ASSIGNMENT

V.MAHENDRA-VTU21880

**TASK-6**

1. Construct a responsive online learning website using the Bootstrap framework that offers a variety of courses to students. Explain how the Bootstrap grid system can be utilized to create a flexible and visually appealing layout for displaying course categories and individual course modules. Discuss the implementation of Bootstrap's components, such as accordions or tabs, to organize course content and

enhance user experience.

Program:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Online Learning</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<body>

<header class="bg-primary text-white text-center py-3">

<h1>Online Learning Platform</h1>

</header>

<div class="container mt-4">

<div class="row">

<div class="col-md-3">

<h5>Categories</h5>

<ul class="list-group">

<li class="list-group-item">Web Development</li>

<li class="list-group-item">Data Science</li>

</ul>

</div>

<div class="col-md-9">

<h5>Available Courses</h5>

<div class="row">

<div class="col-md-4 mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Course 1</h5>

<a href="#" class="btn btn-primary">Enroll</a>

</div>

</div>

</div>

<div class="col-md-4 mb-4">

<div class="card">

<div class="card-body">

<h5 class="card-title">Course 2</h5>

<a href="#" class="btn btn-primary">Enroll</a>

</div>

</div>

</div>

</div

<h5 class="mt-4">Course Modules</h5>

<div class="accordion" id="modules">

<div class="card">

<div class="card-header">

<h5 class="mb-0">

<button class="btn btn-link" data-toggle="collapse" data-target="#module1">Module 1</button>

</h5>

</div>

<div id="module1" class="collapse">

<div class="card-body">Content for Module 1</div>

</div>

</div>

</div>

</div>

</div>

</div>

<footer class="bg-dark text-white text-center py-3">

<p>&copy; 2024 Online Learning</p>

</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

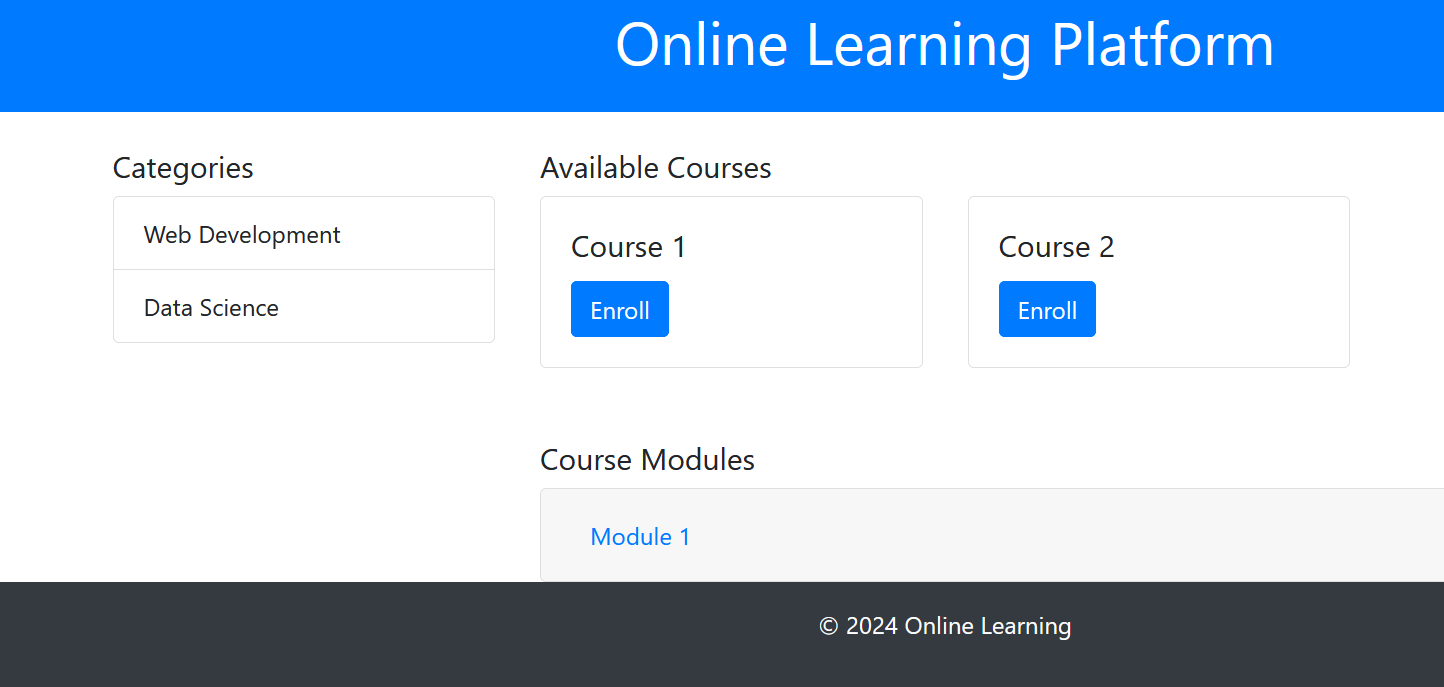
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

Output:



2. In an image gallery website, develop a page that displays a grid of thumbnail

Images. When a thumbnail is clicked, the corresponding full-size image should be displayed in a modal popup with the option to close the popup.

Program:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Gallery</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

.gallery img {

cursor: pointer;

transition: ease-in 1.2s;

width:100px;

height: 100px;

}

.gallery img:hover {

transform: scale(1.05);

width: 200px;

height:150px;

}

body{

background-color: aquamarine;

}

</style>

</head>

<body>

<div class="container mt-4">

<h1 class="text-center">Image Gallery</h1>

<div class="row gallery">

<div class="col-md-4 mb-4">

<img src="veltech.jpg" class="img-fluid" alt="Image 1" data-full="fullsize1.jpg" onclick="openModal(this)">

</div>

<div class="col-md-4 mb-4">

<img src="DOCTOR IMAGE.jpg" class="img-fluid" alt="Image 2" data-full="fullsize2.jpg" onclick="openModal(this)">

</div>

<div class="col-md-4 mb-4">

<img src="hospital images.jpg" class="img-fluid" alt="Image 3" data-full="fullsize3.jpg" onclick="openModal(this)">

</div> </div></div>

<div class="modal fade" id="imageModal" tabindex="-1" role="dialog" aria-labelledby="imageModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-body text-center">

<img id="fullImage" src="" class="img-fluid" alt="Full Image">

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

</div> </div></div></div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script>

function openModal(veltech) {

const fullImageSrc = thumbnail.getAttribute('data-full');

document.getElementById('fullImage').src = fullImageSrc;

$('#imageModal').modal('show');

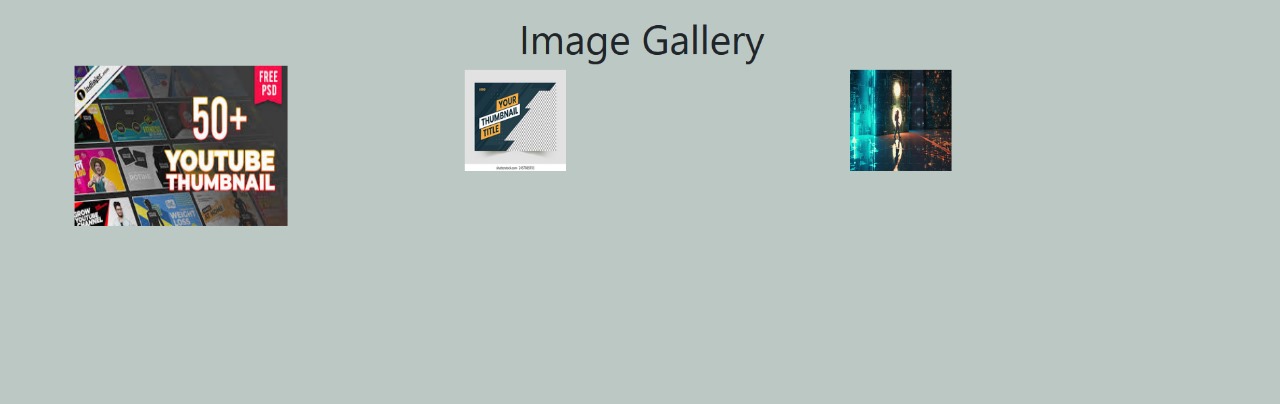
}

</script>

</body>

</html>

Output:



3. Develop a online e-commerce website where users can browse and purchase products. The website should include product listings, product details pages, user registration and login, a shopping cart, and a checkout process. You need to use

HTML, CSS, and PHP to build the website.

Program:

<?php

session\_start();

// Sample product data (in a real app, this would be fetched from a database)

$products = [

['id' => 1, 'name' => 'Product 1', 'price' => 10.00, 'image' => 'images/product1.jpg'],

['id' => 2, 'name' => 'Product 2', 'price' => 15.00, 'image' => 'images/product2.jpg'],

];

// Initialize cart

if (!isset($\_SESSION['cart'])) {

$\_SESSION['cart'] = [];

}

// Display the content based on the requested page

$page = $\_GET['page'] ?? 'home';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>E-Commerce Site</title>

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

<style>

body { font-family: Arial, sans-serif; }

header { background: #333; color: #fff; padding: 10px; text-align: center; }

nav a { margin: 0 15px; color: #fff; text-decoration: none; }

.product-list { display: flex; flex-wrap: wrap; justify-content: center; }

.product-item { margin: 15px; border: 1px solid #ccc; padding: 10px; width: 200px; }

img { max-width: 100%; }

</style>

</head>

<body><header>

<h1>My E-Commerce Store</h1>

<nav>

<a href="?page=home">Home</a>

<a href="?page=cart">Cart (<?= array\_sum($\_SESSION['cart']) ?>)</a>

</nav>

</header>

<main>

<?php if ($page === 'home'): ?>

<h2>Products</h2>

<div class="product-list">

<?php foreach ($products as $product): ?>

<div class="product-item">

<img src="d.jpg" alt="<?= $product['name'] ?>">

<h3><?= $product['name'] ?></h3>

<p>$<?= number\_format($product['price'], 2) ?></p>

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

<input type="hidden" name="product\_id" value="<?= $product['id'] ?>">

<button type="submit">Add to Cart</button>

</form>

<a href="?page=product&id=<?= $product['id'] ?>">View Details</a>

</div>

<?php endforeach; ?>

</div>

<?php elseif ($page === 'cart'): ?>

<h2>Your Cart</h2>

<ul>

<?php foreach ($\_SESSION['cart'] as $id => $quantity): ? </ul>

<a href="?page=checkout">Proceed to Checkout</a>

<?php elseif ($page === 'checkout'): ?>

<h2>Checkout</h2>

<p>Thank you for your purchase!</p>

<p>Total: $<?= number\_format(array\_sum(array\_map(function($id) { return $products[$id - 1]['price'] \* $\_SESSION['cart'][$id]; }, array\_keys($\_SESSION['cart']))), 2) ?></p>

<a href="?page=home">Continue Shopping</a>

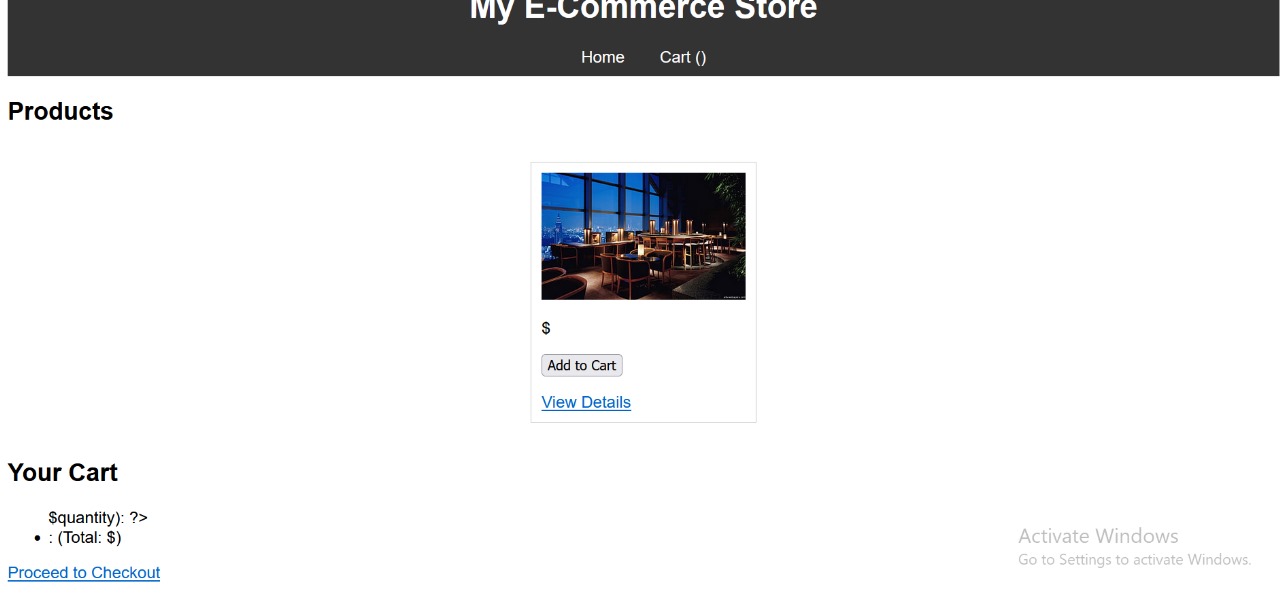
<?php endif; ?>

</main>

</body>

</html>

Output:



4.Understand the purpose and features of Seneca as a micro services framework. Discuss how Seneca helps with service discovery, communication patterns, and handling cross-cutting concerns. Use Seneca Tool to design a micro services for Health care systems.

Program:

// healthcare.js

const seneca = require('seneca')();

// User Service

seneca.add('role:user,cmd:register', function (msg, respond) {

// Logic to register a user (mock implementation)

const userId = Math.floor(Math.random() \* 1000);

respond(null, { success: true, userId: userId });

});seneca.add('role:user,cmd:login', function (msg, respond) {

// Logic to login a user (mock implementation)

respond(null, { success: true, token: 'abcd1234' });

});// Appointment Service

seneca.add('role:appointment,cmd:schedule', function (msg, respond) {

// Check user authentication (mock check)

this.act('role:user,cmd:login', {}, (err, result) => {

if (result.success) {

const appointmentId = Math.floor(Math.random() \* 1000);

respond(null, { success: true, appointmentId: appointmentId });

} else {

respond(null, { success: false, message: 'User not authenticated' });

}

});

});

// Medical Records Service

seneca.add('role:record,cmd:add', function (msg, respond) {

// Logic to add medical records (mock implementation)

const recordId = Math.floor(Math.random() \* 1000)

respond(null, { success: true, recordId: recordId });

});// Billing Service

seneca.add('role:billing,cmd:process', function (msg, respond) {

// Logic to process billing (mock implementation)

const invoiceId = Math.floor(Math.random() \* 1000);

respond(null, { success: true, invoiceId: invoiceId });

});

// Service Listener

seneca.listen({ port: 3000 }, () => {

console.log('Healthcare microservices running on port 3000');

});

// Example Interaction

// For demonstration, simulate an appointment scheduling request

seneca.act('role:appointment,cmd:schedule', {}, (err, result) => {

if (err) {

console.error(err);

} else {

console.log(result);

    }

});

5. Develop a web page which is used to track the value of Cryptocurrency.

Program:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Cryptocurrency Tracker</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

color: #333;

margin: 0;

padding: 20px;

}

header {

text-align: center;

margin-bottom: 20px;

}

#crypto-container {

display: flex;

justify-content: center;

margin-bottom: 20px;

}

#crypto-input {

padding: 10px;

font-size: 16px;

margin-right: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

#track-button {

padding: 10px;

font-size: 16px;

background-color: #28a745;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

#track-button:hover {

background-color: #218838;

}

#result {

text-align: center;

font-size: 20px;

margin-top: 20px;

}

footer {

text-align: center;

margin-top: 20px;

}

</style>

</head>

<body>

<header>

<h1>Cryptocurrency Tracker</h1>

<p>Track real-time cryptocurrency prices</p>

</header>

<main>

<div id="crypto-container">

<input type="text" id="crypto-input" placeholder="Enter cryptocurrency symbol (e.g., BTC)">

<button id="track-button">Track</button>

</div>

<div id="result"></div>

</main>

<footer>

<p>Data provided by <a href="https://coingecko.com">CoinGecko</a></p>

</footer>

<script>

document.getElementById('track-button').addEventListener('click', function() {

const cryptoSymbol = document.getElementById('crypto-input').value.toLowerCase();

const url = https://api.coingecko.com/api/v3/simple/price?ids=${cryptoSymbol}&vs\_currencies=usd;

fetch(url)

.then(response => {

if (!response.ok) {

throw new Error('Cryptocurrency not found');

}

return response.json();

})

.then(data => {

const resultDiv = document.getElementById('result');

if (data[cryptoSymbol]) {

resultDiv.innerHTML = ${cryptoSymbol.toUpperCase()} price: $${data[cryptoSymbol].usd};

} else {

resultDiv.innerHTML = 'Cryptocurrency not found. Please try again.';

}

})

.catch(error => {

document.getElementById('result').innerHTML = error.message;

});

});

</script>

</body>

</html>

Output:

