

Resource Management System

Create a Resource Management System where HR can allocate employees to projects. The HR can view the employees for a given project and delete the allocation if required.

Below are the tables which are required

Employee Table

Column	DataType/Constraint
EmployeeID	Int ,Primary Key, AUTO_INCREMENT starting with 100
EmployeeName	Varchar(50) –Not null
Experience	Int - Check Constraint > 0,Not null

Insert two records into Employee table using direct sql query.

Project Table

Column	DataType/Constraint
ProjectID	Int ,Primary Key ,AUTO_INCREMENT
ProjectName	Varchar(50) –Not null
Client Name	Varchar(50) –null

Insert two records into Project table using direct sql query.

ProjectAllocation

Column	DataType/Constraint
AllocationID	Int ,Primary Key ,AUTO_INCREMENT
Projectid	Int set as foreign key to corresponding table –not null
EmployeeID	Int set as foreign key to corresponding table-not null
Start Date	DateTime –not null
End Date	DateTime – not null

The HR can log into application, assign employees to a project for a specified duration, view the employees mapped to a project and delete the allocation.

Complete login for HR using Forms authentication. After login HR should be redirected to a menu page which will have menu options for Home, Assign Project and ViewAllocation and Logout. The logged in user id can be shown in all pages.

Both Assign Project and ViewAllocation menu option will navigate to "Select Project" page. Logout options should clear session and redirect to home page.

Select project Page

Controls are listed as below

Project Drop down- Populated from Project table. Value should be project id and Text should be project name.First option should be select.

Assign Button

View Allocation button

When Assign Button is clicked navigate to "AssignAllocation" page and when View Allocation Button is clicked redirect to "ViewAllocation" page. Assign Projectid and project name to session when both Assign button and View Allocation is clicked and then redirect to corresponding pages. Complete mandatory validation using javascript for Project dropdown.

Assign Allocation Page

Has below controls

Field	**Remarks**
Project –label	Name Should be populated using session
EmployeeID – TextBox Mandatory Validation	Enter one of the id from Employee Table
Start Allocation Date –Date-Mandatory Validation	Should be future date
End Allocation Date – Date – Mandatory Validation	Should be future date

On Click of submit button add the details to Project allocation table. The details added should be employeeid, projectid (from session), start date and end date. Before adding just check the difference between the start and end allocation date. It should be > 0. If difference is > 0 then insert into allocation table, else show the error message in a separate label below all fields.

View Allocation Page

Show the details of employees such as llocationid, Employeeid, Employeename, StartDate, EndDate in a table format for the passed project id in session. Complete the deletion for any allocation id in the gridview. Hint- Use Joins to display the employee name along with the other required details. Complete Master Page (header & footer) for all pages. Master Page should have the heading as Resouce Management system. Complete validations as required. Exceptions (run time errors) should be handled properly.
The UI should be consistent for all pages.

Objective

The learner will be able to design a web page and manipulate data in MySQL using PHP by applying CRUD operations and providing necessary user authentication related to Resource Management System. The learner can develop a system to allocate employees to projects. HR should be able to view the employees allocated to a given project and delete allocations if required. The system will utilize the specified tables to manage the data.

Procedure

1. Identify System Requirements:

- Understand the functionalities required for the Resource Management System.
- Identify the roles and permissions, particularly focusing on the HR role.
- Define the CRUD operations needed: Create, Read, Update, Delete.

2. Database Design

1. Define Tables:
    - Users Table: Stores user details and roles.
    - Employees Table: Stores employee details.
    - Projects Table: Stores project details.
    - Allocations Table: Manages the many-to-many relationship between employees and projects.
1. Create Tables in MySQL:
    - Design the schema for the above tables with appropriate data types and constraints.
    - Define primary keys, foreign keys, and indexes.

Step 3: Set Up Development Environment

1. Install Required Software:
    - Install a web server (e.g., Apache).
    - Install PHP.

Step 4: Develop Authentication System

1. User Registration:
    - Create a registration form for new users.
    - Implement server-side validation and hash passwords before storing them in the database.
2. User Login:
    - Create a login form.

- Validate user credentials against the stored data in the database.
- Implement session management to maintain user state.

3. User Logout:
    - Implement a logout functionality that destroys the user session.

Step 5: Develop CRUD Operations for Employees and Projects

1. Create Operations:
    - Develop forms to add new employees and projects.
    - Write PHP scripts to handle form submissions and insert data into the database.
2. Read Operations:
    - Develop interfaces to list employees and projects.
    - Write PHP scripts to fetch data from the database and display it in a structured format.
3. Update Operations:
    - Create forms to update employee and project details.
    - Write PHP scripts to handle form submissions and update data in the database.
4. Delete Operations:
    - Implement functionality to delete employees and projects.
    - Write PHP scripts to handle delete requests and remove data from the database.

Step 6: Manage Employee Allocations to Projects

1. Allocation Creation:
    - Create forms to allocate employees to projects.
    - Write PHP scripts to handle form submissions and insert data into the Allocations table.
2. View Allocations:
    - Develop interfaces to list employees allocated to a given project.
    - Write PHP scripts to fetch and display allocation data.
3. Delete Allocations:
    - Implement functionality to delete allocations.
    - Write PHP scripts to handle delete requests and remove data from the Allocations table.

Step 7: Implement User Interface

1. Design Web Pages:

- Use HTML, CSS, and JavaScript to create user-friendly web pages.
- Ensure the pages are responsive and accessible.

2. Integrate PHP Scripts:
    - Embed PHP scripts within the web pages to handle CRUD operations and user authentication.

Input

The project related details are entered in the system and employee related details are given as input. The necessary inputs should be given for performing CRUD operations.

Output

Employee details, project details and project assigned details are shown on the browser based on user input.