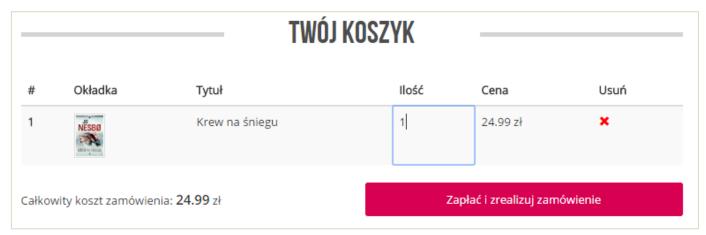
## Dynamiczne zmienianie wartości pola z ilością zamawianych produktów i obliczanie ceny edytowanego zamówienia.



Rysunek 1. Zrzut ekranu aplikacji przedstawiający koszyk sklepu internetowego.

Wykorzystane technologie: język PHP, JavaScript i biblioteka jQuery, format JSON.

```
$ (document) .on('blur', "td[contenteditable=true]", function()
{
    var message_status = $("#status_change_q");
    var field_userid = $(this) .attr("id");
    var value = $(this) .text();
```

Listing 1. Pierwszy fragment skryptu jQuery.

Kliknięcie na element z atrybutem **contenteditable** daje użytkownikowi możliwość jego edycji. Po wprowadzeniu nowej ilości produktów w zamówieniu następuje zdarzenie **blur**, które wykonuje się podczas kliknięcia poza wybrane pole lub wciśnięcie klawisza Tab.

Do zmiennej *message\_status* przypisany jest element HTML o identyfikatorze *status\_change\_q*, który wyświetli powiadomienie o prawidłowej zmianie ilości produktów. Zmienna *field\_userid* pobiera z edytowanej pozycji w koszyku jej identyfikator określony w postaci np. quantity:1, gdzie pierwsza wartość to nazwa tabeli w bazie danych którą będziemy aktualizować, a druga to identyfikator pojedynczego zamówienia. Do zmiennej *value* zostanie zapisana nowa wartość wprowadzona przez użytkownika.

Listing 2. Drugi fragment skryptu jQuery.

Następnie wykonuje się asynchroniczne żądanie do serwera metodą POST, korzystając z funkcji **ajax**, która zaimplementowana jest w bibliotece jQuery. Zmiany z bazie danych zostaną obsłużone w pliku *edit\_quantity.php*. Wysyłane dane mają postać field\_userid + "=" + value, które posiadają wcześniej pobrane wartości. Jeżeli skrypt języka PHP zostanie wykonany, aplikacja wyświetli w odpowiednim polu informacje o pozytywnym lub negatywnym wyniku operacji i po chwili zniknie.

```
if (!empty($_POST)) {
   include "db.php";
   foreach ($_POST as $field_name => $val) {
        $val = (int) $val;
        if (is_int($val)) {
            $field_purchase = strip_tags(trim($field_name));
            $val = strip_tags(trim(mysql_real_escape_string($val)));

        $split_data = explode(':', $field_purchase);
        $purchase_id = $split_data[1];
        $field_name = $split_data[0];
```

Listing 3. Fragment skryptu PHP z pliku edit\_quantity.php.

Na początku skryptu sprawdzamy czy przesłane do serwera dane posiadają nie są puste. Następnie zostanie wykonane połączenie z bazą danych. Sprawdzamy czy przesłana wartość z nową liczbą produktów jest liczbą całkowitą. Jeżeli tak, nazwa wysłanej zmiennej jest rozdzielana i odpowiednio do zmiennej purchase\_id przypisujemy identyfikator zamówienia, a do field\_name – nazwę tabeli którą będziemy aktualizować.

```
if (!empty($purchase_id) && !empty($field_name) && !empty($val)) {
  mysql_query("UPDATE `purchase_item` SET `$field_name`='$val' WHERE
  `purchase_id`='$purchase_id'") or mysql_error();
```

Listing 4. Fragment skryptu PHP z pliku edit\_quantity.php.

Następnie wykonywane jest zapytanie SQL, które aktualizuje rekord z ilością danej książki w danym zamówieniu.

Dalsza część skryptu napisanego w języku JavaScript, wykonuje drugie asynchroniczne żądanie do serwera, które zostaje zrealizowane jeżeli pierwsze zwróci wartość **success**.

Listing 5. Trzeci fragment skryptu jQuery.

Drugie żądanie **Ajax** wykonuje skrypt zawarty w pliku *current\_price.php*, którego zadaniem jest obliczenie aktualnej ceny całego zamówienia. Otrzymane dane z serwera przedstawione zostaną w formacie JSON. Po pomyślnym wykonaniu żądania, w miejsce elementu o identyfikatorze *price\_shopcard* dynamicznie zostanie wstawiona nowa cena zamówienia z koszyka określonego klienta.

```
if ($ SESSION['auth'] == true && $ SESSION['login'] == true) {
    price = 0;
    $user id = $ SESSION['user id'];
    include "db.php";
    Squery pag data = ("SELECT k.price, p.quantity FROM book k, purchase status s,
purchase b INNER JOIN purchase item p ON b.purchase id = p.purchase id and
b.user id='$user id' where s.status id=b.purchase status id and p.book id=k.book id");
    $result pag data = mysql query($query pag data) or die('Database error' .
mysql error());
    if (mysql num rows($result pag data) > 0) {
       while ($row = mysql fetch array($result pag data)) {
            $price = $price + $row['price'] * $row['quantity'];
            $msg['price'] = $price;
        $msg['elements'] = mysql num rows($result pag data);
        echo json encode($msg);
    } else {
       msg['price'] = 0;
       echo json encode($msg);
```

 ${\it Listing~6.~Fragment~skryptu~PHP~z~pliku~current\_price.php.}$ 

Na początku skryptu sprawdzamy czy istnieją zmienne sesyjne, które określają zalogowanego użytkownika. Następnie zapytanie SQL zwraca wszystkie książki znajdujące się w jego koszyku i oblicza ich cenę, którą przypisuje do zmiennej *price*. Funkcją **json\_encode** konwertujemy tablicę *msg*, zawierającą nowe dane do formatu JSON i zwracamy jako odpowiedź do skryptu JavaScript.