Nasagh inc.

27 / November/ 2025

Nsagh

Aromatic Goods

Marketplace

Prepared By

WEB Crafters Team

Elyas Alquwatie	ID: 142537
Hisham Alhinai	ID: 150229
Almuhannad Alhinai	ID: 144759

Instructor: Dr. Ahmad Soleimani

COMP3700 – Introduction to Web Computing

1. Bootstrap Integration Description

Complete Framework Migration

We successfully replaced all custom CSS with Bootstrap 5 utility classes while preserving the luxurious Nasagh brand identity. The migration involved systematically converting custom styling to Bootstrap's comprehensive class system, ensuring consistent design language across all pages.

Responsive Grid Implementation

All website pages now utilize Bootstrap's 12-column grid system, providing optimal viewing experiences across desktop, tablet, and mobile devices. The grid system automatically adjusts content layout based on screen size, maintaining readability and usability.

Navigation System

We implemented a responsive navigation bar that collapses into a hamburger menu on mobile devices while maintaining full functionality. The navbar includes links to all main pages and the three new pages required for this phase.

Form Styling and Validation

All forms across the website now feature Bootstrap-styled inputs, labels, and buttons. We implemented Bootstrap's validation states to provide users with immediate, clear feedback during form submission processes.

Interactive Components

The website incorporates multiple Bootstrap interactive components including modals for product details and success messages, carousels for product image galleries, and collapse elements for FAQ sections. These components enhance user engagement without compromising performance.

Typography and Utilities

We leveraged Bootstrap's extensive utility classes for consistent spacing, text formatting, and layout management. This approach eliminated the need for custom CSS while maintaining design consistency.

---

2. JavaScript Features Description

Dynamic Content Management

Object-Oriented Architecture

We implemented two constructor functions to model the core business entities: Products and Orders. The Product constructor manages inventory items with properties for identification, pricing, and stock levels. The Order constructor handles customer purchases with comprehensive tracking capabilities.

Array-Based Data Management

Two main arrays store product and order information, initialized with sample data that reflects realistic business scenarios. This array-based approach enables efficient data manipulation and retrieval operations.

Real-Time Table Generation

JavaScript functions dynamically generate and update HTML tables by iterating through the arrays. The tables include conditional formatting that changes appearance based on data states, such as low stock warnings or order status indicators.

Data Manipulation Capabilities

Add New Entries

Users can add new products and orders through intuitive forms that validate input before updating the arrays. The system automatically generates unique identifiers and timestamps for new entries.

Advanced Search Functionality

The search feature allows users to filter products and orders based on multiple criteria. The search operates in real-time, immediately updating table displays as users type their queries.

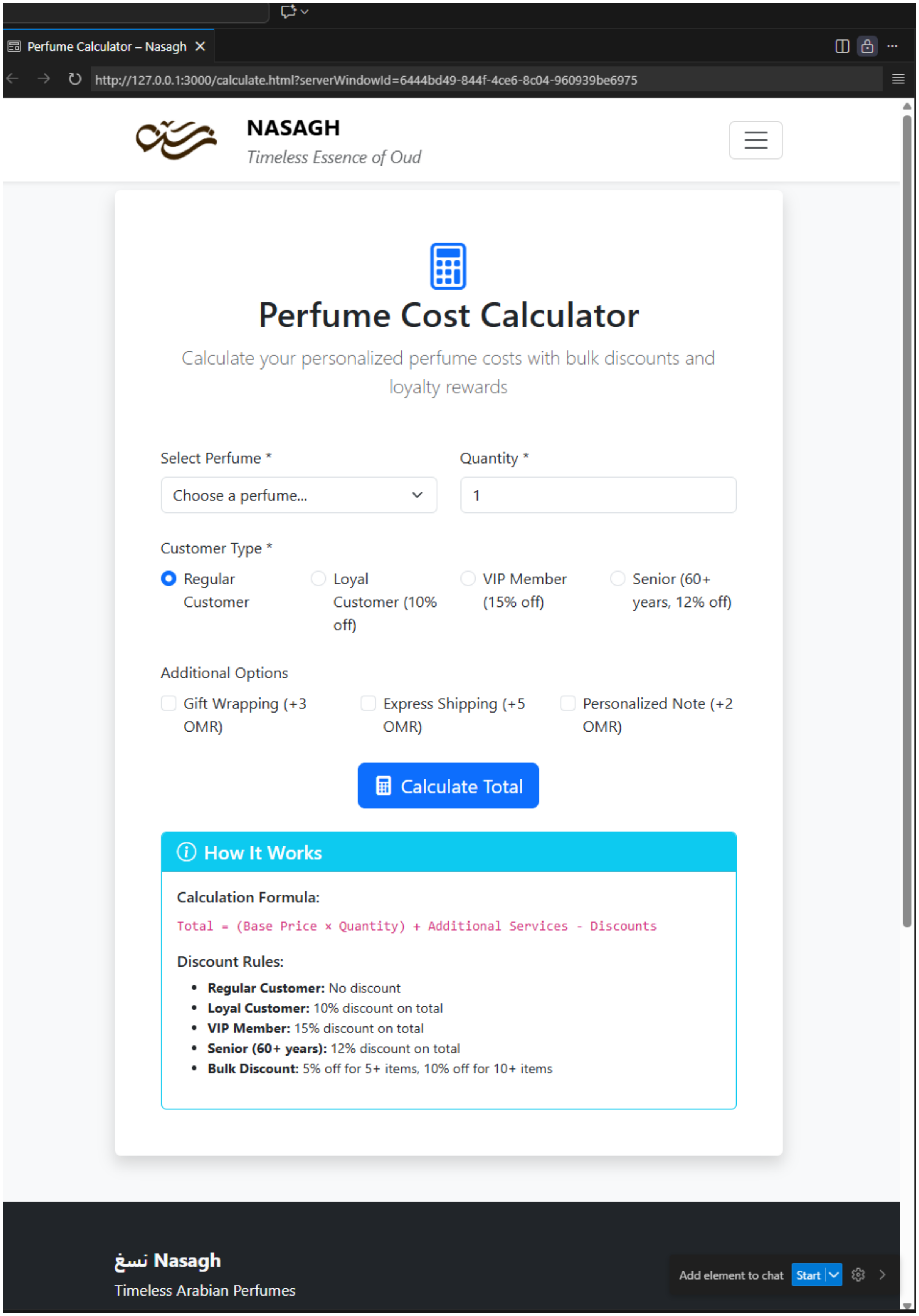
Enhanced User Interactions

Form Validation System

We implemented sophisticated validation patterns that go beyond basic HTML5 validation. The system includes custom validation for Omani phone numbers, comprehensive name validation supporting Arabic characters, and advanced email verification.

Interactive Feedback

Users receive immediate visual feedback during interactions, with success messages, error indicators, and loading states that improve the overall user experience.


Perfume Calculator – Nasagh
http://127.0.0.1:3000/calculate.html?serverWindowId=6444bd49-844f-4ce6-8c04-960939be6975
Add element to chat
Start

NASAGH

Timeless Essence of Oud

Perfume Cost Calculator

Calculate your personalized perfume costs with bulk discounts and loyalty rewards

Select Perfume *	Quantity *
Choose a perfume...	1

Customer Type *

● Regular Customer ○ Loyal Customer (10% off) ○ VIP Member (15% off) ○ Senior (60+ years, 12% off)

Additional Options

□ Gift Wrapping (+3 OMR) □ Express Shipping (+5 OMR) □ Personalized Note (+2 OMR)

How It Works

Calculation Formula:

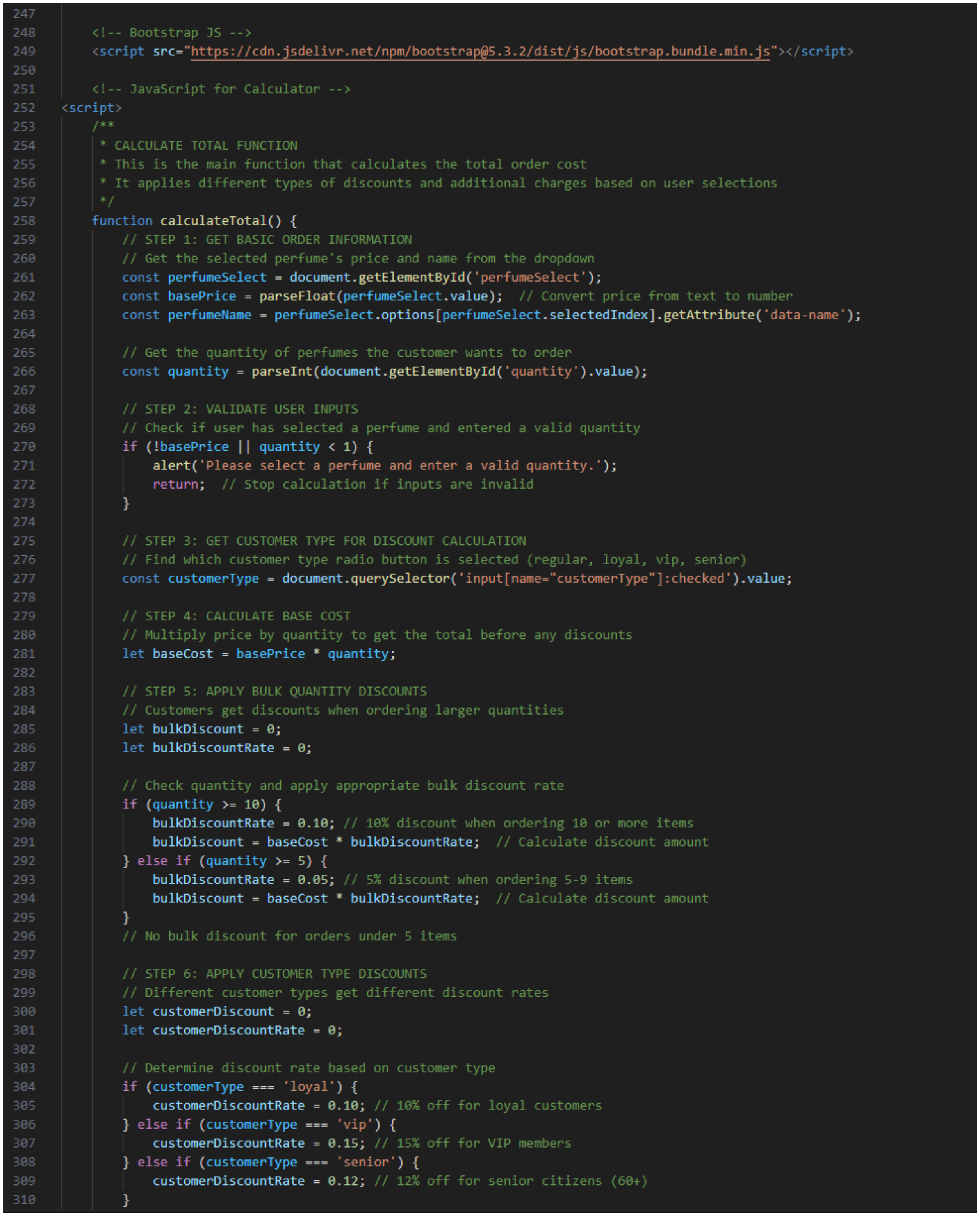
```
Total = (Base Price × Quantity) + Additional Services - Discounts
```

Discount Rules:

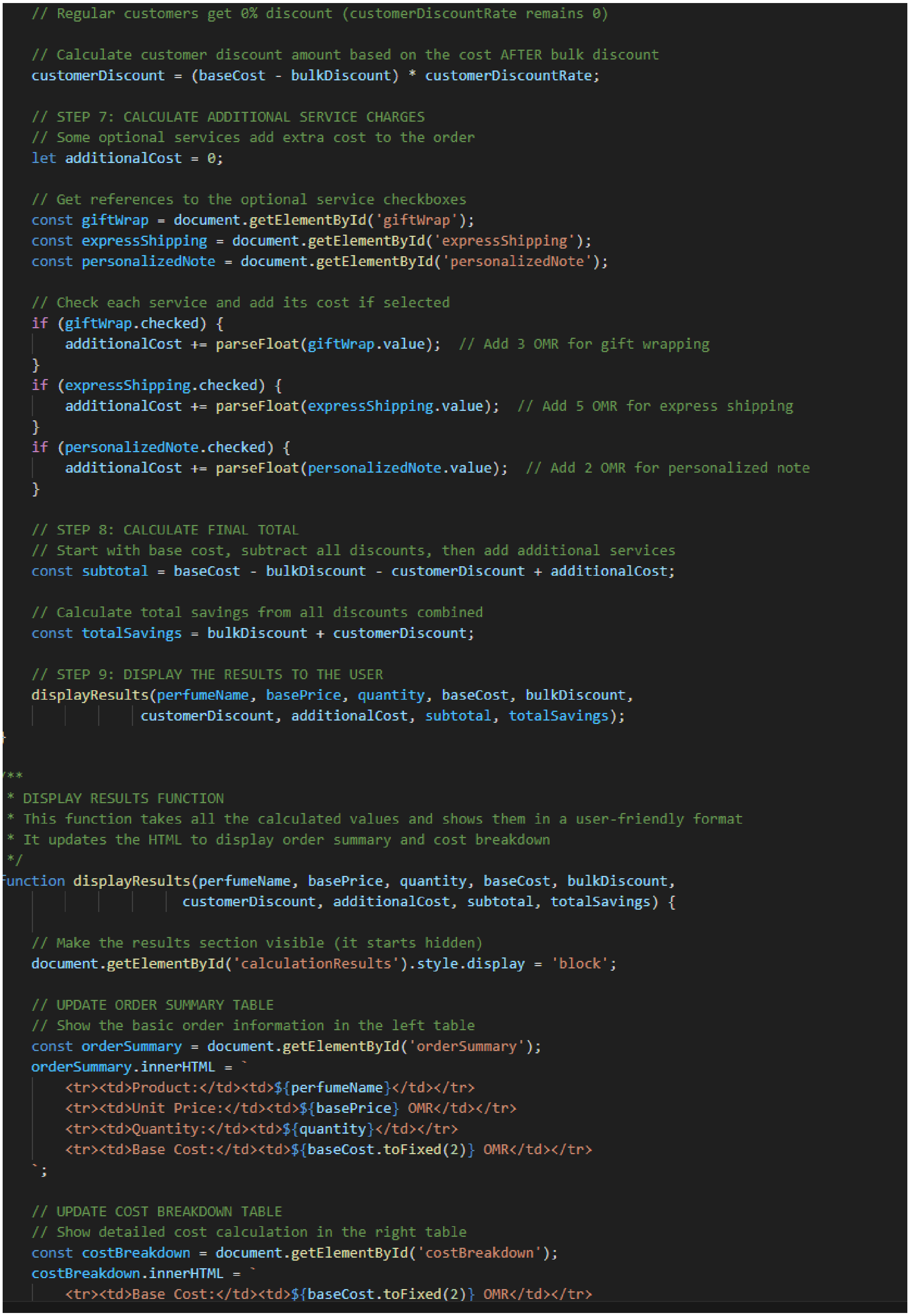
- **Regular Customer:** No discount
- **Loyal Customer:** 10% discount on total
- **VIP Member:** 15% discount on total
- **Senior (60+ years):** 12% discount on total
- **Bulk Discount:** 5% off for 5+ items, 10% off for 10+ items

نسغ Nasagh

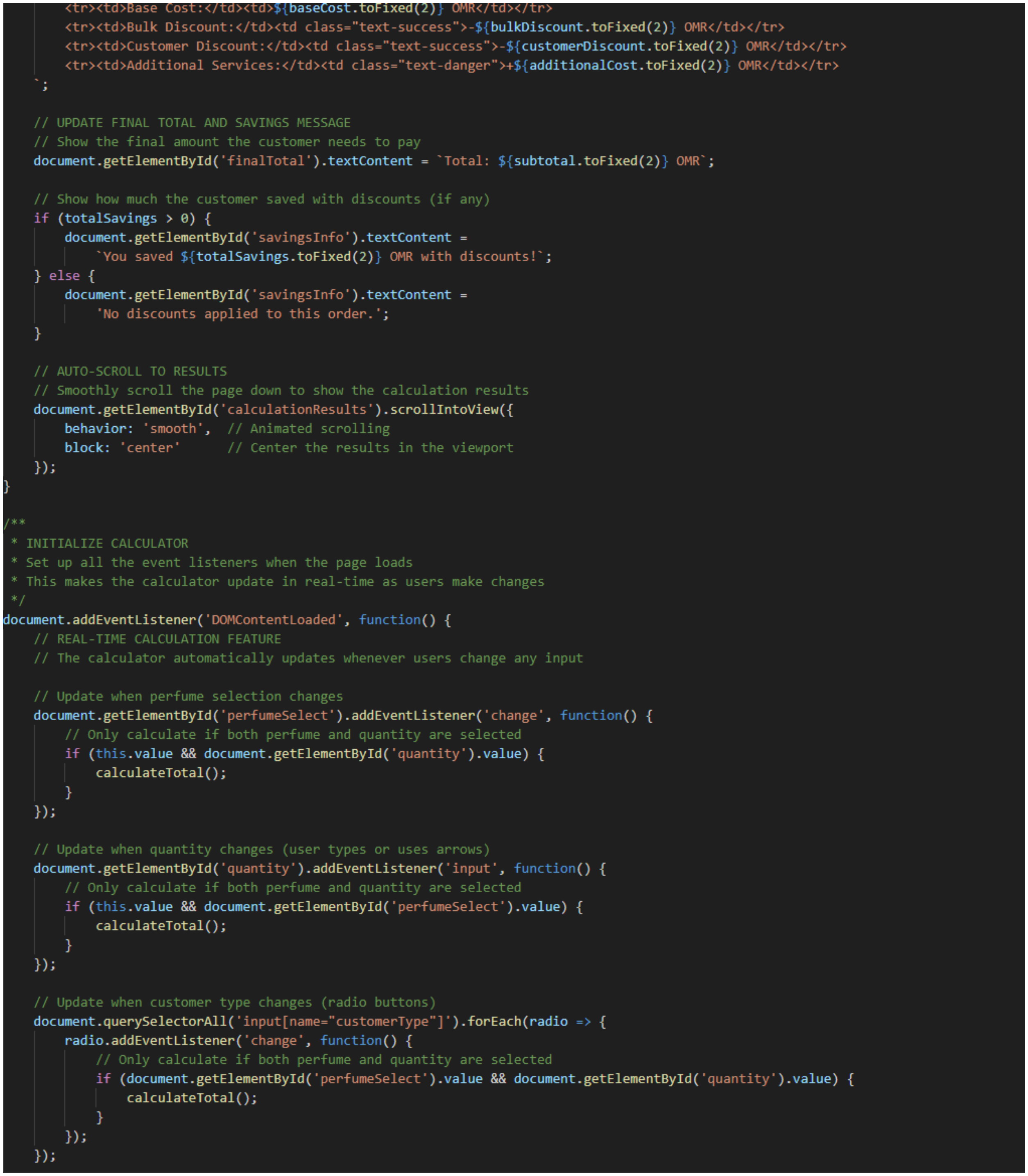
Timeless Arabian Perfumes



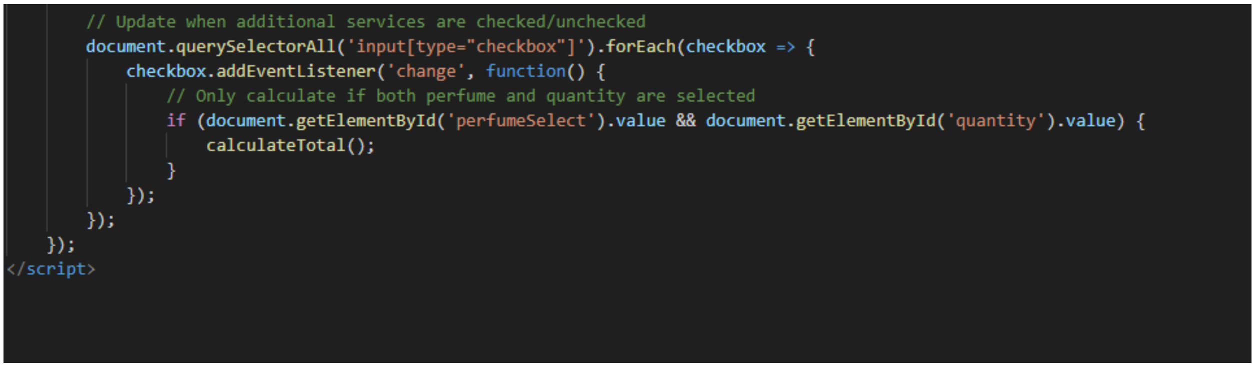
```

    <!-- Bootstrap JS -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

    <!-- JavaScript for Calculator -->
<script>
    /**
     * CALCULATE TOTAL FUNCTION
     * This is the main function that calculates the total order cost
     * It applies different types of discounts and additional charges based on user selections
     */
    function calculateTotal() {
        // STEP 1: GET BASIC ORDER INFORMATION
        // Get the selected perfume's price and name from the dropdown
        const perfumeSelect = document.getElementById('perfumeSelect');
        const basePrice = parseFloat(perfumeSelect.value);  // Convert price from text to number
        const perfumeName = perfumeSelect.options[perfumeSelect.selectedIndex].getAttribute('data-name');

        // Get the quantity of perfumes the customer wants to order
        const quantity = parseInt(document.getElementById('quantity').value);

        // STEP 2: VALIDATE USER INPUTS
        // Check if user has selected a perfume and entered a valid quantity
        if (!basePrice || quantity < 1) {
            alert('Please select a perfume and enter a valid quantity.');
            return;  // Stop calculation if inputs are invalid
        }

        // STEP 3: GET CUSTOMER TYPE FOR DISCOUNT CALCULATION
        // Find which customer type radio button is selected (regular, loyal, vip, senior)
        const customerType = document.querySelector('input[name="customerType"]:checked').value;

        // STEP 4: CALCULATE BASE COST
        // Multiply price by quantity to get the total before any discounts
        let baseCost = basePrice * quantity;

        // STEP 5: APPLY BULK QUANTITY DISCOUNTS
        // Customers get discounts when ordering larger quantities
        let bulkDiscount = 0;
        let bulkDiscountRate = 0;

        // Check quantity and apply appropriate bulk discount rate
        if (quantity >= 10) {
            bulkDiscountRate = 0.10; // 10% discount when ordering 10 or more items
            bulkDiscount = baseCost * bulkDiscountRate;  // Calculate discount amount
        } else if (quantity >= 5) {
            bulkDiscountRate = 0.05; // 5% discount when ordering 5-9 items
            bulkDiscount = baseCost * bulkDiscountRate;  // Calculate discount amount
        }
        // No bulk discount for orders under 5 items

        // STEP 6: APPLY CUSTOMER TYPE DISCOUNTS
        // Different customer types get different discount rates
        let customerDiscount = 0;
        let customerDiscountRate = 0;

        // Determine discount rate based on customer type
        if (customerType === 'loyal') {
            customerDiscountRate = 0.10; // 10% off for loyal customers
        } else if (customerType === 'vip') {
            customerDiscountRate = 0.15; // 15% off for VIP members
        } else if (customerType === 'senior') {
            customerDiscountRate = 0.12; // 12% off for senior citizens (60+)
        }
```



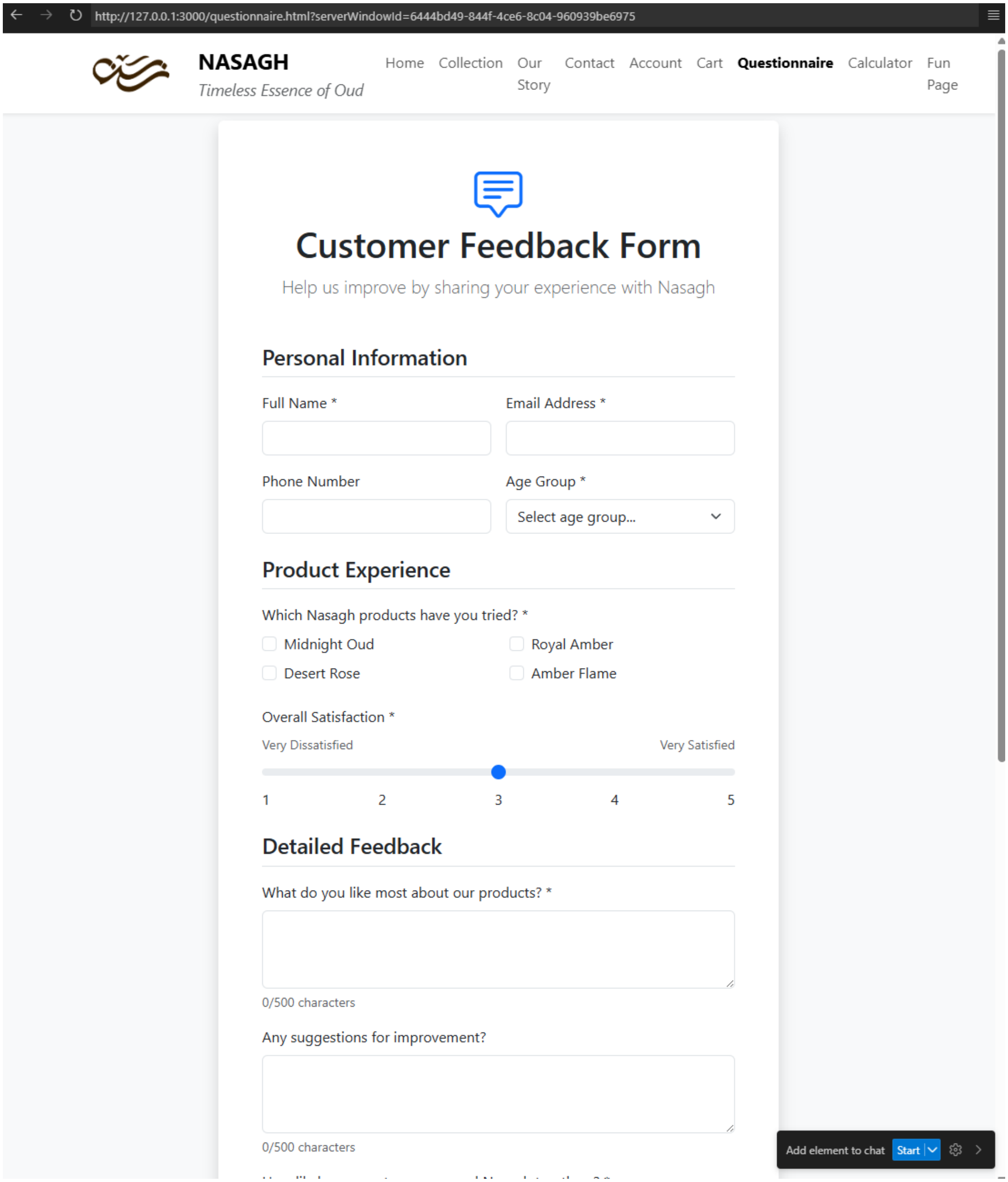
```
    // Regular customers get 0% discount (customerDiscountRate remains 0)

    // Calculate customer discount amount based on the cost AFTER bulk discount
    customerDiscount = (baseCost - bulkDiscount) * customerDiscountRate;

    // STEP 7: CALCULATE ADDITIONAL SERVICE CHARGES
    // Some optional services add extra cost to the order
    let additionalCost = 0;

    // Get references to the optional service checkboxes
    const giftWrap = document.getElementById('giftWrap');
    const expressShipping = document.getElementById('expressShipping');
    const personalizedNote = document.getElementById('personalizedNote');

    // Check each service and add its cost if selected
    if (giftWrap.checked) {
        additionalCost += parseFloat(giftWrap.value);  // Add 3 OMR for gift wrapping
    }
    if (expressShipping.checked) {
        additionalCost += parseFloat(expressShipping.value);  // Add 5 OMR for express shipping
    }
    if (personalizedNote.checked) {
        additionalCost += parseFloat(personalizedNote.value);  // Add 2 OMR for personalized note
    }

    // STEP 8: CALCULATE FINAL TOTAL
    // Start with base cost, subtract all discounts, then add additional services
    const subtotal = baseCost - bulkDiscount - customerDiscount + additionalCost;

    // Calculate total savings from all discounts combined
    const totalSavings = bulkDiscount + customerDiscount;

    // STEP 9: DISPLAY THE RESULTS TO THE USER
    displayResults(perfumeName, basePrice, quantity, baseCost, bulkDiscount,
                 customerDiscount, additionalCost, subtotal, totalSavings);
}

/**
 * DISPLAY RESULTS FUNCTION
 * This function takes all the calculated values and shows them in a user-friendly format
 * It updates the HTML to display order summary and cost breakdown
 */
function displayResults(perfumeName, basePrice, quantity, baseCost, bulkDiscount,
                       customerDiscount, additionalCost, subtotal, totalSavings) {

    // Make the results section visible (it starts hidden)
    document.getElementById('calculationResults').style.display = 'block';

    // UPDATE ORDER SUMMARY TABLE
    // Show the basic order information in the left table
    const orderSummary = document.getElementById('orderSummary');
    orderSummary.innerHTML = `
        <tr><td>Product:</td><td>${perfumeName}</td></tr>
        <tr><td>Unit Price:</td><td>${basePrice} OMR</td></tr>
        <tr><td>Quantity:</td><td>${quantity}</td></tr>
        <tr><td>Base Cost:</td><td>${baseCost.toFixed(2)} OMR</td></tr>
    `;

    // UPDATE COST BREAKDOWN TABLE
    // Show detailed cost calculation in the right table
    const costBreakdown = document.getElementById('costBreakdown');
    costBreakdown.innerHTML = `
        <tr><td>Base Cost:</td><td>${baseCost.toFixed(2)} OMR</td></tr>
```



```
        <tr><td>Base Cost:</td><td>${baseCost.toFixed(2)} OMR</td></tr>
        <tr><td>Bulk Discount:</td><td class="text-success">-${bulkDiscount.toFixed(2)} OMR</td></tr>
        <tr><td>Customer Discount:</td><td class="text-success">-${customerDiscount.toFixed(2)} OMR</td></tr>
        <tr><td>Additional Services:</td><td class="text-danger">+${additionalCost.toFixed(2)} OMR</td></tr>
    `;

    // UPDATE FINAL TOTAL AND SAVINGS MESSAGE
    // Show the final amount the customer needs to pay
    document.getElementById('finalTotal').textContent = `Total: ${subtotal.toFixed(2)} OMR`;

    // Show how much the customer saved with discounts (if any)
    if (totalSavings > 0) {
        document.getElementById('savingsInfo').textContent =
            `You saved ${totalSavings.toFixed(2)} OMR with discounts!`;
    } else {
        document.getElementById('savingsInfo').textContent =
            'No discounts applied to this order.';
    }

    // AUTO-SCROLL TO RESULTS
    // Smoothly scroll the page down to show the calculation results
    document.getElementById('calculationResults').scrollIntoView({
        behavior: 'smooth',  // Animated scrolling
        block: 'center'      // Center the results in the viewport
    });
}

/**
 * INITIALIZE CALCULATOR
 * Set up all the event listeners when the page loads
 * This makes the calculator update in real-time as users make changes
 */
document.addEventListener('DOMContentLoaded', function() {
    // REAL-TIME CALCULATION FEATURE
    // The calculator automatically updates whenever users change any input

    // Update when perfume selection changes
    document.getElementById('perfumeSelect').addEventListener('change', function() {
        // Only calculate if both perfume and quantity are selected
        if (this.value && document.getElementById('quantity').value) {
            calculateTotal();
        }
    });

    // Update when quantity changes (user types or uses arrows)
    document.getElementById('quantity').addEventListener('input', function() {
        // Only calculate if both perfume and quantity are selected
        if (this.value && document.getElementById('perfumeSelect').value) {
            calculateTotal();
        }
    });

    // Update when customer type changes (radio buttons)
    document.querySelectorAll('input[name="customerType"]').forEach(radio => {
        radio.addEventListener('change', function() {
            // Only calculate if both perfume and quantity are selected
            if (document.getElementById('perfumeSelect').value && document.getElementById('quantity').value) {
                calculateTotal();
            }
        });
    });
```



```
        // Update when additional services are checked/unchecked
        document.querySelectorAll('input[type="checkbox"]').forEach(checkbox => {
            checkbox.addEventListener('change', function() {
                // Only calculate if both perfume and quantity are selected
                if (document.getElementById('perfumeSelect').value && document.getElementById('quantity').value) {
                    calculateTotal();
                }
            });
        });
    });
</script>
```



Customer Feedback Form

Help us improve by sharing your experience with Nasagh

Personal Information

Full Name *

Email Address *

Phone Number

Age Group *

Select age group...

Product Experience

Which Nasagh products have you tried? *

□ Midnight Oud □ Royal Amber

□ Desert Rose □ Amber Flame

Overall Satisfaction *

Very Dissatisfied — Very Satisfied

1 2 ● 3 4 5

Detailed Feedback

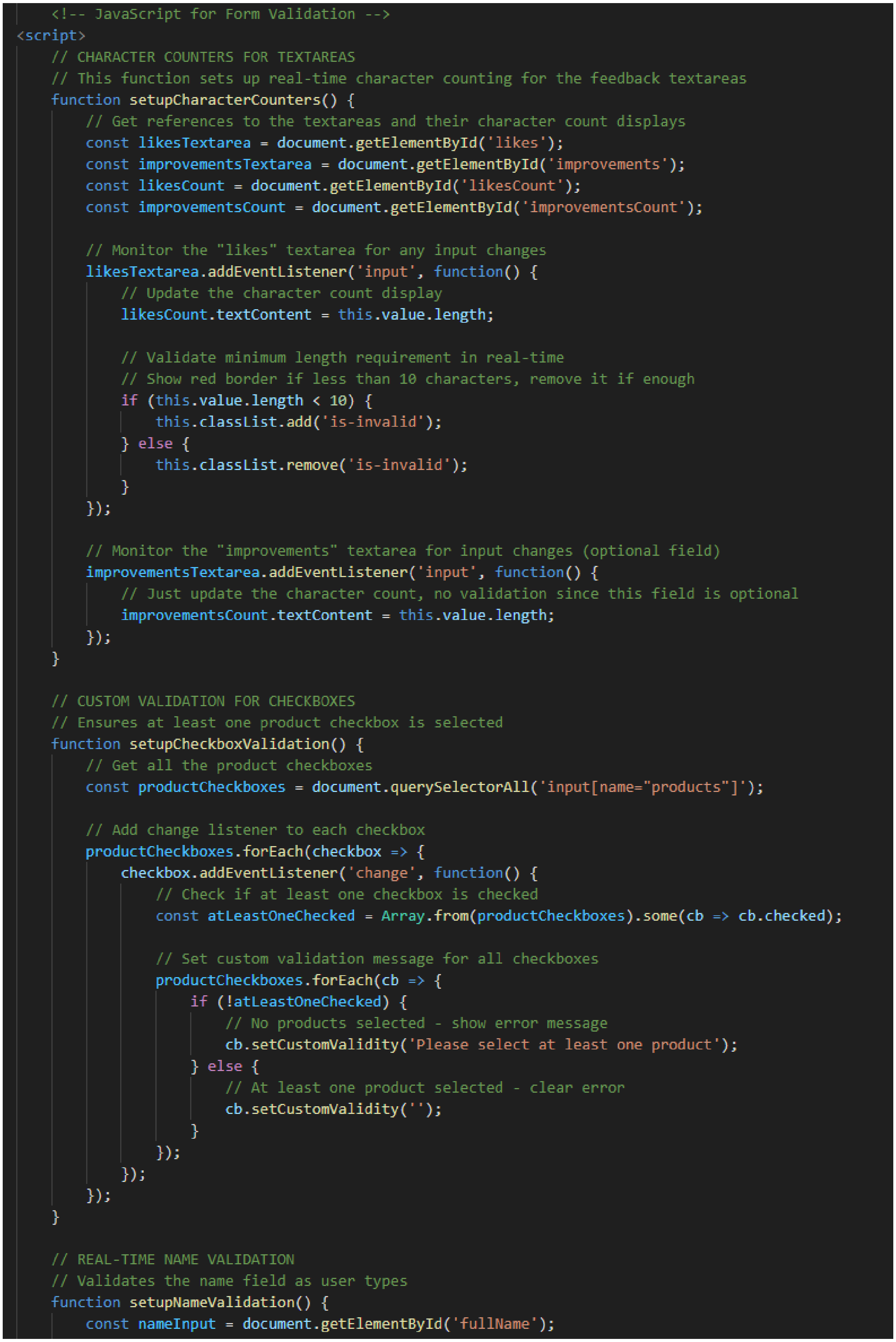
What do you like most about our products? *

0/500 characters

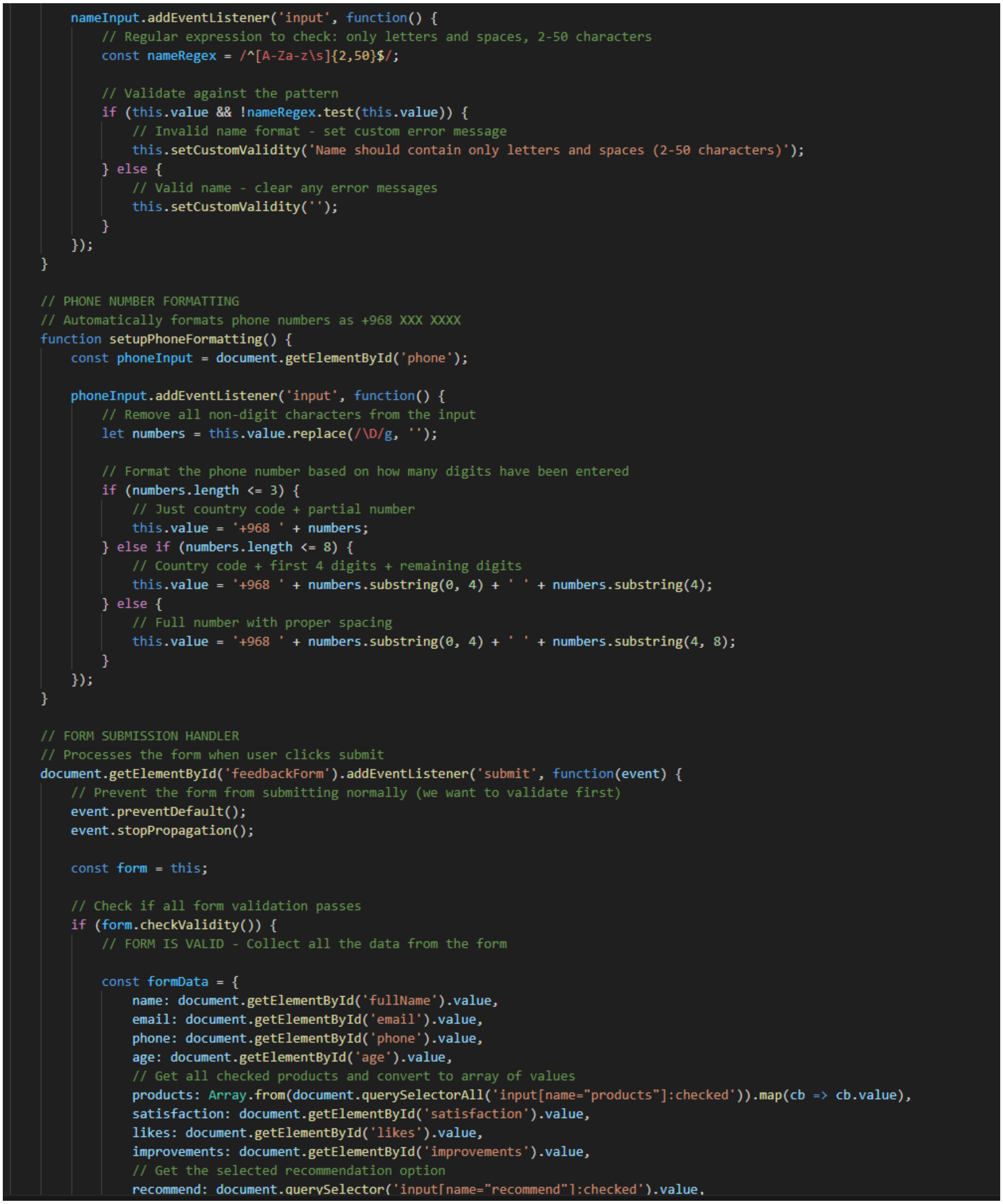
Any suggestions for improvement?

0/500 characters

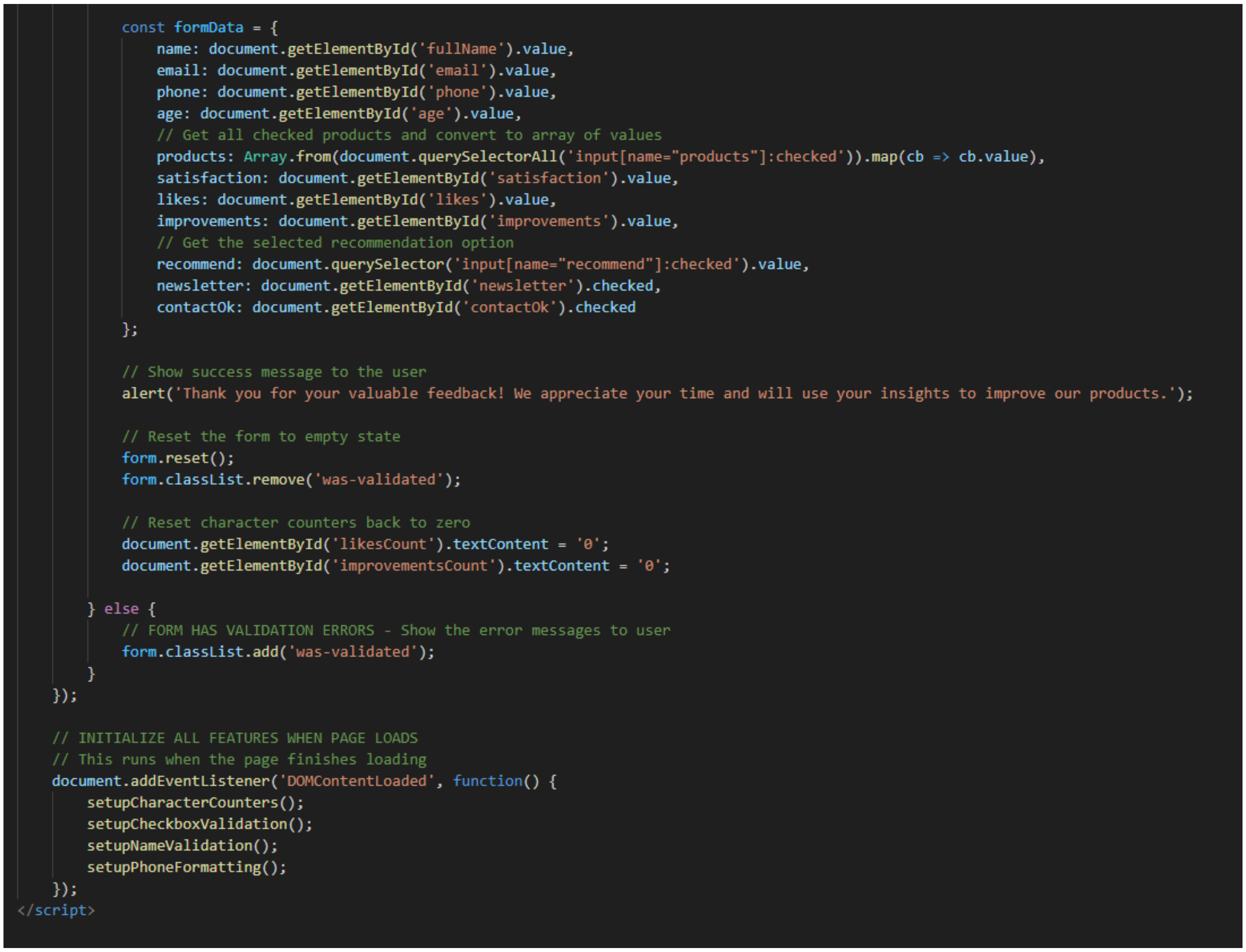
How likely are you to recommend Nasagh to others? *



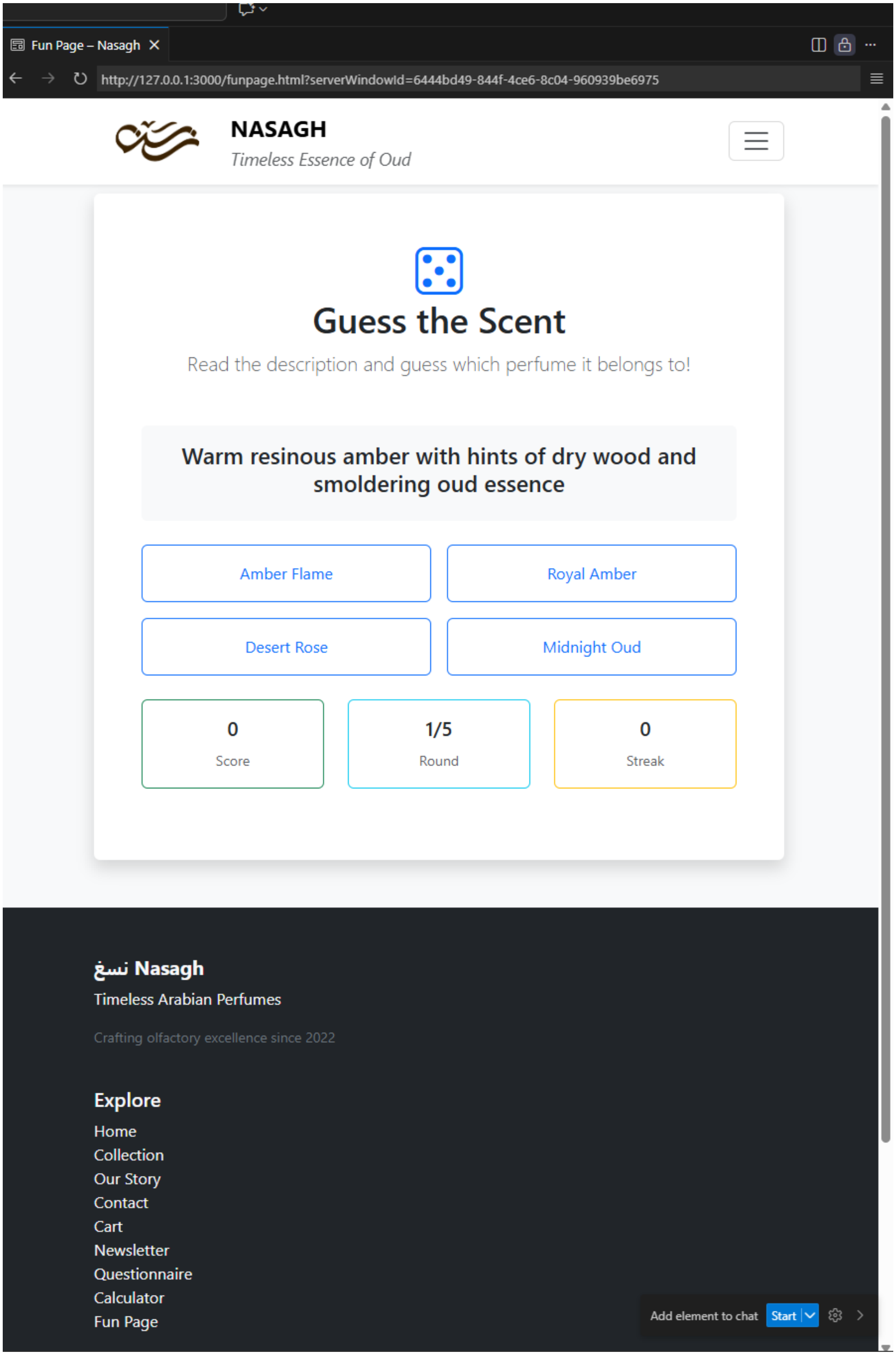
```
    <!-- JavaScript for Form Validation -->
<script>
    // CHARACTER COUNTERS FOR TEXTAREAS
    // This function sets up real-time character counting for the feedback textareas
    function setupCharacterCounters() {
        // Get references to the textareas and their character count displays
        const likesTextarea = document.getElementById('likes');
        const improvementsTextarea = document.getElementById('improvements');
        const likesCount = document.getElementById('likesCount');
        const improvementsCount = document.getElementById('improvementsCount');

        // Monitor the "likes" textarea for any input changes
        likesTextarea.addEventListener('input', function() {
            // Update the character count display
            likesCount.textContent = this.value.length;

            // Validate minimum length requirement in real-time
            // Show red border if less than 10 characters, remove it if enough
            if (this.value.length < 10) {
                this.classList.add('is-invalid');
            } else {
                this.classList.remove('is-invalid');
            }
        });

        // Monitor the "improvements" textarea for input changes (optional field)
        improvementsTextarea.addEventListener('input', function() {
            // Just update the character count, no validation since this field is optional
            improvementsCount.textContent = this.value.length;
        });
    }

    // CUSTOM VALIDATION FOR CHECKBOXES
    // Ensures at least one product checkbox is selected
    function setupCheckboxValidation() {
        // Get all the product checkboxes
        const productCheckboxes = document.querySelectorAll('input[name="products"]');

        // Add change listener to each checkbox
        productCheckboxes.forEach(checkbox => {
            checkbox.addEventListener('change', function() {
                // Check if at least one checkbox is checked
                const atLeastOneChecked = Array.from(productCheckboxes).some(cb => cb.checked);

                // Set custom validation message for all checkboxes
                productCheckboxes.forEach(cb => {
                    if (!atLeastOneChecked) {
                        // No products selected - show error message
                        cb.setCustomValidity('Please select at least one product');
                    } else {
                        // At least one product selected - clear error
                        cb.setCustomValidity('');
                    }
                });
            });
        });
    }

    // REAL-TIME NAME VALIDATION
    // Validates the name field as user types
    function setupNameValidation() {
        const nameInput = document.getElementById('fullName');
```



```
    nameInput.addEventListener('input', function() {
        // Regular expression to check: only letters and spaces, 2-50 characters
        const nameRegex = /^[A-Za-z\s]{2,50}$/;

        // Validate against the pattern
        if (this.value && !nameRegex.test(this.value)) {
            // Invalid name format - set custom error message
            this.setCustomValidity('Name should contain only letters and spaces (2-50 characters)');
        } else {
            // Valid name - clear any error messages
            this.setCustomValidity('');
        }
    });
}

// PHONE NUMBER FORMATTING
// Automatically formats phone numbers as +968 XXX XXXX
function setupPhoneFormatting() {
    const phoneInput = document.getElementById('phone');

    phoneInput.addEventListener('input', function() {
        // Remove all non-digit characters from the input
        let numbers = this.value.replace(/\D/g, '');

        // Format the phone number based on how many digits have been entered
        if (numbers.length <= 3) {
            // Just country code + partial number
            this.value = '+968 ' + numbers;
        } else if (numbers.length <= 8) {
            // Country code + first 4 digits + remaining digits
            this.value = '+968 ' + numbers.substring(0, 4) + ' ' + numbers.substring(4);
        } else {
            // Full number with proper spacing
            this.value = '+968 ' + numbers.substring(0, 4) + ' ' + numbers.substring(4, 8);
        }
    });
}

// FORM SUBMISSION HANDLER
// Processes the form when user clicks submit
document.getElementById('feedbackForm').addEventListener('submit', function(event) {
    // Prevent the form from submitting normally (we want to validate first)
    event.preventDefault();
    event.stopPropagation();

    const form = this;

    // Check if all form validation passes
    if (form.checkValidity()) {
        // FORM IS VALID - Collect all the data from the form

        const formData = {
            name: document.getElementById('fullName').value,
            email: document.getElementById('email').value,
            phone: document.getElementById('phone').value,
            age: document.getElementById('age').value,
            // Get all checked products and convert to array of values
            products: Array.from(document.querySelectorAll('input[name="products"]:checked')).map(cb => cb.value),
            satisfaction: document.getElementById('satisfaction').value,
            likes: document.getElementById('likes').value,
            improvements: document.getElementById('improvements').value,
            // Get the selected recommendation option
            recommend: document.querySelector('input[name="recommend"]:checked').value,
```



```
            const formData = {
                name: document.getElementById('fullName').value,
                email: document.getElementById('email').value,
                phone: document.getElementById('phone').value,
                age: document.getElementById('age').value,
                // Get all checked products and convert to array of values
                products: Array.from(document.querySelectorAll('input[name="products"]:checked')).map(cb => cb.value),
                satisfaction: document.getElementById('satisfaction').value,
                likes: document.getElementById('likes').value,
                improvements: document.getElementById('improvements').value,
                // Get the selected recommendation option
                recommend: document.querySelector('input[name="recommend"]:checked').value,
                newsletter: document.getElementById('newsletter').checked,
                contactOk: document.getElementById('contactOk').checked
            };

            // Show success message to the user
            alert('Thank you for your valuable feedback! We appreciate your time and will use your insights to improve our products.');

            // Reset the form to empty state
            form.reset();
            form.classList.remove('was-validated');

            // Reset character counters back to zero
            document.getElementById('likesCount').textContent = '0';
            document.getElementById('improvementsCount').textContent = '0';

        } else {
            // FORM HAS VALIDATION ERRORS - Show the error messages to user
            form.classList.add('was-validated');
        }
    });

    // INITIALIZE ALL FEATURES WHEN PAGE LOADS
    // This runs when the page finishes loading
    document.addEventListener('DOMContentLoaded', function() {
        setupCharacterCounters();
        setupCheckboxValidation();
        setupNameValidation();
        setupPhoneFormatting();
    });
</script>
```



Fun Page – Nasagh
http://127.0.0.1:3000/funpage.html?serverWindowId=6444bd49-844f-4ce6-8c04-960939be6975
NASAGH
Timeless Essence of Oud

Guess the Scent

Read the description and guess which perfume it belongs to!

Warm resinous amber with hints of dry wood and smoldering oud essence

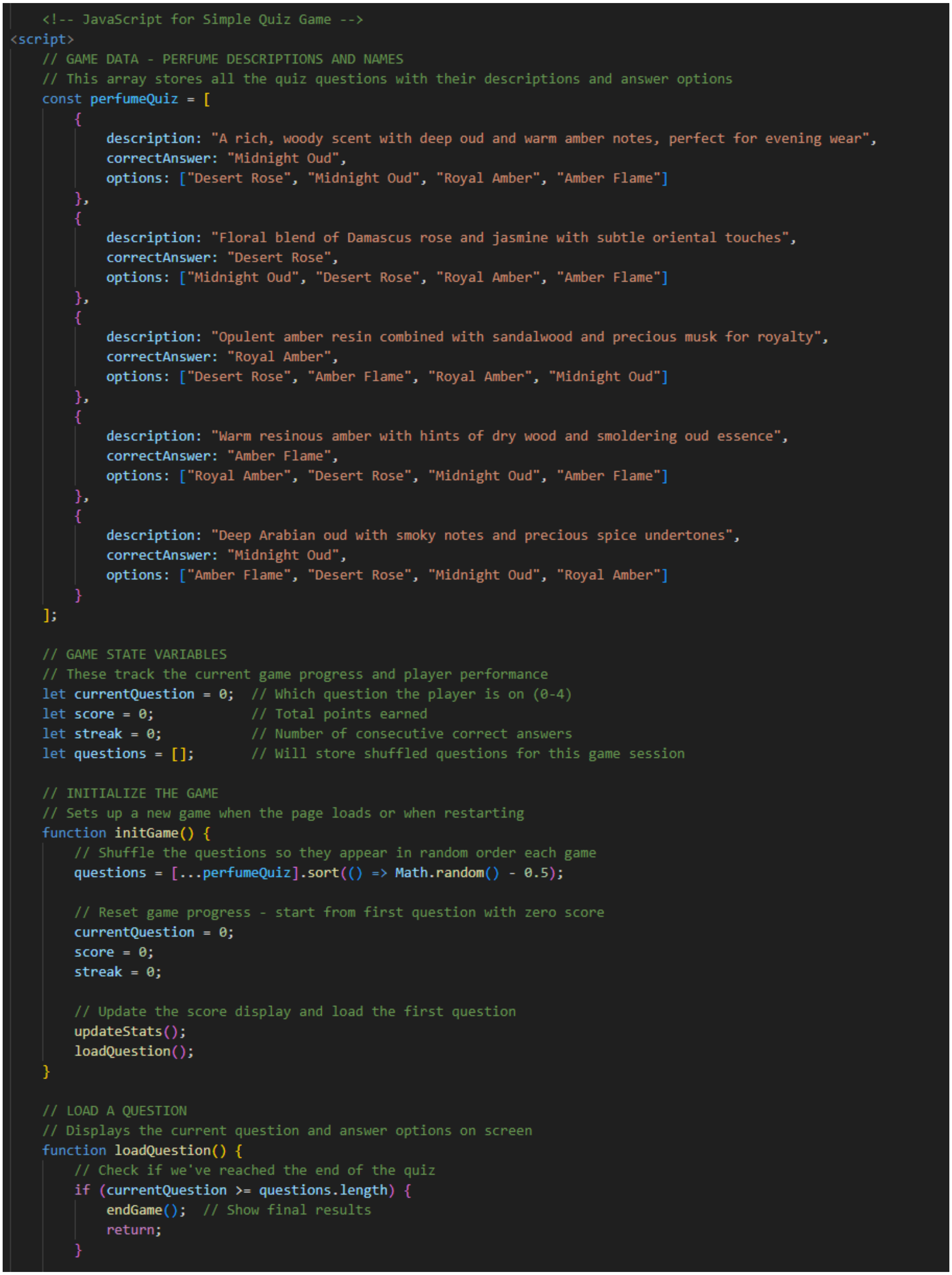
0 Score	1/5 Round	0 Streak

نسغ Nasagh

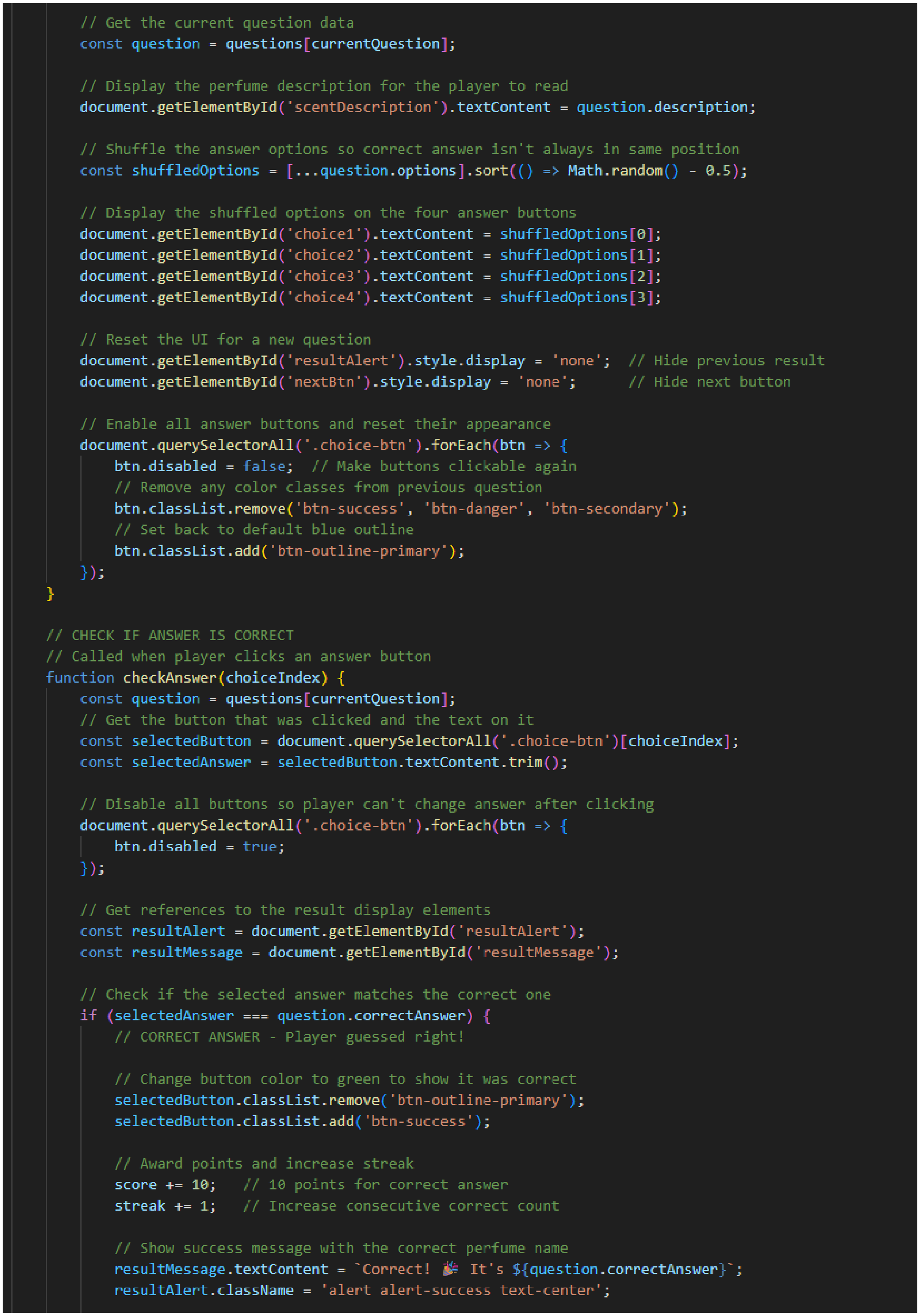
Timeless Arabian Perfumes

Crafting olfactory excellence since 2022

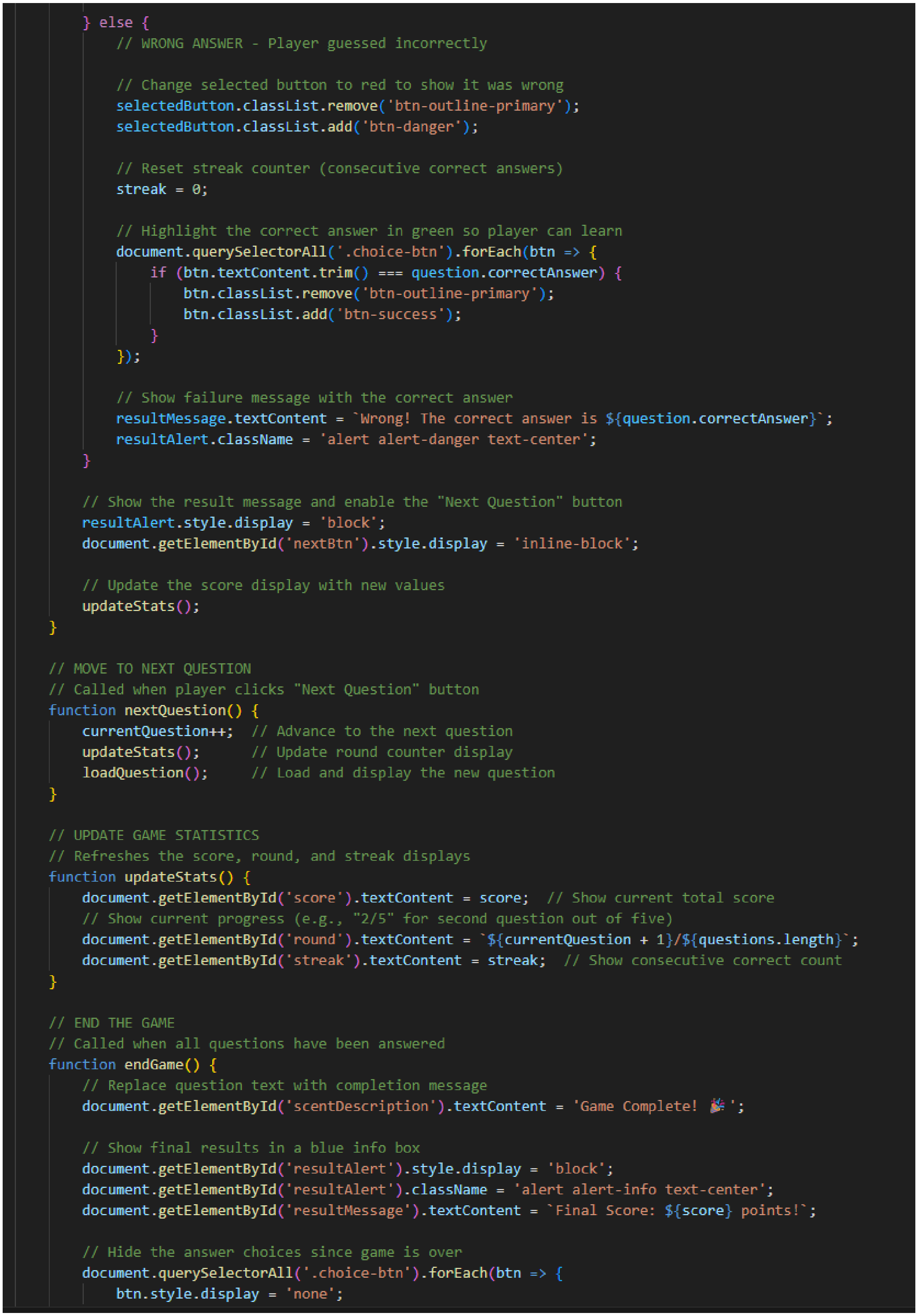
Explore



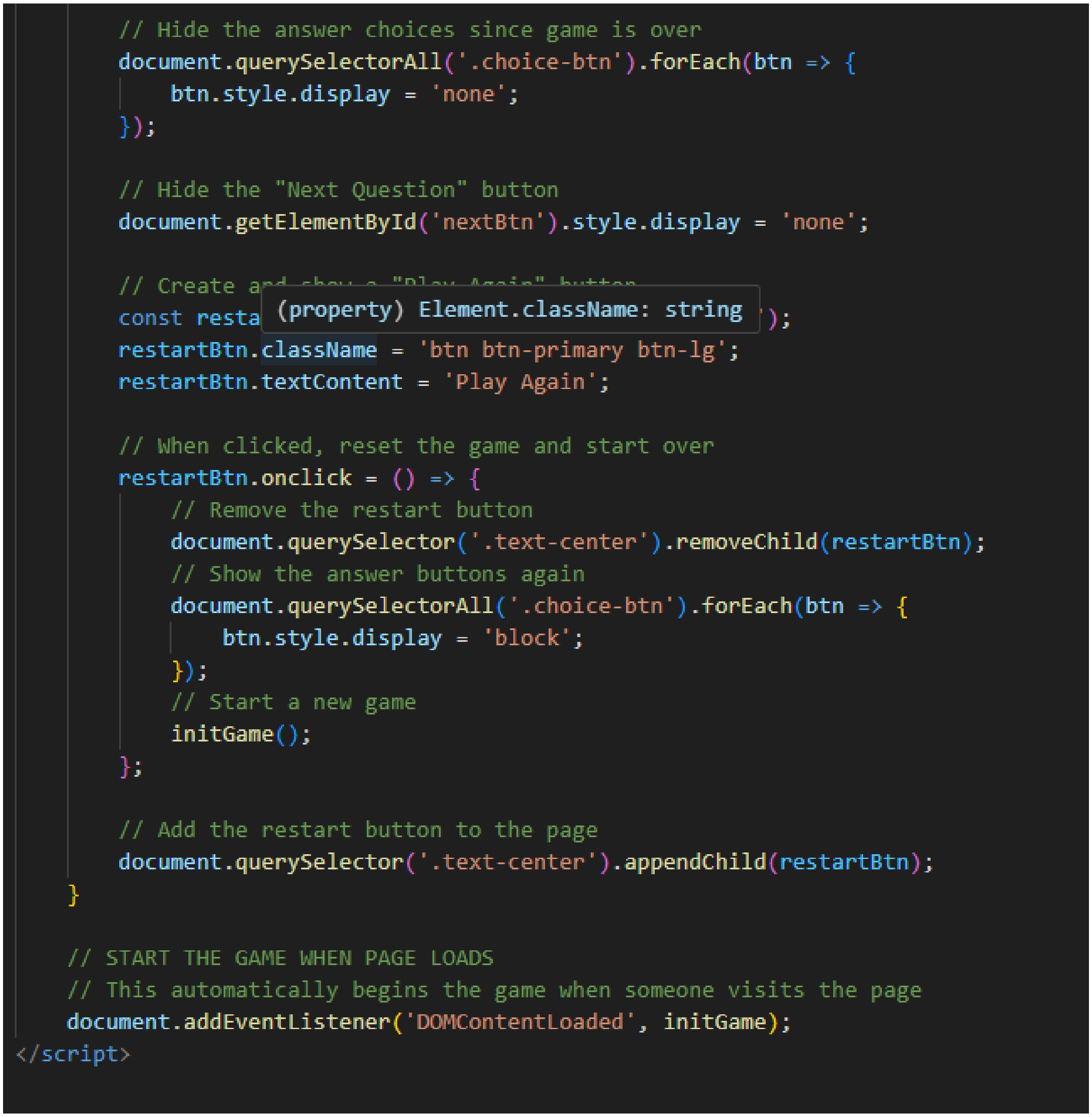
```
    <!-- JavaScript for Simple Quiz Game -->
<script>
    // GAME DATA - PERFUME DESCRIPTIONS AND NAMES
    // This array stores all the quiz questions with their descriptions and answer options
    const perfumeQuiz = [
        {
            description: "A rich, woody scent with deep oud and warm amber notes, perfect for evening wear",
            correctAnswer: "Midnight Oud",
            options: ["Desert Rose", "Midnight Oud", "Royal Amber", "Amber Flame"]
        },
        {
            description: "Floral blend of Damascus rose and jasmine with subtle oriental touches",
            correctAnswer: "Desert Rose",
            options: ["Midnight Oud", "Desert Rose", "Royal Amber", "Amber Flame"]
        },
        {
            description: "Opulent amber resin combined with sandalwood and precious musk for royalty",
            correctAnswer: "Royal Amber",
            options: ["Desert Rose", "Amber Flame", "Royal Amber", "Midnight Oud"]
        },
        {
            description: "Warm resinous amber with hints of dry wood and smoldering oud essence",
            correctAnswer: "Amber Flame",
            options: ["Royal Amber", "Desert Rose", "Midnight Oud", "Amber Flame"]
        },
        {
            description: "Deep Arabian oud with smoky notes and precious spice undertones",
            correctAnswer: "Midnight Oud",
            options: ["Amber Flame", "Desert Rose", "Midnight Oud", "Royal Amber"]
        }
    ];

    // GAME STATE VARIABLES
    // These track the current game progress and player performance
    let currentQuestion = 0;  // Which question the player is on (0-4)
    let score = 0;            // Total points earned
    let streak = 0;           // Number of consecutive correct answers
    let questions = [];       // Will store shuffled questions for this game session

    // INITIALIZE THE GAME
    // Sets up a new game when the page loads or when restarting
    function initGame() {
        // Shuffle the questions so they appear in random order each game
        questions = [...perfumeQuiz].sort(() => Math.random() - 0.5);

        // Reset game progress - start from first question with zero score
        currentQuestion = 0;
        score = 0;
        streak = 0;

        // Update the score display and load the first question
        updateStats();
        loadQuestion();
    }

    // LOAD A QUESTION
    // Displays the current question and answer options on screen
    function loadQuestion() {
        // Check if we've reached the end of the quiz
        if (currentQuestion >= questions.length) {
            endGame();  // Show final results
            return;
        }
```



```
    // Get the current question data
    const question = questions[currentQuestion];

    // Display the perfume description for the player to read
    document.getElementById('scentDescription').textContent = question.description;

    // Shuffle the answer options so correct answer isn't always in same position
    const shuffledOptions = [...question.options].sort(() => Math.random() - 0.5);

    // Display the shuffled options on the four answer buttons
    document.getElementById('choice1').textContent = shuffledOptions[0];
    document.getElementById('choice2').textContent = shuffledOptions[1];
    document.getElementById('choice3').textContent = shuffledOptions[2];
    document.getElementById('choice4').textContent = shuffledOptions[3];

    // Reset the UI for a new question
    document.getElementById('resultAlert').style.display = 'none';  // Hide previous result
    document.getElementById('nextBtn').style.display = 'none';      // Hide next button

    // Enable all answer buttons and reset their appearance
    document.querySelectorAll('.choice-btn').forEach(btn => {
        btn.disabled = false;  // Make buttons clickable again
        // Remove any color classes from previous question
        btn.classList.remove('btn-success', 'btn-danger', 'btn-secondary');
        // Set back to default blue outline
        btn.classList.add('btn-outline-primary');
    });
}

// CHECK IF ANSWER IS CORRECT
// Called when player clicks an answer button
function checkAnswer(choiceIndex) {
    const question = questions[currentQuestion];
    // Get the button that was clicked and the text on it
    const selectedButton = document.querySelectorAll('.choice-btn')[choiceIndex];
    const selectedAnswer = selectedButton.textContent.trim();

    // Disable all buttons so player can't change answer after clicking
    document.querySelectorAll('.choice-btn').forEach(btn => {
        btn.disabled = true;
    });

    // Get references to the result display elements
    const resultAlert = document.getElementById('resultAlert');
    const resultMessage = document.getElementById('resultMessage');

    // Check if the selected answer matches the correct one
    if (selectedAnswer === question.correctAnswer) {
        // CORRECT ANSWER - Player guessed right!

        // Change button color to green to show it was correct
        selectedButton.classList.remove('btn-outline-primary');
        selectedButton.classList.add('btn-success');

        // Award points and increase streak
        score += 10;   // 10 points for correct answer
        streak += 1;   // Increase consecutive correct count

        // Show success message with the correct perfume name
        resultMessage.textContent = `Correct! 🎉 It's ${question.correctAnswer}`;
        resultAlert.className = 'alert alert-success text-center';
```



```
        } else {
            // WRONG ANSWER - Player guessed incorrectly

            // Change selected button to red to show it was wrong
            selectedButton.classList.remove('btn-outline-primary');
            selectedButton.classList.add('btn-danger');

            // Reset streak counter (consecutive correct answers)
            streak = 0;

            // Highlight the correct answer in green so player can learn
            document.querySelectorAll('.choice-btn').forEach(btn => {
                if (btn.textContent.trim() === question.correctAnswer) {
                    btn.classList.remove('btn-outline-primary');
                    btn.classList.add('btn-success');
                }
            });

            // Show failure message with the correct answer
            resultMessage.textContent = `Wrong! The correct answer is ${question.correctAnswer}`;
            resultAlert.className = 'alert alert-danger text-center';
        }

        // Show the result message and enable the "Next Question" button
        resultAlert.style.display = 'block';
        document.getElementById('nextBtn').style.display = 'inline-block';

        // Update the score display with new values
        updateStats();
    }

    // MOVE TO NEXT QUESTION
    // Called when player clicks "Next Question" button
    function nextQuestion() {
        currentQuestion++;  // Advance to the next question
        updateStats();      // Update round counter display
        loadQuestion();     // Load and display the new question
    }

    // UPDATE GAME STATISTICS
    // Refreshes the score, round, and streak displays
    function updateStats() {
        document.getElementById('score').textContent = score;  // Show current total score
        // Show current progress (e.g., "2/5" for second question out of five)
        document.getElementById('round').textContent = `${currentQuestion + 1}/${questions.length}`;
        document.getElementById('streak').textContent = streak;  // Show consecutive correct count
    }

    // END THE GAME
    // Called when all questions have been answered
    function endGame() {
        // Replace question text with completion message
        document.getElementById('scentDescription').textContent = 'Game Complete! 🎉 ';

        // Show final results in a blue info box
        document.getElementById('resultAlert').style.display = 'block';
        document.getElementById('resultAlert').className = 'alert alert-info text-center';
        document.getElementById('resultMessage').textContent = `Final Score: ${score} points!`;

        // Hide the answer choices since game is over
        document.querySelectorAll('.choice-btn').forEach(btn => {
            btn.style.display = 'none';
```



```
        // Hide the answer choices since game is over
        document.querySelectorAll('.choice-btn').forEach(btn => {
            btn.style.display = 'none';
        });

        // Hide the "Next Question" button
        document.getElementById('nextBtn').style.display = 'none';

        // Create and show a "Play Again" button
        const resta (property) Element.className: string ');
        restartBtn.className = 'btn btn-primary btn-lg';
        restartBtn.textContent = 'Play Again';

        // When clicked, reset the game and start over
        restartBtn.onclick = () => {
            // Remove the restart button
            document.querySelector('.text-center').removeChild(restartBtn);
            // Show the answer buttons again
            document.querySelectorAll('.choice-btn').forEach(btn => {
                btn.style.display = 'block';
            });
            // Start a new game
            initGame();
        };

        // Add the restart button to the page
        document.querySelector('.text-center').appendChild(restartBtn);
    }

    // START THE GAME WHEN PAGE LOADS
    // This automatically begins the game when someone visits the page
    document.addEventListener('DOMContentLoaded', initGame);
</script>
```

Task Allocation Table

Team Member	Tasks Completed	Percentage
Elyas Alquwatie	Bootstrap Integration, JavaScript Objects & Arrays, Dynamic Tables, Search Functionality	35%
Hisham Alhinai	Questionnaire Page, Form Validation, Homepage Enhancements, Calculator Page	35%
Almuhannad Alhinai	Fun Page, Navigation, Responsive Design, Documentation	30%

https://elyasaq.github.io/comp3700-project/