Q1. write a program to show current data and time using defined module in node js.

Ans :

// Importing the built-in module 'fs' (file system)

const fs = require('fs');

// Function to get current date and time

function getCurrentDateTime() {

// Creating a new Date object

const currentDate = new Date();

// Formatting date and time

const formattedDateTime = currentDate.toLocaleString();

return formattedDateTime;

}

// Main function to execute the program

function main() {

// Get current date and time

const currentDateTime = getCurrentDateTime();

// Print current date and time

console.log('Current Date and Time:', currentDateTime);

}

// Call the main function to execute the program

main();

Q2. write a php program to store the username in a cookie and check whether the user has successfully logged in or not

Ans:

<?php

// Function to set a cookie with the username

function setCookieUsername($username) {

// Set cookie with username, expiry time is set to 1 hour (3600 seconds)

setcookie("username", $username, time() + 3600, "/");

}

// Function to check if the user is logged in

function isLoggedIn() {

// Check if the username cookie is set

return isset($\_COOKIE['username']);

}

// Main logic

// Check if the form is submitted

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

// Get the username from the form

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

// Store the username in a cookie

setCookieUsername($username);

// Redirect to a logged-in page

header("Location: logged\_in.php");

exit;

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Login Page</title>

</head>

<body>

<h2>Login</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

Username: <input type="text" name="username"><br><br>

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

</form>

</body>

</html>

Q3. Write a program in NodeJS to perform file CRUD operation by using fs module.

Ans :

const fs = require('fs');

const path = require('path');

// Function to create a new file

function createFile(filePath, data, callback) {

fs.writeFile(filePath, data, (err) => {

if (err) {

callback(err);

} else {

callback(null, 'File created successfully.');

}

});

}

// Function to read a file

function readFile(filePath, callback) {

fs.readFile(filePath, 'utf8', (err, data) => {

if (err) {

callback(err);

} else {

callback(null, data);

}

});

}

// Function to update a file

function updateFile(filePath, newData, callback) {

fs.writeFile(filePath, newData, (err) => {

if (err) {

callback(err);

} else {

callback(null, 'File updated successfully.');

}

});

}

// Function to delete a file

function deleteFile(filePath, callback) {

fs.unlink(filePath, (err) => {

if (err) {

callback(err);

} else {

callback(null, 'File deleted successfully.');

}

});

}

// Example usage:

const filePath = path.join(\_\_dirname, 'example.txt');

// Create a new file

createFile(filePath, 'Hello, world!', (err, message) => {

if (err) {

console.error('Error creating file:', err);

} else {

console.log(message);

// Read the file

readFile(filePath, (err, data) => {

if (err) {

console.error('Error reading file:', err);

} else {

console.log('File content:', data);

// Update the file

updateFile(filePath, 'Updated content!', (err, message) => {

if (err) {

console.error('Error updating file:', err);

} else {

console.log(message);

// Read the updated file

readFile(filePath, (err, data) => {

if (err) {

console.error('Error reading file:', err);

} else {

console.log('Updated file content:', data);

// Delete the file

deleteFile(filePath, (err, message) => {

if (err) {

console.error('Error deleting file:', err);

} else {

console.log(message);

}

});

}

});

}

});

}

});

}

});

Q4. Write a PHP script to display employees belongs to Sales department and salary is in between 50000 to 90000 and store found records into another table.

Ans :

<?php

// Database connection

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "your\_database";

// Create connection

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

// Check connection

if ($conn->connect\_error) {

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

}

// Select employees from the Sales department with salary between 50000 and 90000

$sql = "SELECT \* FROM employees WHERE department = 'Sales' AND salary BETWEEN 50000 AND 90000";

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

// Check if there are any results

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

// Store found records into another table

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

$id = $row["id"];

$name = $row["name"];

$department = $row["department"];

$salary = $row["salary"];

// Insert into another table (change the table name and columns according to your database schema)

$insert\_sql = "INSERT INTO another\_table (id, name, department, salary) VALUES ('$id', '$name', '$department', '$salary')";

if ($conn->query($insert\_sql) === TRUE) {

echo "Record inserted successfully<br>";

} else {

echo "Error inserting record: " . $conn->error . "<br>";

}

}

} else {

echo "No employees found in the Sales department with salary between 50000 and 90000<br>";

}

// Close connection

$conn->close();

?>

**Q5. Write a PHP script to design Employee Registration form. Insert 5 records in database and display all the inserted records on new page.**

**Ans :**

**employee\_registration\_form.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Employee Registration Form</title>

</head>

<body>

<h2>Employee Registration Form</h2>

<form action="insert\_employee.php" method="POST">

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

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

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

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

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

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

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

</form>

</body>

</html>

**insert\_employee.php:**

<?php

// Database connection

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database";

// Create connection

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

// Check connection

if ($conn->connect\_error) {

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

}

// Retrieve form data

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

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

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

// Insert data into the database

$sql = "INSERT INTO employees (name, designation, salary) VALUES ('$name', '$designation', '$salary')";

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

echo "New record inserted successfully<br>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

<a href="display\_employees.php">View all employees</a>

**display\_employees.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Employee Records</title>

</head>

<body>

<h2>Employee Records</h2>

<table border="1">

<tr>

<th>Name</th>

<th>Designation</th>

<th>Salary</th>

</tr>

<?php

// Database connection

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database";

// Create connection

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

// Check connection

if ($conn->connect\_error) {

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

}

// Fetch data from the database

$sql = "SELECT \* FROM employees";

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

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

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

echo "<tr><td>" . $row["name"] . "</td><td>" . $row["designation"] . "</td><td>" . $row["salary"] . "</td></tr>";

}

} else {

echo "0 results";

}

$conn->close();

?>

</table>

</body>

</html>

Q6. Create an Angular program which will demonstrate the use of ngswitch directive.

Ans :

**switch-demo.component.html**

<div [ngSwitch]="color">

<p \*ngSwitchCase="'red'">You selected Red</p>

<p \*ngSwitchCase="'blue'">You selected Blue</p>

<p \*ngSwitchCase="'green'">You selected Green</p>

<p \*ngSwitchDefault>Invalid color selection</p>

</div>

<select [(ngModel)]="color">

<option value="red">Red</option>

<option value="blue">Blue</option>

<option value="green">Green</option>

</select>

**switch-demo.component.ts**

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

@Component({

selector: 'app-switch-demo',

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

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

})

export class SwitchDemoComponent {

color: string = 'red'; // Default color selection

}