

Recenzje produktu

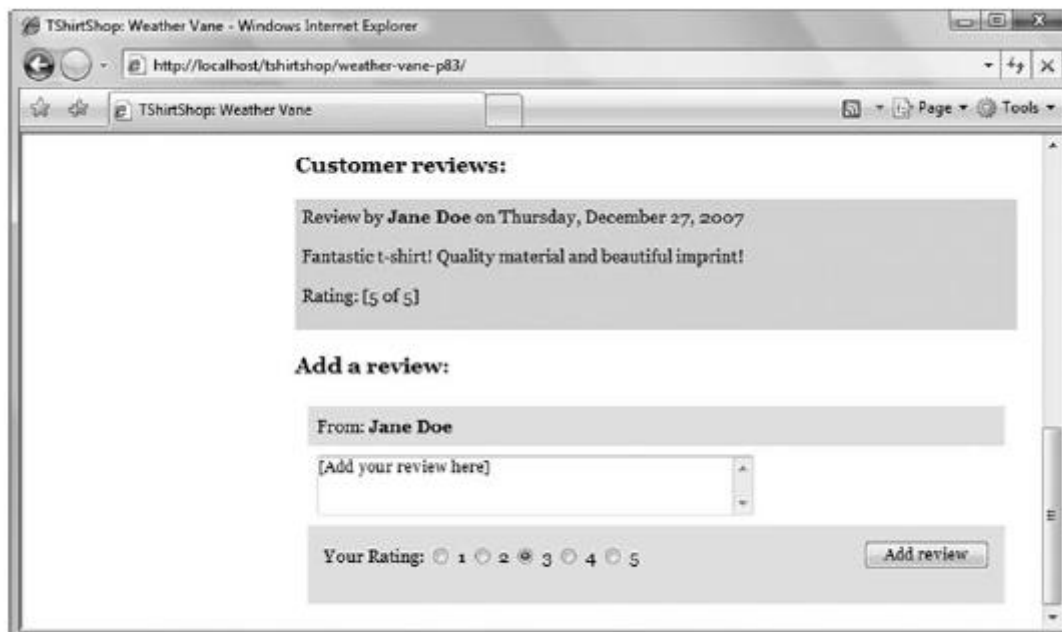
W tym momencie mamy kompletną i funkcjonalną witrynę e-commerce. Jednak to nie powstrzymuje nas przed dodaniem jeszcze większej liczby funkcji, aby uczynić go bardziej użytecznym i przyjemnym dla odwiedzających. Dodając system recenzji produktów do swojej witryny internetowej, możesz zwiększyć szanse, że odwiedzający powrócą do Twojej witryny, aby napisać recenzję zakupionego produktu lub zobaczyć, co inni ludzie myślą o tym produkcie. System recenzji może również pomóc w poznaniu gustów klientów, co pozwala ulepszać rekomendacje produktów, a nawet wprowadzać zmiany na stronie internetowej lub w strukturze katalogu produktów w oparciu o opinie klientów.

Aby ułatwić pracę zarówno klientowi, jak i nam, dodamy listę recenzji produktów i formularz dodawania nowej recenzji produktu do stron szczegółów produktu. Formularz dodawania nowego produktu pojawi się tylko dla zarejestrowanych użytkowników, ponieważ postanowiliśmy nie zezwalać na anonimowe recenzje (jednak możesz to łatwo zmienić, jeśli chcesz). Stworzymy kod dla tej nowej funkcji w zwykły sposób, zaczynając od bazy danych, a kończąc na interfejsie użytkownika (UI).

Planowanie funkcji recenzji produktów

Funkcja recenzji produktów jest dość prosta. Rządzą nim trzy proste decyzje projektowe:

- Lista recenzji i elementy interfejsu niezbędne do dodawania nowych recenzji powinny być wyświetlane pod listą rekomendacji produktów na stronach szczegółów produktu, jak pokazano na Rysunku.



- Jeśli produkt nie ma recenzji, powinniśmy zaprosić czytelnika do napisania pierwszej recenzji, jak pokazano na rysunku.



- Tylko zarejestrowani użytkownicy mogą pisać recenzje produktów.

Przechowujemy recenzje produktów w tabeli bazy danych o nazwie recenzja, która jest manipulowana przez dwie procedury składowane - `catalog_get_product_reviews` i `catalog_create_product_review` - których nazwy same się opisują. Inny kod, który napiszemy, będzie zawierał tę funkcjonalność tylko w formie dostępnej dla odwiedzających.

Wdrażanie recenzji produktów

Wykonaj kroki tego ćwiczenia, aby utworzyć tabelę bazy danych, procedury składowane, warstwę biznesową i funkcje warstwy prezentacji wymagane dla funkcji recenzji produktów.

Ćwiczenie: Wdrażanie recenzji produktów

1. Załaduj phpMyAdmin, wybierz bazę danych `tshirtshop` i otwórz nową stronę zapytania SQL. Następnie wykonaj ten kod, który doda tabelę recenzji do bazy danych `tshirtshop`:

```
-- Create review table
```

```
CREATE TABLE `review` (
  `review_id` INT NOT NULL AUTO_INCREMENT,
  `customer_id` INT NOT NULL,
```

```

`product_id` INT NOT NULL,
`review` TEXT NOT NULL,
`rating` SMALLINT NOT NULL,
`created_on` DATETIME NOT NULL,
PRIMARY KEY (`review_id`),
KEY `idx_review_customer_id` (`customer_id`),
KEY `idx_review_product_id` (`product_id`)
);

```

2. Wykonaj poniższy kod, który utworzy procedurę składowaną `catalog_get_product_reviews` w bazie danych `tshirtshop` (nie zapomnij ustawić ogranicznika na `$$`). Procedura przechowywana `catalog_get_product_review` pobiera recenzje dla produktu zidentyfikowanego przez parametr `inProductId`. Potrzebujemy również nazwiska recenzenta, więc wykonaliśmy INNER JOIN z tabelą `customer`.

```

-- Create catalog_get_product_reviews stored procedure
CREATE PROCEDURE catalog_get_product_reviews(IN inProductId INT)
BEGIN
SELECT c.name, r.review, r.rating, r.created_on
FROM review r
INNER JOIN customer c
ON c.customer_id = r.customer_id
WHERE r.product_id = inProductId
ORDER BY r.created_on DESC;
END$$

```

3. Wykonaj poniższy kod, który dodaje procedurę składowaną `catalog_create_product_review` w bazie danych `tshirtshop` (nie zapomnij ustawić ogranicznika na `$$`). Gdy zarejestrowany użytkownik dodaje recenzję produktu, wywoływana jest procedura przechowywana `catalog_create_product_review`.

```

-- Create catalog_create_product_review stored procedure
CREATE PROCEDURE catalog_create_product_review(IN inCustomerId INT,
IN inProductId INT, IN inReview TEXT, IN inRating SMALLINT)
BEGIN
INSERT INTO review (customer_id, product_id, review, rating, created_on)
VALUES (inCustomerId, inProductId, inReview, inRating, NOW());
END$$

```

4. Dodaj odpowiednie metody warstwy biznesowej do klasy Catalog z pliku business/catalog.php:

```
// Gets the reviews for a specific product
public static function GetProductReviews($productId)
{
    // Build the SQL query
    $sql = 'CALL catalog_get_product_reviews(:product_id)';
    // Build the parameters array
    $params = array (':product_id' => $productId);
    // Execute the query and return the results
    return DatabaseHandler::GetAll($sql, $params);
}

// Creates a product review
public static function CreateProductReview($customer_id, $productId,
    $review, $rating)
{
    // Build the SQL query
    $sql = 'CALL catalog_create_product_review(:customer_id, :product_id,
    :review, :rating)';
    // Build the parameters array
    $params = array (':customer_id' => $customer_id,
    ':product_id' => $productId,
    ':review' => $review,
    ':rating' => $rating);
    // Execute the query
    DatabaseHandler::Execute($sql, $params);
}
```

5. Interfejs użytkownika składa się ze składowego szablonu recenzji, który zostanie umieszczony na stronie szczegółów produktu. Zaczynij od stworzenia prezentacji/templates/reviews.tpl i dodania do niej następującego kodu:

```
{* reviews.tpl *}

{load_presentation_object filename="reviews" assign="obj"}

{if $obj->mTotalReviews != 0}
```

```

<h2>Customer reviews:</h2>
<ul class="reviews-list">
{section name=i loop=$obj->mReviews}
<li>
<p>
Review by <strong>{$obj->mReviews[i].name}</strong> on
{$obj->mReviews[i].created_on | date_format:"%A, %B %e, %Y"}
</p>
<p>{$obj->mReviews[i].review}</p>
<p>Rating: [{$obj->mReviews[i].rating} of 5]</p>
</li>
{/section}
</ul>
{else}
<h2>Be the first person to voice your opinion!</h2>
{/if}
{if $obj->mEnableAddProductReviewForm}
{* add review form *}
<h2>Add a review:</h2>
<form method="post" action="{ $obj->mLinkToProduct}">
<table class="review-table">
<tr>
<td class="add-review">From: <strong>{$obj->mReviewerName}</strong></td>
</tr>
<tr>
<td>
<textarea name="review"
rows="3" cols="65">[Add your review here]</textarea>
</td>
</tr>
<tr>

```

```

<td class="add-review">
<table class="review-table">
<tr>
<td>
Your Rating:
<input type="radio" name="rating" value="1" /> 1
<input type="radio" name="rating" value="2" /> 2
<input type="radio" name="rating" value="3" checked="checked" /> 3
<input type="radio" name="rating" value="4" /> 4
<input type="radio" name="rating" value="5" /> 5
</td>
<td align="right">
<input type="submit" name="AddProductReview" value="Add review" />
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
{else}
<p>You must log in to add a review.</p>
{/if}

```

6. Utwórz plik Presentation/reviews.php i dodaj w nim:

```

<?php
// Class that handles product reviews
class Reviews
{
public $mProductId;
public $mReviews;
public $mTotalReviews;

```

```

public $mReviewerName;
public $mEnableAddProductReviewForm = false;
public $mLinkToProduct;
public function __construct()
{
    if (isset($_GET['ProductId']))
        $this->mProductId = (int)$_GET['ProductId'];
    else
        trigger_error('ProductId not set', E_USER_ERROR);
    $this->mLinkToProduct = Link::ToProduct($this->mProductId);
}
public function init()
{
    // If visitor is logged in ...
    if (Customer::IsAuthenticated())
    {
        // Check if visitor is adding a review
        if (isset($_POST['AddProductReview']))
            Catalog::CreateProductReview(Customer::GetCurrentCustomerId(),
            $this->mProductId, $_POST['review'],
            $_POST['rating']);
        // Display "add review" form because visitor is registered
        $this->mEnableAddProductReviewForm = true;
        // Get visitor's (reviewer's) name
        $customer_data = Customer::Get();
        $this->mReviewerName = $customer_data['name'];
    }
    // Get reviews for this product
    $this->mReviews = Catalog::GetProductReviews($this->mProductId);
    // Get the number of the reviews
    $this->mTotalReviews = count($this->mReviews);
}

```

```
}  
}  
?>
```

7. Otwórz Presentation/templates/product.tpl i dodaj na końcu następujący wiersz:

```
{include file="reviews.tpl"}
```

8. Dodaj następujące style na końcu tshirtshop.css:

```
.reviews-list li {  
background: #ccddff;  
border-bottom: #fff solid 3px;  
display: block;  
padding: 5px;  
}  
.review-table {  
width: 100%;  
}  
.review-table tr td {  
border: none;  
}  
.add-review {  
background: #e6e6e6;  
padding: 5px;  
}
```

9. Załaduj tshirtshop w przeglądarce i kliknij produkt, aby wyświetlić jego stronę szczegółów produktu. Jeśli nie jesteśmy zalogowani, a produkt nie ma recenzji, zobaczymy dane wyjściowe pokazane na drugim rysunku. Przykładowe dane wyjściowe produktu, który ma jedną recenzję, przedstawiono na pierwszym rysunku.

Jak to działa: szablon oparty na składnikach recenzji

Komponentowy szablon recenzji zajmuje się zarówno wyświetlaniem recenzji, jak i dodawaniem nowej recenzji. Pierwsza część pliku Reviews.tpl określa, czy mamy jakieś recenzje do wyświetlenia dla bieżącego produktu. Jeśli tego nie zrobimy, pojawi się krótka wiadomość zachęcająca odwiedzającego do napisania pierwszej recenzji.

```
{if $obj->mTotalReviews != 0}  
<h2>Customer reviews:</h2>
```



```
[a list with reviews]
```

```
{else}
```

```
<h2>Be the first person to voice your opinion!</h2>
```

```
{/if}
```

Druga część szablonu wyświetla formularz do dodania recenzji lub wiadomość, która zaprasza odwiedzającego do zalogowania się, aby móc dodać recenzję:

```
{if $obj->mEnableAddProductReviewForm}
```

```
{* add review form *}
```

```
<h2>Add a review:<h2>
```

```
[add review form]
```

```
{else}
```

```
<p>You must log in to add a review.</p>
```

```
{/if}
```

Kod z obiektu prezentacji jest dość prosty i nie powinien stanowić dla Ciebie problemu.

Podsumowanie

Tak, to było takie proste. Chociaż możesz chcieć dodać ulepszenia do własnego rozwiązania (na przykład zezwolić odwiedzającym na edycję swoich recenzji lub zabronić im dodawania większej liczby recenzji), baza jest dostępna i działa zgodnie z oczekiwaniami. Jesteś teraz gotowy do przejścia do ostatniego rozdziału tej książki, w którym dowiemy się, jak sprzedawać klientom produkty z zewnętrznego źródła (wybraliśmy Amazon.com) za pomocą usług XML Web Services.