Sr. No Program Name

1 Program to play audio with controls attribute for your web page.

```
<!DOCTYPE html>
<html>
<body>

<audio controls autoplay>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>

</body>
</html>
```

2 Program to play video with autoplay attribute for your web page.

```
<!DOCTYPE html>
<html>
<body>

<video width="400" controls>
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogg" type="video/ogg">
  Your browser does not support HTML video.
</video>

<p>
Video courtesy of
<a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.
</p>

</body>
</html>
```

3 Program to design form using following HTML5 semantics tags. 1.<progress> 2. <main> 3. <mark> 4. <time> 5. <article>

```
<!DOCTYPE html>
<html>
<head>
 <title>Form Design</title>
</head>
<body>
 <main>
  <article>
   <h2>Registration Form</h2>
   <form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="dob">Date of Birth:</label>
    <input type="date" id="dob" name="dob" required><br><br>

    <label for="progress">Progress:</label>
    <progress id="progress" value="50" max="100"></progress><br><br>

    <label for="interest">Interests:</label><br>
    <mark>Choose at least one:</mark><br>
    <input type="checkbox" id="interest1" name="interest1">
    <label for="interest1">Sports</label><br>
    <input type="checkbox" id="interest2" name="interest2">
    <label for="interest2">Music</label><br>
```

```
        <input type="checkbox" id="interest3" name="interest3">
        <label for="interest3">Art</label><br><br>

        <label for="submission">Submission Time:</label>
        <time id="submission" datetime="2023-06-25T10:30:00Z">June 25, 2023 10:30
AM</time><br><br>

        <input type="submit" value="Submit">
      </form>
    </article>
  </main>
</body>
</html>
```

4 Draw a text with strokeText() and draw linear gradient using Canvas.

```
<!DOCTYPE html>
<html>
<head>
  <title>Canvas Text with Stroke and Linear Gradient</title>
</head>
<body>
  <canvas id="myCanvas" width="400" height="200"></canvas>

  <script>
    // Get the canvas element
    const canvas = document.getElementById('myCanvas');
    const ctx = canvas.getContext('2d');

    // Set the font style
    ctx.font = '40px Arial';

    // Set stroke color and width
    ctx.strokeStyle = '#FF0000';
    ctx.lineWidth = 3;

    // Draw text with stroke
    ctx.strokeText('Hello, Canvas!', 50, 100);

    // Create a linear gradient
    const gradient = ctx.createLinearGradient(0, 0, canvas.width, 0);
    gradient.addColorStop(0, 'red');
    gradient.addColorStop(0.5, 'yellow');
    gradient.addColorStop(1, 'blue');
```

```
    // Apply the gradient to the text
    ctx.fillStyle = gradient;
    ctx.fillText('Hello, Canvas!', 50, 100);
  </script>
</body>
</html>
```

5 Draw rectangle, rounded rectangle, circle, star using SVG graphics.

```
<!DOCTYPE html>
<html>
<head>
  <title>SVG Graphics</title>
</head>
<body>
  <svg width="400" height="400" xmlns="http://www.w3.org/2000/svg">
    <!-- Rectangle -->
    <rect x="50" y="50" width="100" height="50" fill="blue" />

    <!-- Rounded Rectangle -->
    <rect x="50" y="120" rx="10" ry="10" width="100" height="50" fill="red" />

    <!-- Circle -->
    <circle cx="200" cy="100" r="50" fill="yellow" />

    <!-- Star -->
    <polygon points="275,120 300,180 350,180 310,220 330,280 275,240 220,280 240,220 200,180
250,180" fill="green" />
  </svg>
</body>
</html>
```

6 Program to implement HTML5 drag and drop feature.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
#div1 {
  width: 350px;
  height: 70px;
  padding: 10px;
  border: 1px solid #aaaaaa;
}
</style>
<script>
function allowDrop(ev) {
  ev.preventDefault();
}

function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
```

Drag the W3Schools image into the rectangle:

<img id="drag1" src="img_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">

</body>

</html>

7 Design Employee registartion form using HTML5 tags and use input type- text, radio, date, checkbox, number, range, color, email, search, url, tel, submit.

```
<!DOCTYPE html>
<html>
<head>
 <title>Employee Registration Form</title>
</head>
<body>
 <h2>Employee Registration Form</h2>
 <form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required><br><br>

  <label for="gender">Gender:</label>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br><br>

  <label for="dob">Date of Birth:</label>
  <input type="date" id="dob" name="dob" required><br><br>

  <label for="department">Department:</label>
  <input type="checkbox" id="dept1" name="department" value="dept1">
  <label for="dept1">Department 1</label>
  <input type="checkbox" id="dept2" name="department" value="dept2">
  <label for="dept2">Department 2</label>
  <input type="checkbox" id="dept3" name="department" value="dept3">
  <label for="dept3">Department 3</label><br><br>
```

```
    <label for="experience">Experience (in years):</label>
    <input type="number" id="experience" name="experience" min="0" max="50" step="1"><br><br>

    <label for="salary">Salary:</label>
    <input type="range" id="salary" name="salary" min="0" max="100000" step="1000"><br><br>

    <label for="color">Favorite Color:</label>
    <input type="color" id="color" name="color"><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="search">Search:</label>
    <input type="search" id="search" name="search"><br><br>

    <label for="website">Website:</label>
    <input type="url" id="website" name="website"><br><br>

    <label for="phone">Phone Number:</label>
    <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"><br><br>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

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

HTML file (index.html):

```
<!DOCTYPE html>
<html>
<head>
  <title>External Styles</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Welcome to External Styles</h1>
  <p>This text is styled externally.</p>
  <div class="box">This is a styled box.</div>
</body>
</html>
```

Style.css

```
/* styles.css */
body {
  font-family: Arial, sans-serif;
  background-color: #f5f5f5;
}

h1 {
  color: #333;
  text-align: center;
}

p {
```

```
  font-size: 18px;
  color: #666;
}

.box {
  width: 200px;
  height: 200px;
  background-color: #f9f9f9;
  border: 1px solid #ccc;
  margin: 20px auto;
  text-align: center;
  line-height: 200px;
  opacity: 0.8;
}
```

Internal stylesheet

```
<!DOCTYPE html>
<html>
<head>
  <title>Internal Styles</title>
</head>
<body>
  <h1>Welcome to Internal Styles</h1>
  <p style="font-family: Arial, sans-serif; color: #333; text-align: center;">This text is styled internally.
</p>
  <div class="box" style="width: 200px; height: 200px; background-color: #f9f9f9; border: 1px solid
#ccc; margin: 20px auto; text-align: center; line-height: 200px; opacity: 0.8;">This is a styled box.
</div>
</body>
</html>
```

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

```
<!DOCTYPE html>
<html>
<head>
 <title>Transformation Example</title>
 <style>
  .box {
   width: 100px;
   height: 100px;
   background-color: red;
   margin: 50px;
   transform-origin: center;
   transition: transform 0.5s ease;
  }

  .box:hover {
   transform: translate(50px, 50px) rotate(45deg) scale(1.5);
  }
 </style>
</head>
<body>
 <h2>Transformation Example</h2>
 <div class="box"></div>
</body>
</html>
```

10 Program to show current date and time using user defined module in node.js

```
const dateObject = new Date();

const date = (`0 ${dateObject.getDate()}`).slice(-2);

const month = (`0 ${dateObject.getMonth() + 1}`).slice(-2);

const year = dateObject.getFullYear();

const hours = dateObject.getHours();

const minutes = dateObject.getMinutes();

const seconds = dateObject.getSeconds();
console.log('Disha Dharmadhikari 22027')

console.log(`${year}-${month}-${date} ${hours}:${minutes}:${seconds}`);
```

11 Program using built-in modules in node.js to split the query string into readable parts.

```
const querystring = require('querystring');

// Sample query string
const queryString = 'name=Disha&roll=22027&age=22';

// Parsing the query string
const parsedQuery = querystring.parse(queryString);

// Accessing the individual parts
const name = parsedQuery.name;
const roll = parsedQuery.roll;
const age = parsedQuery.age;


// Output the results
console.log('Name:', name);
console.log('Roll No:', roll);
console.log('Age:', age);
```

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

```
const _ = require('lodash');

// Sample string
const inputString = 'Welcome';

// Convert to lowercase
const lowercaseString = _.toLower(inputString);

// Convert to uppercase
const uppercaseString = _.toUpper(inputString);

// Output the results
console.log('Lowercase:', lowercaseString);
console.log('Uppercase:', uppercaseString);
```

13 Write a program to create a calculator using Node JS. (Install and configure Node JS and Server)

```
const argvs = process.argv

const argv = argvs.slice(2)

const operation = argv[0]

const operator1 = parseInt(argv[1])

const operator2 = parseInt(argv[2])


if (operation === 'add') {

	console.log(operation + ' is '

		+ (operator1 + operator2));

}

if (operation === 'subtract') {

	console.log(operation + ' is '

		+ (operator1 - operator2));

}

if (operation === 'multiply') {

	console.log(operation + ' is '

		+ (operator1 * operator2));

}

if (operation === 'divide') {

	console.log(operation + ' is '

		+ (operator1 / operator2));

}
```

14 Write program for Form validation in Angular.

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
@Component({
 selector: 'app-your-component',
 template: `
  <form [formGroup]="myForm" (ngSubmit)="onSubmit()">
   <input type="text" formControlName="firstName" placeholder="First Name">
   <input type="text" formControlName="lastName" placeholder="Last Name">
   <input type="email" formControlName="email" placeholder="Email">
   <input type="password" formControlName="password" placeholder="Password">
   <button type="submit" [disabled]="myForm.invalid">Submit</button>
  </form>`,
 styles: []
})
export class YourComponent {
 myForm: FormGroup;
 constructor(private formBuilder: FormBuilder) {
  this.myForm = this.formBuilder.group({
   firstName: ['', Validators.required],
   lastName: ['', Validators.required],
   email: ['', [Validators.required, Validators.email]],
   password: ['', [Validators.required, Validators.minLength(8)]],
  });
 }
 onSubmit() {
  if (this.myForm.invalid) {
   return.
  }
  console.log ('Form submitted!');
  console.log(this.myForm.value);
 }
}
```

15 Program to demonstrate the ngif, ngfor, ngswitch statements.

```
import { Component } from '@angular/core';
@Component({
 selector: 'app-example',
 template: `
  <h1>ngIf Example</h1>
  <div *ngIf="showMessage">
   <p>This message is shown using ngIf directive.</p>
  </div>

  <h1>ngFor Example</h1>
  <ul>
   <li *ngFor="let item of items">{{ item }}</li>
  </ul>

  <h1>ngSwitch Example</h1>
  <div [ngSwitch]="selectedOption">
   <p *ngSwitchCase="'option1'">Option 1 is selected.</p>
   <p *ngSwitchCase="'option2'">Option 2 is selected.</p>
   <p *ngSwitchDefault>Select an option.</p>
  </div>
 `,
})
export class ExampleComponent {
 showMessage: boolean = true;
 items: string[] = ['Item 1', 'Item 2', 'Item 3'];
 selectedOption: string = 'option2';
}
```

16) Create angular project which will demonstrate the usage of component directive, structural directive, and attribute directives.

```
import { Component } from '@angular/core';
@Component({
 selector: 'app-demo',
 template: `
  <div *appCustomIf="showElement">
   <p>Content inside custom structural directive.</p>
  </div>
  <div>
   <button appCustomAttr (click)="toggleElement()">Toggle Element</button>
  </div>`
})
export class DemoComponent {
 showElement = false;
 toggleElement() {
  this.showElement = !this.showElement;
 }
}
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { DemoComponent } from './demo.component';
@NgModule({
 declarations: [
  AppComponent,
  DemoComponent
 ],
 imports: [
  BrowserModule
 ],
 providers: [],
 bootstrap: [AppComponent]
})
export class AppModule { }
```

17 Create angular project which has HTML template and handle the click event on click of the button (Installation of Angular and Bootstrap 4 CSS Framework)

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

@Component({
 selector: 'app-root',
 template: `
  <button (click)="handleClick()">Click Me</button>
 `
})
export class AppComponent {
 handleClick() {
  console.log('Button clicked!');
  // Add your desired logic here
 }
}
```

18 Program to get uppercase and lowercase strings using AngularJS filters.

```
import { Component } from '@angular/core';
@Component({
 selector: 'app-root',
 template: `
  <h1>String Converter</h1>
  <h2>Uppercase Conversion:</h2>
  <p>{{ inputString | uppercase }}</p>
  <h2>Lowercase Conversion:</h2>
  <p>{{ inputString | lowercase }}</p>
  <input type="text" [(ngModel)]="inputString" placeholder="Enter a string">`
})
export class AppComponent {
 inputString: string = '';
}
```

19 Program for basic operations, array and user interface handling.

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

@Component({
  selector: 'app-root',
  template: `
    <h1>Basic Operations</h1>

    <button (click)="addNumbers()">Add</button>
    <p>Result: {{ 5 + 10 }}</p>

    <button (click)="subtractNumbers()">Subtract</button>
    <p>Result: {{ 15 - 7 }}</p>

    <button (click)="sortArray()">Sort</button>
    <p>Original Array: [5, 3, 9, 1, 7]</p>
    <p>Sorted Array: {{ [5, 3, 9, 1, 7].sort((a, b) => a - b) }}</p>

    <input type="text" [(ngModel)]="userInput" placeholder="Enter a value">
    <button (click)="handleUserInput()">Submit</button>
    <p>Last Submitted Value: {{ userInput }}</p>
  `
})
export class AppComponent {
  userInput: string;

  handleUserInput() {
    this.userInput = '';
  }
}
```

20 Program to demonstrate session management using various techniques.

```
import { Component } from '@angular/core';
import { SessionStorageService } from 'ngx-webstorage';
@Component({
 selector: 'app-root',
 template: `
  <h1>Session Management</h1>
  <button (click)="saveSessionData()">Save Session Data</button>
  <button (click)="retrieveSessionData()">Retrieve Session Data</button>
  <button (click)="removeSessionData()">Remove Session Data</button>
  <button (click)="clearAllSessionData()">Clear All Session Data</button>
 `
})
export class AppComponent {
 constructor(private sessionStorage: SessionStorageService) {}

 saveSessionData() {
  this.sessionStorage.store('username', 'John');
 }

 retrieveSessionData() {
  const username = this.sessionStorage.retrieve('username');
 }

 removeSessionData() {
  this.sessionStorage.clear('username');
 }

 clearAllSessionData() {
  this.sessionStorage.clear();
 }
}
```

21 Program to insert 10 records in database and read any random data using laravel framework.

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Route;

class RecordsSeeder extends Seeder
{
   public function run()
   {
      // Step 1: Migration
      if (!Schema::hasTable('records')) {
         Schema::create('records', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->timestamps();
         });
      }

      // Step 2: Seeder
      $records = [];
      for ($i = 1; $i <= 10; $i++) {
         $records[] = [
            'name' => 'User ' . $i,
            'email' => 'user' . $i . '@example.com',
            'created_at' => now(),
            'updated_at' => now()
         ];
      }
```

```
        DB::table('records')->insert($records);

        // Step 3: Route
        Route::get('/random', function () {
            $randomRecord = DB::table('records')->inRandomOrder()->first();
            return $randomRecord;
        });
    }
}
$seeder = new RecordsSeeder();
$seeder->run();
```

22 Program to print "Hello World" in PHP using echo and print statement.

```
<?php
// Using echo statement
echo "Hello World"; // Output: Hello World

// Using print statement
print "Hello World"; // Output: Hello World
?>
```

23 Program to demostarte variables in PHP. Use local, global and static keywords to access variables.

```
<?php
// Global variable
$globalVar = "I am a global variable";
function testFunction() {
    // Local variable
    $localVar = "I am a local variable";
    // Accessing global variable
    global $globalVar;
    // Printing local and global variables
    echo "Local variable: " . $localVar . "<br>";
```

```
    echo "Global variable: " . $globalVar . "<br>";
}

testFunction();

// Static variable
function countCalls() {
    static $counter = 0;
    $counter++;

    echo "Function called " . $counter . " times.<br>";
}
countCalls();
countCalls();
countCalls();
?>
```

24 Write script to demonstrate superglobals (predefined variable) in PHP.

```
<?php
// Using $_SERVER superglobal
echo "Server IP Address: " . $_SERVER['SERVER_ADDR'] . "<br>";
echo "Server Name: " . $_SERVER['SERVER_NAME'] . "<br>";
echo "Server Software: " . $_SERVER['SERVER_SOFTWARE'] . "<br>";

// Using $_GET superglobal
echo "Name: " . $_GET['name'] . "<br>";
echo "Age: " . $_GET['age'] . "<br>";

// Using $_POST superglobal
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    echo "Username: " . $_POST['username'] . "<br>";
```

```
    echo "Password: " . $_POST['password'] . "<br>";
}

// Using $_COOKIE superglobal
echo "Cookie Value: " . $_COOKIE['cookie_name'] . "<br>";

// Using $_SESSION superglobal
session_start();
$_SESSION['username'] = 'John';
echo "Session Value: " . $_SESSION['username'] . "<br>";

// Using $_FILES superglobal
if ($_FILES) {
    $file = $_FILES['file'];
    echo "Uploaded File Name: " . $file['name'] . "<br>";
    echo "Uploaded File Size: " . $file['size'] . " bytes<br>";
    echo "Uploaded File Type: " . $file['type'] . "<br>";
}
?>
```

25 Write PHP Script to demonstrate Constants.

```
<?php
// Define a constant
define("GREETING", "Hello, World!");

// Print the constant value
echo GREETING . "<br>";

// Constants are case-sensitive
define("greeting", "Hi, there!");

// Print the case-sensitive constant value
echo greeting . "<br>";
```

```
// Check if a constant is defined
if (defined("GREETING")) {
    echo "GREETING constant is defined.<br>";
} else {
    echo "GREETING constant is not defined.<br>";
}

// Redefining a constant will result in an error
// define("GREETING", "Hello, again!"); // Uncomment this line to see the error
?>
```

26 Program to demonstate concept of $$ variable in PHP.

```
<?php
$variable = "name";
$name = "John";
echo $$variable; // Output: John
?>
```

27. PHP Script to demonstrate different operators in PHP:

```
```php
<?php
$a = 10;
$b = 5;
echo $a + $b; // Addition
echo $a - $b; // Subtraction
echo $a * $b; // Multiplication
echo $a / $b; // Division
echo $a % $b; // Modulus
?>
```
```

28. PHP Script to demonstrate String functions:

```php
<?php
$str = "Hello, World!";

// String length
echo strlen($str); // Output: 13

// Substring
echo substr($str, 0, 5); // Output: Hello

// Uppercase and lowercase
echo strtoupper($str); // Output: HELLO, WORLD!
echo strtolower($str); // Output: hello, world!

// Replace
echo str_replace("World", "John", $str); // Output: Hello, John!
?>
```

29. PHP script to demonstrate Different Data Types in PHP:

```php
<?php
// String
$name = "John";

// Integer
$age = 25;

// Float
$price = 9.99;
```

```
// Boolean
$isStudent = true;

// Array
$fruits = array("Apple", "Banana", "Orange");

// Null
$address = null;

// Output
echo $name . "<br>";
echo $age . "<br>";
echo $price . "<br>";
echo $isStudent . "<br>";
print_r($fruits);
echo "<br>";
var_dump($address);
?>
```

30. PHP script to demonstrate Indexed Arrays in PHP:

```php
<?php
$fruits = array("Apple", "Banana", "Orange");

echo $fruits[0]; // Output: Apple
echo $fruits[1]; // Output: Banana
echo $fruits[2]; // Output: Orange

// Loop through the array
foreach ($fruits as $fruit) {
    echo $fruit . "<br>";
}
```

```
?>
```

31. PHP script to demonstrate Associative Arrays in PHP:

```
```php
<?php
$student = array(
    "name" => "John",
    "age" => 20,
    "grade" => "A"
);

echo $student["name"]; // Output: John
echo $student["age"]; // Output: 20
echo $student["grade"]; // Output: A

// Loop through the array
foreach ($student as $key => $value) {
    echo $key . ": " . $value . "<br>";
}
?>
```

32. PHP script to demonstrate Multidimensional Arrays in PHP:

```
```php
<?php
$students = array(
    array("John", 20, "A"),
    array("Jane", 22, "B"),
    array("David", 18, "C")
);

echo $students[0][0]; // Output: John
```

```
echo $students[1][1]; // Output: 22
echo $students[2][2]; // Output: C

// Loop through the array
foreach ($students as $student) {
    foreach ($student as $value) {
        echo $value . " ";
    }
    echo "<br>";
}
?>
```

33. PHP script to demonstrate sort function Arrays in PHP:

```
```php
<?php
$fruits = array("Orange", "Apple", "Banana");

sort($fruits);

foreach ($fruits as $fruit) {
    echo $fruit . "<br>";
}
?>
```

34. PHP script to find leap year:

```
```php
<?php
$year = 2024;

if (($year % 4 == 0 && $year % 100 != 0) || $year % 400 == 0) {
    echo $year . " is a leap year";
```

```
} else {
    echo $year . " is not a leap year";
}
?>
```

35 Write PHP script for retrieving top 10 records from database. (Assume suitable data)

```
<?php
// Database connection settings
$host = "localhost"
$username = "your_username";
$password = "your_password";
$database = "your_database";
// Establish database connection
$conn = new mysqli($host, $username, $password, $database);
// Check for connection errors
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// Query to retrieve top 10 records
$query = "SELECT * FROM your_table_name ORDER BY column_name DESC LIMIT 10";

// Execute the query
$result = $conn->query($query);

// Check if any records were found
if ($result->num_rows > 0) {
   // Output data for each row
   while ($row = $result->fetch_assoc()) {
      echo "ID: " . $row["id"] . " | Name: " . $row["name"] . " | Email: " . $row["email"] . "<br>";
   }
} else {
   echo "No records found.";
}

// Close the database connection
$conn->close();
?>
```

36 Write PHP code to display students belongs to management department and age is in between 21-30 years and store found records into another table. (Assume suitable table structure)

```
<?php

// Database connection settings

$host = "localhost";

$username = "your_username";

$password = "your_password";

$database = "your_database";


// Establish database connection

$conn = new mysqli($host, $username, $password, $database);


// Check for connection errors

if ($conn->connect_error) {

   die("Connection failed: " . $conn->connect_error);

}


// Query to retrieve students from management department and age between 21-30 years

$query = "SELECT * FROM students WHERE department = 'management' AND age BETWEEN 21 AND
30";


// Execute the query

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


// Check if any records were found

if ($result->num_rows > 0) {

   // Prepare the insert statement for the filtered students

   $insertQuery = "INSERT INTO filtered_students (name, department, age) VALUES ";


   // Build the values to be inserted

   $values = array();

   while ($row = $result->fetch_assoc()) {

      $name = $row["name"];

      $department = $row["department"];

      $age = $row["age"];

      $values[] = "('$name', '$department', $age)";
```

```
    }

    // Combine the values and insert them into the filtered_students table
    $insertQuery .= implode(", ", $values);
    if ($conn->query($insertQuery) === TRUE) {
        echo "Records inserted successfully.";
    } else {
        echo "Error inserting records: " . $conn->error;
    }
} else {
    echo "No records found.";
}

// Close the database connection
$conn->close();
?>
```

37 Write PHP program to fill online form for AADHAR card registration (Design registration form with suitable fields) and insert into database. Write PHP script to display details on different pages using $_GET and $_POST.

```
<!DOCTYPE html>
<html>
<head>
    <title>AADHAR Card Details</title>
</head>
<body>
    <h1>AADHAR Card Details</h1>
    <?php
    // Database connection settings
    $host = "localhost";
    $username = "your_username";
    $password = "your_password";
    $database = "your_database";

    // Establish database connection
    $conn = new mysqli($host, $username, $password, $database);
```

```
    // Check for connection errors
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    if (isset($_GET["id"])) {
        $id = $_GET["id"];

        // Retrieve the details from the database based on the ID
        $selectQuery = "SELECT * FROM aadhar_registration WHERE id = $id";

        // Execute the query
        $result = $conn->query($selectQuery);

        // Check if any records were found
        if ($result->num_rows > 0) {
            $row = $result->fetch_assoc();

            // Display the details
            echo "Name: " . $row["name"] . "<br>";
            echo "Email: " . $row["email"] . "<br>";
            echo "Address: " . $row["address"] . "<br>";
            echo "Phone: " . $row["phone"] . "<br>";
        } else {
            echo "No records found.";
        }
    }

    // Close the database connection
    $conn->close();
    ?>
</body>
</html>
```

38 Write PHP script to update specific record in database. Assume student table with required fields in database. May update address.

```php
<?php
// Database connection settings
$host = "localhost";
$username = "your_username";
$password = "your_password";
$database = "your_database";

// Establish database connection
$conn = new mysqli($host, $username, $password, $database);

// Check for connection errors
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Update the address of a specific student
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $studentId = $_POST["student_id"];
    $newAddress = $_POST["new_address"];

    $updateQuery = "UPDATE student SET address = '$newAddress' WHERE id = $studentId";

    if ($conn->query($updateQuery) === TRUE) {
        echo "Address updated successfully for student ID: " . $studentId;
    } else {
        echo "Error updating address: " . $conn->error;
    }
}

// Close the database connection
$conn->close();
?>
```

39 Write PHP script to create and retrieve cookie and modify a cookie value. Check if cookies are enabled.

```php
<?php
// Function to check if cookies are enabled
function areCookiesEnabled()
{
   setcookie('cookie_test', 'test', time() + 60, '/');
   if (isset($_COOKIE['cookie_test']) && $_COOKIE['cookie_test'] === 'test') {
      // Cookies are enabled
      setcookie('cookie_test', '', time() - 3600, '/');
      return true;
   } else {
      // Cookies are disabled
      return false;
   }
}

// Check if cookies are enabled
if (areCookiesEnabled()) {
   echo "Cookies are enabled.<br><br>";

   // Check if the cookie exists and retrieve its value
   if (isset($_COOKIE['my_cookie'])) {
      $cookieValue = $_COOKIE['my_cookie'];
      echo "Cookie value: " . $cookieValue . "<br>";
   } else {
      echo "Cookie not set.<br>";
   }
   // Set or modify the cookie value
   $newValue = "New Value";
   setcookie('my_cookie', $newValue, time() + 86400, '/');
   echo "Cookie has been set/modified with value: " . $newValue;
} else {
   echo "Cookies are disabled.";
}?>
```

40 Write a PHP script to Create a function with return value.

```php
<?php
// Function that calculates the sum of two numbers
function calculateSum($num1, $num2)
{
   $sum = $num1 + $num2;
   return $sum;
}
// Call the function and store the return value in a variable
$result = calculateSum(5, 3);
// Output the result
echo "The sum is: " . $result;
?>
```