AIT

1 Create Resume by using HTML and HTML5 tags

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Resume</title>

<style>

body {

font-family: Arial, sans-serif;

}

h1, h2 {

text-align: center;

}

.section {

margin-bottom: 20px;

}

.section h3 {

margin-bottom: 10px;

}

.section p {

margin-bottom: 5px;

}

</style>

</head>

<body>

<header>

<h1>Amar Jha</h1>

<p>Fresher</p>

<p>jamar1410@gmail.com | (+91) 8104081639 | LinkedIn: <a href="https://www.linkedin.com/in/johndoe">Amar Jha</a></p>

</header>

<section class="section">

<h2>Summary</h2>

<p>Frontend developer with a passion for creating user-friendly and visually appealing websites. Proficient in HTML, CSS, and JavaScript.</p>

</section>

<section class="section">

<h2>Skills</h2>

<ul>

<li>HTML5</li>

<li>CSS3</li>

<li>JavaScript</li>

<li>Java</li>

<li>SQL</li>

</ul>

</section>

<section class="section">

<h2>Education</h2>

<h3>Master of Computer Application(Pursuing)

<h3>Bachelor of Computer Application</h3>

<p>Graduated in May 2022</p>

</section>

<section class="section">

<h2>Experience</h2>

<h3>None</h3>

<p> Present</p>

</section>

</body>

</html>

2 Program to implement Audio and Video features for your web page.

<!DOCTYPE html>

<html>

<head>

<title>Audio and Video tag in HTML5 </title>

</head>

<body>

<h2>Audio Tag</h2>

<audio controls>

<source src="C:\Users\asus\OneDrive\Desktop\AIT lab code\something.mp3" type="audio/mpeg">

</audio>

<h2>Video Tag</h2>

<video controls width="400">

<source src="C:\Users\asus\OneDrive\Desktop\AIT lab code\Dance.mp4" type="video/mp4">

</video>

</body>

</html>

3 Write HTML Programs design different geometrical shapes using Canvas and SVG (minimum 5 Shapes).

<!DOCTYPE html>

<html>

<head>

<title>Shapes using Canvas</title>

<style>

#myCanvas {

border: 1px solid black;

}

#myCanvas1 {

border: 1px solid black;

}

</style>

</head>

<body>

<canvas id="myCanvas" width="200" height="200"></canvas>

<canvas id="myCanvas1" width="200" height="200"></canvas>

<script>

var canvas = document.getElementById("myCanvas");

var context = canvas.getContext("2d");

context.fillRect(50, 50, 100, 80);

var canvas1 = document.getElementById("myCanvas1");

var context = canvas1.getContext("2d");

context.beginPath();

context.arc(100, 100, 50, 0, 2 \* Math.PI);

context.fill();

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>Shapes using SVG</title>

<style>

svg {

border: 1px solid black;

}

</style>

</head>

<body>

<svg width="200" height="200">

<polygon points="100,20 20,180 180,180" fill="red" />

</svg>

<svg width="200" height="200">

<ellipse cx="100" cy="100" rx="80" ry="50" fill="blue" />

</svg>

<svg width="200" height="200">

<line x1="20" y1="20" x2="180" y2="180" stroke="green" />

</svg>

</body>

</html>

4 Program to design form using HTML5 elements, attributes and Semantics tags and apply HTML5 validation to it

<!DOCTYPE html>

<html>

<head>

<title>HTML5 Form</title>

</head>

<body>

<h1>HTML5 Form</h1>

<form action="#" method="post">

<div>

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

</div>

<div>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

</div>

<div>

<label for="age">Age:</label>

<input type="number" id="age" name="age" min="18" max="100" required>

</div>

<div>

<label for="gender">Gender:</label>

<select id="gender" name="gender" required>

<option value="">-- Select --</option>

<option value="male">Male</option>

<option value="female">Female</option>

<option value="other">Other</option>

</select>

</div>

<div>

<label for="message">Message:</label>

<textarea id="message" name="message" required></textarea>

</div>

<div>

<input type="submit" value="Submit">

</div>

</form>

</body>

</html>

5 Programs to demonstrate external and internal styles in the web page using font, text, background, borders, opacity and other CSS 3 properties

Internal css

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="styles.css">

<style>

body {

font-family: Verdana, Geneva, sans-serif;

background-color: #f7f7f7;

}

h1 {

color: #ff0000;

}

p {

font-size: 18px;

line-height: 1.6;

}

.box {

width: 200px;

height: 200px;

border: 1px solid #333333;

background-color: #ffffcc;

opacity: 0.8;

}

</style>

</head>

<body>

<h1>Internal Style Sheet</h1>

<p>This is an example of using an internal style sheet.</p>

<div class="box">

<p>This is a box with some text inside.</p>

</div>

</body>

</html>

External css

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>External Style Sheet</h1>

<p>This is an example of using an external style sheet.</p>

<div class="container">

<p>This is a box with some text inside.</p>

</div>

</body>

</html>

**Styles.css**

**/\* styles.css \*/**

**body {**

**font-family: Arial, sans-serif;**

**background-color: #f2f2f2;**

**}**

**h1 {**

**color: #ff0000;**

**}**

**p {**

**font-size: 16px;**

**line-height: 1.5;**

**}**

**.container {**

**width: 80%;**

**margin: 0 auto;**

**padding: 20px;**

**border: 1px solid #dddddd;**

**background-color: #ffffff;**

**}**

6 Implement Transformation using Translation, Rotation and Scaling in your web page

<!DOCTYPE html>

<html>

<head>

<title>Transformation Demo</title>

<style>

#myElement {

width: 100px;

height: 100px;

background-color: red;

transition: transform 0.3s ease;

}

</style>

</head>

<body>

<div id="myElement"></div>

<button onclick="translate()">Translate</button>

<button onclick="rotate()">Rotate</button>

<button onclick="scale()">Scale</button>

<script src="ait7.js"></script>

</body>

</html>

.js file

function translate() {

var element = document.getElementById("myElement");

element.style.transform = "translate(50px, 50px)";

}

function rotate() {

var element = document.getElementById("myElement");

element.style.transform = "rotate(45deg)";

}

function scale() {

var element = document.getElementById("myElement");

element.style.transform = "scale(1.5)";

}

7 Program to show current date and time using user defined module in AngularJs.

<!DOCTYPE html>

<html ng-app="DateTimeApp">

<head>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js"></script>

<script src="ait8.js"></script>

</head>

<body>

<div ng-controller="DateTimeController">

<h1>Current Date and Time</h1>

<p>{{ currentDate | date:'fullDate' }}</p>

<p>{{ currentTime | date:'mediumTime' }}</p>

</div>

</body>

</html>

.js file

// app.js

angular.module('DateTimeApp', [])

.controller('DateTimeController', ['$scope', '$interval', function($scope, $interval) {

$interval(function() {

$scope.currentDate = new Date();

$scope.currentTime = new Date();

}, 1000);

}]);

8 Write calculator program in Angular Js to perform basic arithmetic operations(+, -, \*, /) using angular controller function

<!DOCTYPE html>

<head>

<title>AngularJS Calculator</title>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<script>

angular.module('calculatorApp', [])

.controller('CalculatorController', function ($scope) {

$scope.num1 = 0;

$scope.num2 = 0;

$scope.result = 0;

$scope.add = function () {

$scope.result = $scope.num1 + $scope.num2;

};

$scope.subtract = function () {

$scope.result = $scope.num1 - $scope.num2;

};

$scope.multiply = function () {

$scope.result = $scope.num1 \* $scope.num2;

};

$scope.divide = function () {

$scope.result = $scope.num1 / $scope.num2;

};

});

</script>

</head>

<body ng-controller="CalculatorController">

<h1>AngularJS Calculator</h1>

<label for="num1">Number 1:</label>

<input type="number" id="num1" ng-model="num1">

<label for="num2">Number 2:</label>

<input type="number" id="num2" ng-model="num2">

<br>

<button ng-click="add()">Add</button>

<button ng-click="subtract()">Subtract</button>

<button ng-click="multiply()">Multiply</button>

<button ng-click="divide()">Divide</button>

<br>

<label for="result">Result:</label>

<input type="number" id="result" ng-model="result" disabled>

</body>

</html>

9 Program using NPM which will convert entered string into either case

<!DOCTYPE html>

<html>

<head>

<title>String Case Converter</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js"></script>

<script>

function convertString() {

var inputString = document.getElementById('inputString').value;

var lowercaseString = \_.toLower(inputString);

var uppercaseString = \_.toUpper(inputString);



document.getElementById('lowercaseResult').textContent = lowercaseString;

document.getElementById('uppercaseResult').textContent = uppercaseString;

}

</script>

</head>

<body>

<h1>String Case Converter</h1>

<label for="inputString">Enter a string:</label>

<input type="text" id="inputString">

<button onclick="convertString()">Convert</button>

<h2>Lowercase:</h2>

<div id="lowercaseResult"></div>

<h2>Uppercase:</h2>

<div id="uppercaseResult"></div>

</body>

</html>

10 Program to demonstrate angular project which will demonstrate the usage of component directive, structural directive and attribute directives

Directive component.html

<h2>Component Directive</h2>

<p>This is a component directive.</p>

<h2>Structural Directive</h2>

<div \*ngIf="showContent">

<p>This content is shown using a structural directive.</p>

</div>

<h2>Attribute Directive</h2>

<p [appHighlight]="'yellow'">This text is highlighted using an attribute directive.</p>

Component.ts

import { Component } from '@angular/core';

@Component({

selector: 'app-directive-demo',

templateUrl: './directive-demo.component.html',

styleUrls: ['./directive-demo.component.css']

})

export class DirectiveDemoComponent {

showContent = true;

}

Highlight.ts

import { Directive, ElementRef, Input, Renderer2 } from '@angular/core';

@Directive({

selector: '[appHighlight]'

})

export class HighlightDirective {

@Input('appHighlight') highlightColor!: string;

constructor(private el: ElementRef, private renderer: Renderer2) {}

ngOnInit() {

this.renderer.setStyle(this.el.nativeElement, 'background-color', this.highlightColor);

}

}

App-component.ts

<app-directive-demo></app-directive-demo>

11 Write Program for Form validation in Angular.

App.component.html

<div class="container">

<h1>

Welcome to {{title}}!!

</h1>

<form [formGroup]="angForm" novalidate>

<div class="form-group">

<label>Name:</label>

<input class="form-control" formControlName="name" type="text">

</div>

<div \*ngIf="angForm.controls['name'].invalid && (angForm.controls['name'].dirty || angForm.controls['name'].touched)" class="alert alert-danger">

<div \*ngIf="angForm.controls['name'].errors required">

Name is required.

</div>

</div>

<div class="form-group">

<label>Address:</label>

<input class="form-control" formControlName="address" type="text">

</div>

<div \*ngIf="angForm.controls['address'].invalid && (angForm.controls['address'].dirty || angForm.controls['address'].touched)" class="alert alert-danger">

<div \*ngIf="angForm.controls['address'].errors required">

email is required.

</div>

</div>

<button type="submit"

[disabled]="angForm.pristine || angForm.invalid" class="btn btn-success">

Save

</button>

</form>

<br />

<p>Form value: {{ angForm.value | json }}</p>

<p>Form status: {{ angForm.status | json }}</p>

</div>

App.component.ts

import { Component } from '@angular/core';

import { FormGroup, FormBuilder, Validators } from '@angular/forms';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'Angular Form Validation Tutorial';

angForm !: FormGroup;

constructor(private fb: FormBuilder) {

this.createForm();

}

createForm() {

this.angForm = this.fb.group({

name: ['', Validators.required ],

address: ['', Validators.required ]

});

}

}

App.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ReactiveFormsModule } from '@angular/forms';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ReactiveFormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

12 Write a program to create a calculator using Angular.

App.component.html

<div class="calculator">

<h2>Calculator</h2>

<input type="text" [(ngModel)]="display" readonly>

<div class="buttons">

<button (click)="addToDisplay('1')">1</button>

<button (click)="addToDisplay('2')">2</button>

<button (click)="addToDisplay('3')">3</button>

<button (click)="addToDisplay('4')">4</button>

<button (click)="addToDisplay('5')">5</button>

<button (click)="addToDisplay('6')">6</button>

<button (click)="addToDisplay('7')">7</button>

<button (click)="addToDisplay('8')">8</button>

<button (click)="addToDisplay('9')">9</button>

<button (click)="addToDisplay('0')">0</button>

<button (click)="addToDisplay('+')">+</button>

<button (click)="addToDisplay('-')">-</button>

<button (click)="addToDisplay('\*')">\*</button>

<button (click)="addToDisplay('/')">/</button>

<button (click)="clearDisplay()">C</button>

<button (click)="calculateResult()">=</button>

</div>

</div>

App.component.ts

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

display: string = '';

addToDisplay(value: string) {

this.display += value;

}

clearDisplay() {

this.display = '';

}

calculateResult() {

try {

this.display = eval(this.display);

} catch (e) {

this.display = 'Error';

}

}

}

App.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { CalculatorComponent } from './calculator/calculator.component';

@NgModule({

declarations: [

AppComponent,

CalculatorComponent

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

13 Write a program to create a calculator using Node JS

Calculator.js

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('Enter the first number: ', (num1) => {

rl.question('Enter the second number: ', (num2) => {

rl.question('Enter the operation (+, -, \*, /): ', (operator) => {

const n1 = parseFloat(num1);

const n2 = parseFloat(num2);

let result;

switch (operator) {

case '+':

result = n1 + n2;

break;

case '-':

result = n1 - n2;

break;

case '\*':

result = n1 \* n2;

break;

case '/':

result = n1 / n2;

break;

default:

console.log('Invalid operator');

rl.close();

return;

}

console.log(`Result: ${result}`);

rl.close();

});

});

});

14 (1) Write REPL code to print number from 1 to 50.

for (var i=1;i<=50;i++){

... console.log(i)

... }

14 (2) Write REPL code to print largest number from three given number

var x=5;

var y=10;

var z=15;

console.log(Math.max(x,y,z)+" is the greatest");

15 Create angular project which has HTML template and handle the click event on click of the button

App.component.html

<h1>Angular Button Click Example</h1>

<button (click)="handleClick()">Click me!</button>

<p>{{ message }}</p>

App.component.ts

import { Component } from "@angular/core";

@Component({

selector: "app-root",

templateUrl: "./app.component.html",

styleUrls: ["./app.component.css"],

})

export class AppComponent {

message: string = '';

handleClick(): void {

this.message = 'Button clicked!';

}

}

App.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { MyComponentComponent } from './my-component/my-component.component';

@NgModule({

declarations: [

AppComponent,

MyComponentComponent

],

imports: [

BrowserModule,

AppRoutingModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

16 Program for basic operations, array and user interface handling

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

function performBasicOperations() {

rl.question('Enter two numbers: ', (input) => {

const numbers = input.split(' ').map(Number);

const sum = numbers[0] + numbers[1];

const difference = numbers[0] - numbers[1];

const product = numbers[0] \* numbers[1];

const quotient = numbers[0] / numbers[1];

console.log('Sum:', sum);

console.log('Difference:', difference);

console.log('Product:', product);

console.log('Quotient:', quotient);

rl.close();

});

}

function manipulateArray() {

rl.question('Enter elements of an array (space-separated): ', (input) => {

const array = input.split(' ');

console.log('Array:', array);

console.log('Length:', array.length);

console.log('First element:', array[0]);

console.log('Last element:', array[array.length - 1]);

rl.close();

});

}

function handleUserInterface() {

rl.question('Choose an operation: 1. Basic Operations 2. Array Manipulation ', (choice) => {

if (choice === '1') {

performBasicOperations();

} else if (choice === '2') {

manipulateArray();

} else {

console.log('Invalid choice! Please try again.');

handleUserInterface();

}

});

}

handleUserInterface();

17 Program to demonstrate session management using various techniques

const express = require("express")

const session = require('express-session')

const app = express()

// Port Number Setup

var PORT = process.env.port || 3000

// Session Setup

app.use(session({

// It holds the secret key for session

secret: 'Your\_Secret\_Key',

// Forces the session to be saved

// back to the session store

resave: true,

// Forces a session that is "uninitialized"

// to be saved to the store

saveUninitialized: true

}))

app.get("/", function(req, res){

// req.session.key = value

req.session.name = 'This is session'

return res.send("Session Set")

})

app.get("/session", function(req, res){

var name = req.session.name

return res.send(name)

/\* To destroy session you can use

this function

req.session.destroy(function(error){

console.log("Session Destroyed")

})

\*/

})

app.listen(PORT, function(error){

if(error) throw error

console.log("Server created Successfully on PORT :", PORT)

})

18 Design Employee Information form in PHP and perform the CRUD Operations using PHP Script.

a) Create mydb database using PHP code

b) Create employee\_info table using PHP code

c) Write PHP program to design Employee information form which includes fields like (Employee ID, First Name, Last name, Contact No, Designation, Department, Salary….)

A: db.php

<?php

$servername = "localhost";

$username = "root";

$password = "";

// Create connection

$conn = new mysqli($servername, $username, $password);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Create database

$sql = "CREATE DATABASE IF NOT EXISTS mydb";

if ($conn->query($sql) === TRUE) {

echo "Database created successfully";

} else {

echo "Error creating database: " . $conn->error;

}

$conn->close();

?>

B: Table.php

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "mydb";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Create table

$sql = "CREATE TABLE IF NOT EXISTS employee\_info (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(30) NOT NULL,

last\_name VARCHAR(30) NOT NULL,

contact\_no VARCHAR(15) NOT NULL,

designation VARCHAR(50) NOT NULL,

department VARCHAR(50) NOT NULL,

salary FLOAT(10, 2) NOT NULL

)";

if ($conn->query($sql) === TRUE) {

echo "Table 'employee\_info' created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

$conn->close();

?>

C: EMP form

<!DOCTYPE html>

<html>

<head>

<title>Employee Information Form</title>

</head>

<body>

<h2>Employee Information Form</h2>

<form method="post" action="">

<label for="employee\_id">Employee ID:</label>

<input type="text" name="employee\_id" id="employee\_id" required><br><br>

<label for="first\_name">First Name:</label>

<input type="text" name="first\_name" id="first\_name" required><br><br>

<label for="last\_name">Last Name:</label>

<input type="text" name="last\_name" id="last\_name" required><br><br>

<label for="contact\_no">Contact No:</label>

<input type="text" name="contact\_no" id="contact\_no" required><br><br>

<label for="designation">Designation:</label>

<input type="text" name="designation" id="designation" required><br><br>

<label for="department">Department:</label>

<input type="text" name="department" id="department" required><br><br>

<label for="salary">Salary:</label>

<input type="text" name="salary" id="salary" required><br><br>

<input type="submit" name="submit" value="Submit">

</form>

</body>

</html>

19 Design Students Registration form (Signup form), Student Login form (signIn form) in PHP and implement session, cookies concept in it.

a) Registration form consists (Student id, Student first & Surname, mobile No, Address…)

b) Student Login form (Student Username, Password)

c) Sign-out form (session destroy implementation)

A Signup Form

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

</head>

<body>

<h2>Registration Form</h2>

<form method="post" action="register.php">

<label for="student\_id">Student ID:</label>

<input type="text" name="student\_id" id="student\_id" required><br>

<label for="first\_name">First Name:</label>

<input type="text" name="first\_name" id="first\_name" required><br>

<label for="last\_name">Last Name:</label>

<input type="text" name="last\_name" id="last\_name" required><br>

<label for="mobile\_no">Mobile No:</label>

<input type="text" name="mobile\_no" id="mobile\_no" required><br>

<label for="address">Address:</label>

<textarea name="address" id="address" required></textarea><br>

<label for="username">Username:</label>

<input type="text" id="username" name="username" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required><br><br>

<input type="submit" value="Register">

</form>

</body>

</html>

Register.php

<?php

// Start a session

session\_start();

// Process registration form data

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Retrieve form data

$student\_id = $\_POST["student\_id"];

$first\_name = $\_POST["first\_name"];

$last\_name = $\_POST["last\_name"];

$mobile\_no = $\_POST["mobile\_no"];

$address = $\_POST["address"];

$username= $\_POST["username"];

$password=$\_POST["password"];

// Store form data in session variables

$\_SESSION["student\_id"] = $student\_id;

$\_SESSION["first\_name"] = $first\_name;

$\_SESSION["last\_name"] = $last\_name;

$\_SESSION["mobile\_no"] = $mobile\_no;

$\_SESSION["address"] = $address;

$\_SESSION["username"] = $username;

$\_SESSION["password"] = $password;

// Redirect to the login page

header("Location: login\_form.php");

exit();

}

?>

B: Login form

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

</head>

<body>

<h2>Login Form</h2>

<form method="post" action="login.php">

<label for="username">Username:</label>

<input type="text" name="username" id="username" required><br>

<label for="password">Password:</label>

<input type="password" name="password" id="password" required><br>

<input type="submit" value="Login">

</form>

</body>

</html>

Login.php

<?php

// Start a session

session\_start();

// Process login form data

if ($\_SERVER["REQUEST\_METHOD"] === "POST") {

// Retrieve form data

$username = $\_POST["username"];

$password = $\_POST["password"];

// Check if the username and password are correct (You'll need to implement your own logic here)

// For this example, we'll use a hardcoded username and password

if ($username === $\_SESSION["username"] && $password === $\_SESSION["password"]) {

// Set a cookie to remember the login

setcookie("loggedin", "true", time() + (86400 \* 30), "/"); // Cookie expires in 30 days

// Redirect to the home page

header("Location: home.php");

exit();

} else {

echo "Invalid username or password";

}

}

?>

Home.php

<?php

// Start a session

session\_start();

// Check if the user is logged in

if (!isset($\_COOKIE["loggedin"])) {

// Redirect to the login page if not logged in

header("Location: login\_form.php");

exit();

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<h2>Welcome, <?php echo $\_SESSION["first\_name"] . " " . $\_SESSION["last\_name"]; ?></h2>

<p>Student ID: <?php echo $\_SESSION["student\_id"]; ?></p>

<p>Mobile No: <?php echo $\_SESSION["mobile\_no"]; ?></p>

<p>Address: <?php echo $\_SESSION["address"]; ?></p>

<form method="post" action="logout.php">

<input type="submit" value="Sign Out">

</form>

</body>

</html>

C: Signout

<?php

// Start a session

session\_start();

// Destroy the session

session\_destroy();

// Clear the login cookie

setcookie("loggedin", "", time() - 3600, "/");

// Redirect to the login page

header("Location: login\_form.php");

exit();

?>

20 Write event drive program to perform operations like select, insert and delete on Employee information using PHP & MySQL.

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "mydb";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Function to select and display all employee records

function selectEmployees()

{

global $conn;

$sql = "SELECT \* FROM employee\_info";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

echo "<h3>Employee Records:</h3>";

echo "<table>

<tr>

<th>ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Contact No</th>

<th>Designation</th>

<th>Department</th>

<th>Salary</th>

</tr>";

while ($row = $result->fetch\_assoc()) {

echo "<tr>

<td>".$row["id"]."</td>

<td>".$row["first\_name"]."</td>

<td>".$row["last\_name"]."</td>

<td>".$row["contact\_no"]."</td>

<td>".$row["designation"]."</td>

<td>".$row["department"]."</td>

<td>".$row["salary"]."</td>

</tr>";

}

echo "</table>";

} else {

echo "No employee records found";

}

}

// Function to insert an employee record

function insertEmployee($employee\_id, $first\_name, $last\_name, $contact\_no, $designation, $department, $salary)

{

global $conn;

$sql = "INSERT INTO employee\_info (id, first\_name, last\_name, contact\_no, designation, department, salary)

VALUES ('$employee\_id', '$first\_name', '$last\_name', '$contact\_no', '$designation', '$department', '$salary')";

if ($conn->query($sql) === TRUE) {

echo "Employee record created successfully";

} else {

echo "Error creating employee record: " . $conn->error;

}

}

// Function to delete an employee record

function deleteEmployee($employee\_id)

{

global $conn;

$sql = "DELETE FROM employee\_info WHERE id='$employee\_id'";

if ($conn->query($sql) === TRUE) {

echo "Employee record deleted successfully";

} else {

echo "Error deleting employee record: " . $conn->error;

}

}

// Event-driven logic

if (isset($\_POST['select'])) {

selectEmployees();

}

if (isset($\_POST['insert'])) {

$employee\_id = $\_POST['employee\_id'];

$first\_name = $\_POST['first\_name'];

$last\_name = $\_POST['last\_name'];

$contact\_no = $\_POST['contact\_no'];

$designation = $\_POST['designation'];

$department = $\_POST['department'];

$salary = $\_POST['salary'];

insertEmployee($employee\_id, $first\_name, $last\_name, $contact\_no, $designation, $department, $salary);

}

if (isset($\_POST['delete'])) {

$employee\_id = $\_POST['employee\_id'];

deleteEmployee($employee\_id);

}

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

<title>Employee Information</title>

</head>

<body>

<h2>Employee Information</h2>

<form method="post" action="">

<input type="submit" name="select" value="Select Employees"><br><br>

<label for="employee\_id">Employee ID:</label>

<input type="text" name="employee\_id" id="employee\_id" required><br><br>

<label for="first\_name">First Name:</label>

<input type="text" name="first\_name" id="first\_name" ><br><br>

<label for="last\_name">Last Name:</label>

<input type="text" name="last\_name" id="last\_name" ><br><br>

<label for="contact\_no">Contact No:</label>

<input type="text" name="contact\_no" id="contact\_no" ><br><br>

<label for="designation">Designation:</label>

<input type="text" name="designation" id="designation" ><br><br>

<label for="department">Department:</label>

<input type="text" name="department" id="department" ><br><br>

<label for="salary">Salary:</label>

<input type="text" name="salary" id="salary" ><br><br>

<input type="submit" name="insert" value="Insert Employee">

<input type="submit" name="delete" value="Delete Employee">

</form>

</body>

</html>

21 Explain Selectors and Pseudo Classes of CSS3 with example. Write HTML code with CSS3 classes to design Indian Flag.(Apply Transition & Animation)

Flag.html

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="flag.css">

</head>

<body>

<div class="wrapper">

<div class="stick"></div>

<div class="flag">

<div class="wave"></div>

<div class="top"></div>

<div class="middle">

<div class="wheel">

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

<span class="line"></span>

</div>

</div>

<div class="bottom"></div>

</div>

</div>

</body>

</html>

Flag.css

.wrapper {

display: flex;

}

.stick {

height: 450px;

width: 10px;

background: black;

border-top-left-radius: 10px;

border-bottom-left-radius: 5px;

border-bottom-right-radius: 5px;

}

.flag {

width: 270px;

height: 180px;

box-shadow: 0px 0px 90px 1px #989;

background-color: transparent;

position: relative;

}

.top {

height: 60px;

background-color: #ff9933

}

.middle {

height: 60px;

display: flex;

justify-content: center;

align-items: center;

}

.bottom {

height: 60px;

background-color: green

}

.wheel {

height: 43px;

width: 43px;

border: 1px solid darkblue;

border-radius: 45px;

position: relative;

margin: 0 auto;

animation-name: wheel;

animation-iteration-count: infinite;

animation-duration: 5s;

animation-timing-function: linear;

}

.wheel .line {

height: 100%;

width: 1px;

display: inline-block;

position: absolute;

left: 50%;

background: darkblue;

}

.line:nth-child(1) {

transform: rotate(15deg)

}

.line:nth-child(2) {

transform: rotate(30deg)

}

.line:nth-child(3) {

transform: rotate(45deg)

}

.line:nth-child(4) {

transform: rotate(60deg)

}

.line:nth-child(5) {

transform: rotate(75deg)

}

.line:nth-child(6) {

transform: rotate(90deg)

}

.line:nth-child(7) {

transform: rotate(105deg)

}

.line:nth-child(8) {

transform: rotate(120deg)

}

.line:nth-child(9) {

transform: rotate(135deg)

}

.line:nth-child(10) {

transform: rotate(150deg)

}

.line:nth-child(11) {

transform: rotate(165deg)

}

.line:nth-child(12) {

transform: rotate(180deg)

}

@keyframes wheel {

0%{

transform: rotate(0deg);

}

100%{

transform: rotate(360deg);

}

}

.wave{

position: absolute;

height: 100%;

width: 100%;

background-image: linear-gradient(

128deg, rgba(89,72,21,0) 39%,

rgba(250,245,245,0.8474025974025974)

46%, rgba(89,72,21,0) 53%);

animation-name: wavy;

animation-duration: 10s;

animation-iteration-count: infinite;

animation-timing-function: linear;

}

@keyframes wavy {

0%{

background-position:

-400px 0px, -400px 0px,

-400px 0px,-400px 0px;

}

100%{

background-position: 800px 0px,

800px 0px, 800px 0px, 800px 0px;

}

}