

## Building a Shopping Cart using WYSIWYG Web Builder

Internet shopping is growing at a phenomenal rate, and it's never been easier to get your own store online. We show you how to get started by building a shopping cart using WYSIWYG Web Builder and PHP. This article does not provide a complete solution (yet), but as time goes on and based on feedback we will update this article so that it will provide a solid foundation in which you can continue to build in extra features.

Shopping Cart features:

- Uses WYSIWYG Web Builder to manage the layout
- Uses PHP to manage the cart
- Designed for a small number of products
- Fully customizable storefront
- Customer and Merchant email order confirmations
- Easy extendable
- No database required!
- Now with PayPal checkout!

A demo project is available for download, so you do not have to copy/paste all code if you want to experiment with this shopping cart.

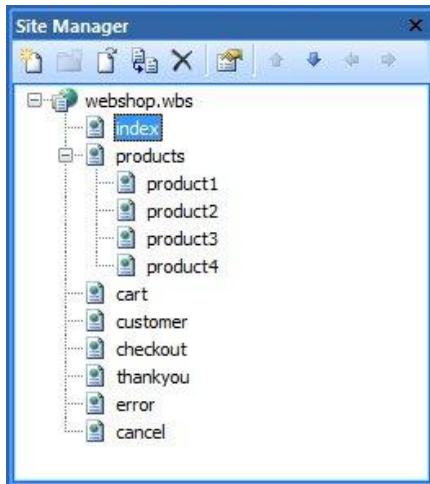
The shopping cart basically consists of the following pages:

<b>products</b>	Displays the available products
<b>cart</b>	Manages the content of the Shopping Cart
<b>customer</b>	Customer information input form (part of the checkout)
<b>checkout</b>	Checkout page, Shipping and Payment details
<b>thankyou</b>	Order completed page

The demo project also has a few additional pages:

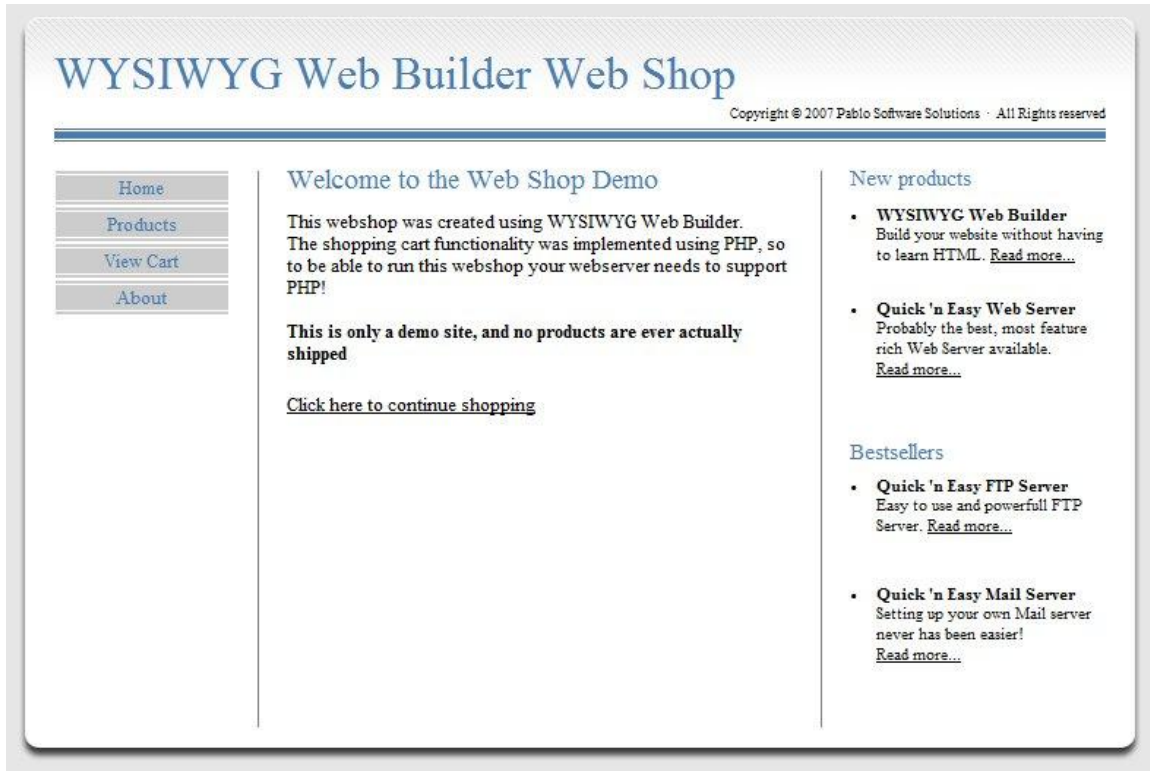
<b>index</b>	Introduction to the online store
<b>product1</b>	Detailed description of a product 1
<b>product2</b>	Detailed description of a product 2
<b>product3</b>	Detailed description of a product 3
<b>product4</b>	Detailed description of a product 4
<b>error</b>	This page display errors which might occur
<b>cancel</b>	Cancel order page, empties the shopping cart

The site structure of this project looks like this (in the Site Manager).



## ***Index Page***

The index page is the home of the shopping cart. It does not contain any special coding. The main purpose is to provide a link to the actual products we want to sell.




## Products Page

The products page gives an overview of all the available products. Each product has an image, a short description, a link to more details and a form. The form will be used to add the product to the cart.

# Products

Copyright © 2007 Pablo Software Solutions · All Rights reserved


- Home
- Products
- View Cart
- About



**Quick 'n Easy FTP Server**  
Easy to use and powerfull FTP Server.  
[Read more...](#)

**Price:** \$ 29.95


**Quantity:**



**Quick 'n Easy Mail Server**  
Setting up your own Mail server never has been easier!  
[Read more...](#)

**Price:** \$ 29.95


**Quantity:**



**Quick 'n Easy Web Server**  
Probably the best, most feature rich Web Server available.  
[Read more...](#)

**Price:** \$ 29.95

**Quantity:**



**WYSIWYG Web Builder**  
Build your website without having to learn HTML.  
[Read more...](#)

**Price:** \$ 34.95

**Quantity:**

## How to add a product to the products page?

### Step 1

Add a form and set the following properties:

Form Name: give the form a nice name;)

Action: cart.php

Method: POST

Encoding type: <empty>

We will use hidden fields to specify some details of the product, these are later used to display the product name and price in the shopping cart. If you like you can add your own items.

<b>productcode</b>	Specifies an unique id for the product.
<b>productname</b>	Specifies the name of the product.
<b>price</b>	Specifies the price of the product.

Form Properties

General Layout Events

General

Form Name: fmProduct1

Action: cart.php

Method: POST

Encoding type:

Hidden Fields

Name	Value
productcode	1001
productname	Quick 'n Easy FTP Server
price	29.95

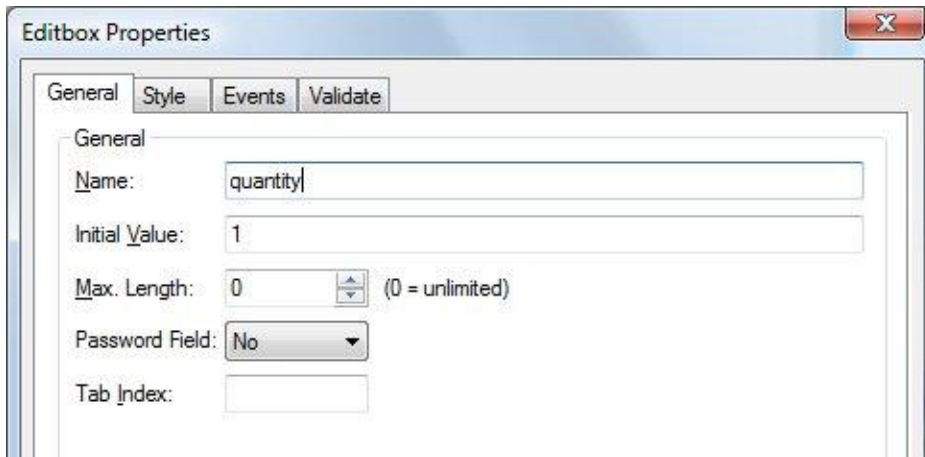
Add... Edit... Delete

OK Cancel

Note that the action is set to **cart.php**. For more details about the cart page see also 'Cart Page' which is explained later in this tutorial.

## Step 2

Add an **EditBox** to the form with the name property set to 'quantity'. This allows the customer to specify the number of items he/she wishes to buy.

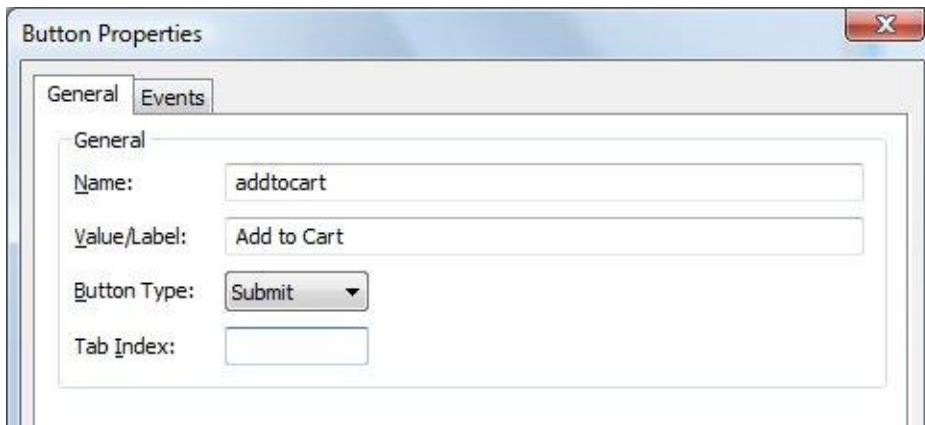


The screenshot shows the 'Editbox Properties' dialog box with the 'General' tab selected. The 'Name' property is set to 'quantity', the 'Initial Value' is '1', and the 'Max. Length' is '0' (unlimited). The 'Password Field' is set to 'No' and the 'Tab Index' is empty.

Property	Value
Name	quantity
Initial Value	1
Max. Length	0 (0 = unlimited)
Password Field	No
Tab Index	

## Step 3

Add a **Button** to the form with the label property set to 'Add to Cart'. If the customer presses this button the product will be added to the cart. Button type: **Submit**.



The screenshot shows the 'Button Properties' dialog box with the 'General' tab selected. The 'Name' property is set to 'addtocart', the 'Value/Label' is 'Add to Cart', and the 'Button Type' is 'Submit'. The 'Tab Index' is empty.

Property	Value
Name	addtocart
Value/Label	Add to Cart
Button Type	Submit
Tab Index	

## Cart Page

The cart page manages the content of the shopping cart. This page takes care of adding and/or removing items. It also allows customers to change the quantity of the items in the cart.

Product code	Product name	Quantity	Price	Total
1001	Quick 'n Easy FTP Server	1	\$29.95	\$29.95
			Total	\$29.95

### Important note:

Make sure you set the file extension in Page Properties to **php**!

## Implementing the cart

### Step 1

- Click Menu->Page->Page HTML
- Click Start of Page
- Copy/paste the following PHP code

```
<?php
session_start();

define("PRODUCTCODE", 0);
define("PRODUCTNAME", 1);
define("QUANTITY", 2);
define("PRICE", 3);

if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
```

```

if (isset($_POST['productcode']))
{
    AddToCart();
}
else
{
    $action = isset($_POST['action']) ? $_POST['action'] : '';
    $value = strtoupper(substr($action, 0, 5));
    switch ($value)
    {
        // continue shopping
        case "CONTI":
            header("Location: "."products.html");
            break;

        // recalculate
        case "RECAL":
            RecalculateCart();
            break;

        // proceed to checkout
        case "CHECK":
            header("Location: "."customer.php");
            break;
    }
}
}

function AddToCart()
{
    $cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : '';
    $itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;

    $cart[PRODUCTCODE][$itemcount] = $_POST['productcode'];
    $cart[PRODUCTNAME][$itemcount] = $_POST['productname'];
    $cart[QUANTITY][$itemcount] = intval($_POST['quantity']);
    $cart[PRICE][$itemcount] = $_POST['price'];
    $itemcount = $itemcount + 1;

    $_SESSION['cart'] = $cart;
    $_SESSION['itemcount'] = $itemcount;
}

function RecalculateCart()
{
    $cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : '';
    $itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;

    for ($i=0; $i<$itemcount; $i++)
    {
        $quantity = $_POST['quantity'][$i];
        if (empty($quantity))
        {
            $quantity = 0;
        }
        else
        if (($quantity < 0) || (!is_numeric($quantity)))
        {
            $quantity = 0;
        }
        $cart[QUANTITY][$i] = intval($quantity);
    }
}

```



```

    }

    for ($j=0; $j<$itemcount; $j++)
    {
        $quantity = $cart[QUANTITY][$j];

        // remove item from the cart
        if ($quantity == 0)
        {
            $itemcount--;

            $curitem = $j;

            while(($curitem+1) < count($cart[0]))
            {
                for ($k=0; $k<4; $k++)
                {
                    $cart[$k][$curitem] = $cart[$k][$curitem+1];
                    $cart[$k][$curitem+1] = '';
                }
                $curitem++;
            }
        }
    }
    $_SESSION['itemcount'] = $itemcount;
    $_SESSION['cart'] = $cart;
}

?>

```

The PHP code is execute each time the page is called.  
Before the contents of the cart is displayed, the code checks if one of the following tasks should be performed:

- Add new item to the cart.
- Continue shopping; redirect customer to products page.
- Recalculate cart.
- Checkout; redirect the customer to the customer page.

### Step 2

Add a form and set the following properties:

Form Name: give the form a nice name;)

Action: cart.php

Method: POST

Encoding type: <empty>

### Step 3

Add a **Button** to the form with the label property set to **Continue Shopping**.

Set the name property to **action**. Button type: **Submit**.

If the customer presses this button he will be redirected to the products page.

### Step 4

Add a **Button** to the form with the label property set to **Recalculate**.

Set the name property to **action**. Button type: **Submit**.

If the customer presses this button the total price of the items in the cart will be recalculated.

## Step 5

Add a **Button** to the form with the label property set to **Check out**.

Set the name property to **action**. Button type: **Submit**.

If the customer presses this button he will be redirected to the checkout page.

## Step 6

Add a **HTML object** to the form. This will be used to dynamically display the items in the cart. Copy/Paste following PHP code:

```
<?php

$cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : '';
$itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;

$strHTML = "";

if ($itemcount == 0)
{
    $strHTML = "<h3>Your shopping cart is empty.</h3>";
}
else
{
    $strHTML = "<div style=\"overflow:auto; height=358px;\">\".\"\\n\";
    $strHTML .= "<table border=\"0\" cellpadding=\"3\" cellspacing=\"2\"
width=\"100%\">\".\"\\n\";
    $strHTML .= "<tr>\".\"\\n\";
    $strHTML .= "<td>Product code</td>\".\"\\n\";
    $strHTML .= "<td>Product name</td>\".\"\\n\";
    $strHTML .= "<td>Quantity</td>\".\"\\n\";
    $strHTML .= "<td>Price</td>\".\"\\n\";
    $strHTML .= "<td>Total</td></tr>\".\"\\n\";

    $total = 0;
    for ($i=0; $i<$itemcount; $i++)
    {
        $strHTML .= "<tr>\".\"\\n\";
        $strHTML .= "<td>\".\"$cart[PRODUCTCODE][$i].\"</td>\".\"\\n\";
        $strHTML .= "<td>\".\"$cart[PRODUCTNAME][$i].\"</td>\".\"\\n\";
        $strHTML .= "<td><input type=\"text\" name=\"quantity\".($i).\"\"
value=\"\".\"$cart[QUANTITY][$i].\"\" size=\"3\"></td>\".\"\\n\";
        $strHTML .= "<td>\".\"$\".number_format($cart[PRICE][$i],2).\"</td>\".\"\\n\";
        $strHTML .=
        "<td>\".\"$\".number_format($cart[PRICE][$i]*$cart[QUANTITY][$i],2).\"</td>\".\"\\n\";
        $strHTML .= "</tr>\".\"\\n\";
        $total = $total + ($cart[PRICE][$i]*$cart[QUANTITY][$i]);
    }

    $strHTML .= "<tr>\".\"\\n\";
    $strHTML .= "<td></td><td></td><td></td>\".\"\\n\";
    $strHTML .= "<td>Total</td>\".\"\\n\";
    $strHTML .= "<td>\".\"$\".number_format($total, 2).\"</td>\".\"\\n\";
    $strHTML .= "</tr>\".\"\\n\";
    $strHTML .= "</table>\".\"\\n\";
    $strHTML .= "</div>\".\"\\n\";
}
echo $strHTML;

?>
```

## Customer Page

The customer page is part of the checkout procedure. It collects customer information like the name, address and telephone number. Of course you can modify this form to suit your purposes.

We use session variables to remember the entered data in case the customer decides to visit another page of the online store before completing the checkout.

The screenshot shows a web form titled "Customer Information" with a copyright notice "Copyright © 2007 Pablo Software Solutions · All Rights reserved". On the left is a navigation menu with links: Home, Products, View Cart, and About. The main content area is titled "Please fill in the following information:" and contains the following fields:

- First name:
- Last name:
- E-mail:
- Address:
- City:
- Zip code:
- State:
- Country:
- Telephone:
- Fax:

A "Continue" button is located at the bottom right of the form.

### Important note:

Make sure you set the file extension in Page Properties to **php**!

### Step 1

- Click Menu->Page->Page HTML
- Click Start of Page
- Copy/paste the following PHP code

```

<?php
session_start();

$itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;
if ($itemcount == 0)
{
    header("Location: "."error.php?msg=".rawurlencode("Please add items to your
shopping cart before checking out."));
    exit;
}

if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    if (isset($_POST['email']))
    {
        $_SESSION['firstname'] = $_POST['firstname'];
        $_SESSION['lastname'] = $_POST['lastname'];
        $_SESSION['address'] = $_POST['address'];
        $_SESSION['address2'] = $_POST['address2'];
        $_SESSION['city'] = $_POST['city'];
        $_SESSION['zip'] = $_POST['zip'];
        $_SESSION['state'] = $_POST['state'];
        $_SESSION['country'] = $_POST['country'];
        $_SESSION['phone'] = $_POST['phone'];
        $_SESSION['fax'] = $_POST['fax'];

        if (ValidateEmail($_POST['email']))
        {
            $_SESSION['email'] = $_POST['email'];
            header("Location: "."checkout.php");
        }
        else
        {
            header("Location: "."error.php?msg=".rawurlencode("Invalid email
address! Make sure you enter valid email address."));
        }
    }
}

$firstname = isset($_SESSION['firstname']) ? $_SESSION['firstname'] : '';
$lastname = isset($_SESSION['lastname']) ? $_SESSION['lastname'] : '';
$email = isset($_SESSION['email']) ? $_SESSION['email'] : '';
$address = isset($_SESSION['address']) ? $_SESSION['address'] : '';
$address2 = isset($_SESSION['address2']) ? $_SESSION['address2'] : '';
$city = isset($_SESSION['city']) ? $_SESSION['city'] : '';
$zip = isset($_SESSION['zip']) ? $_SESSION['zip'] : '';
$state = isset($_SESSION['state']) ? $_SESSION['state'] : '';
$country = isset($_SESSION['country']) ? $_SESSION['country'] : '';
$phone = isset($_SESSION['phone']) ? $_SESSION['phone'] : '';
$fax = isset($_SESSION['fax']) ? $_SESSION['fax'] : '';

function ValidateEmail($email)
{
    $pattern = '/^([0-9a-z]([-.\w]*[0-9a-z])*@(([0-9a-z])+([-\w]*[0-9a-
z])*\.)+[a-z]{2,6})$/i';
    return preg_match($pattern,$email);
}

?>

```

This PHP code will copy all entered values in session variables and then redirect the customer to the checkout page.

## **Step 2**

Add a form and set the following properties:

Form Name: give the form a nice name;)

Action: customer.php

Method: POST

Encoding type: <empty>

### **Note:**

This form will call itself if the data has been submitted.

After the data has been saved into session variables it will redirect the customer to the checkout page.

## **Step 3**

Add some form controls for the data you wish to collect.

### **Note:**

The names of the controls must match the session variable names.

For example if you decide to rename or remove **address2**, you must also modified the code in step 1!!!

## Checkout Page

The checkout page is part of the checkout procedure. It collects shipping and payment information. You can modify this form to suit your purposes.

**Complete your order**

Copyright © 2007 Pablo Software Solutions · All Rights reserved

Home  
Products  
View Cart  
About

**Shipping information (if different from customer information)**

Name:

Address:

City:

Zip code:

State:

Country:

**Payment information**

Type:

Card name:

Card no.:

Expiration date:

### Important note:

Make sure you set the file extension in Page Properties to **php!**

### Step 1

- Click Menu->Page->Page HTML
- Click Start of Page
- Copy/paste the following PHP code

```

<?php
session_start();

$itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;
if ($itemcount == 0)
{
    header("Location: "."error.php?msg=".rawurlencode("Please add items to your
shopping cart before checking out."));
    exit;
}

if (!isset($_SESSION['email']))
{
    header("Location: "."error.php?msg=".rawurlencode("We did not find your
information, please fill the needed information again."));
    exit;
}

if ($_SERVER['REQUEST_METHOD'] == 'POST')
{
    $_SESSION['shipname'] = $_POST['shipname'];
    $_SESSION['shipaddress'] = $_POST['shipaddress'];
    $_SESSION['shipzip'] = $_POST['shipzip'];
    $_SESSION['shipcity'] = $_POST['shipcity'];
    $_SESSION['shipstate'] = $_POST['shipstate'];
    $_SESSION['shipcountry'] = $_POST['shipcountry'];
    $_SESSION['paymenttype'] = $_POST['paymenttype'];
    $_SESSION['cardname'] = $_POST['cardname'];
    $_SESSION['cardno'] = $_POST['cardno'];
    $_SESSION['cardmonth'] = $_POST['cardmonth'];
    $_SESSION['cardyear'] = $_POST['cardyear'];

    header("Location: "."thankyou.php");
}

$shipname = isset($_SESSION['shipname']) ? $_SESSION['shipname'] : '';
$shipaddress = isset($_SESSION['shipaddress']) ? $_SESSION['shipaddress'] : '';
$shipzip = isset($_SESSION['shipzip']) ? $_SESSION['shipzip'] : '';
$shipcity = isset($_SESSION['shipcity']) ? $_SESSION['shipcity'] : '';
$shipstate = isset($_SESSION['shipstate']) ? $_SESSION['shipstate'] : '';
$shipcountry = isset($_SESSION['shipcountry']) ? $_SESSION['shipcountry'] : '';
$cardname = isset($_SESSION['cardname']) ? $_SESSION['cardname'] : '';
$cardno = isset($_SESSION['cardno']) ? $_SESSION['cardno'] : '';

?>

```

This PHP code will copy all entered values in session variables and then redirect the customer to the thankyou page.

## Step 2

Add a form and set the following properties:

Form Name: give the form a nice name;)

Action: checkout.php

Method: POST

Encoding type: <empty>

### Note:

This form will call itself if the data has been submitted.

After the data has been saved into session variables it will redirect the customer to the checkout page.

## Step 3

Add some form controls for the data you wish to collect in the page.

### Note:

The names of the controls must match the session variable names.

For example if you decide to rename or remove **shipcountry**, you must also modified the code in step 1!!!

## Security warning

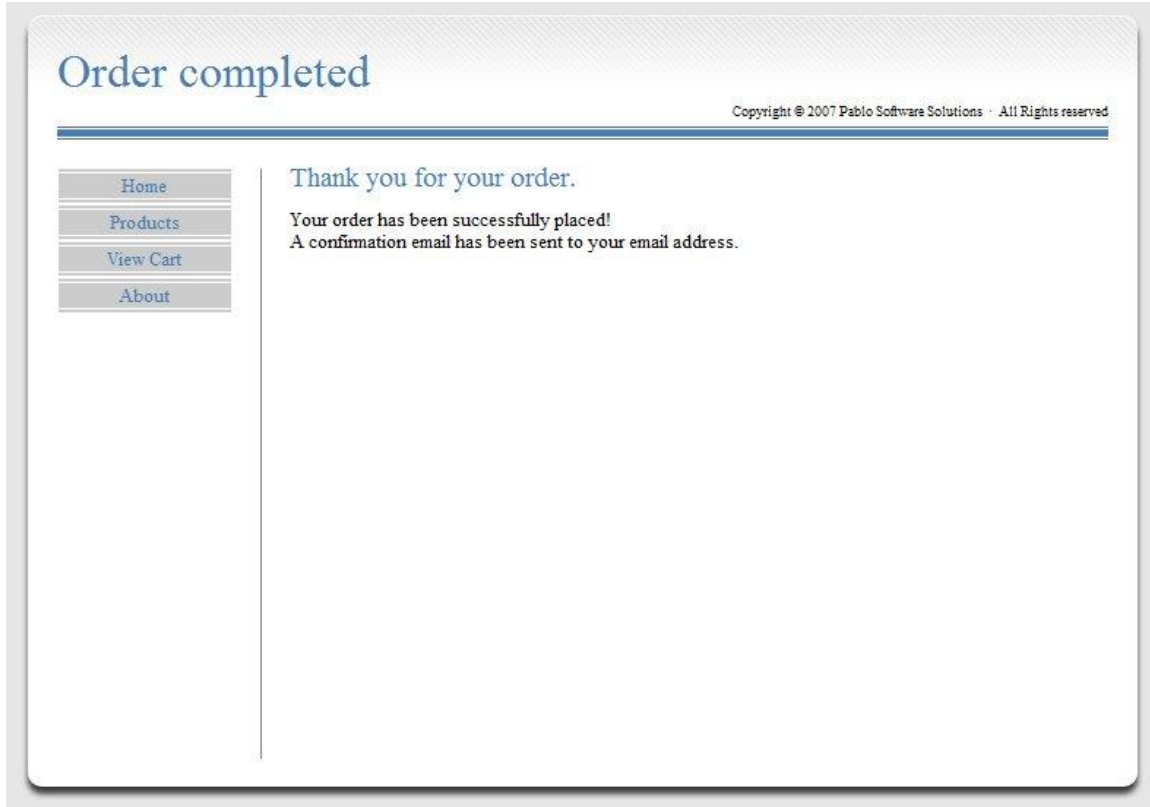
Although we collect the customer's credit card information in this demo, this is not recommended if you do not run this application on a secure server. All entered data is sent unencrypted over the Internet unless you run your shopping cart on a secure web server (https://).



## ThankYou Page

The thankyou page is the final step of the checkout procedure. It generates the confirmation emails and displays a 'Thank You for your order' message.

Almost all code on this page is manually added PHP code, which is needed to generate the confirmation emails.



### Important note:

Make sure you set the file extension in Page Properties to **php**!

### Step 1

- a) Click Menu->Page->Page HTML
- b) Click **Start of Page**
- c) Copy/paste the following PHP code

```
<?php
session_start();

define("PRODUCTCODE", 0);
define("PRODUCTNAME", 1);
define("QUANTITY", 2);
define("PRICE", 3);

$itemcount = isset($_SESSION['itemcount']) ? $_SESSION['itemcount'] : 0;
if ($itemcount == 0)
{
```

```

    header("Location: ".$error.php?msg=".rawurlencode("Please add items to your
shopping cart before checking out.));
    exit;
}

if (!isset($_SESSION['email']))
{
    header("Location: ".$error.php?msg=".rawurlencode("We did not find your
information, please fill the needed information again.));
    exit;
}

// create unique order id
$orderid = strtotime("now").$_SERVER['REMOTE_ADDR'];
$orderid = str_replace(".", "", $orderid);

// get timestamp for this order
$timestamp = date("Y-m-d H:i:s");

// create mail message to customer
$subject = "Your order with shopdemo.wysiwygwebbuilder.com";
$title = "Thank you for your order!";
SendEmail($_SESSION['email'], $subject, $title, $orderid, $timestamp, true);

// create mail message to merchant
$subject = "Order confirmation for shopdemo.wysiwygwebbuilder.com";
$title = "The message is sent for your records of a purchase. Please review
the purchase below, and fulfill as necessary.";
SendEmail("orders@yourdomain.com", $subject, $title, $orderid, $timestamp,
false);

function SendEmail($mailto, $subject, $title, $orderid, $timestamp,
$maskcardno)
{
    $header = "From: orders@yourdomain.com"."\\r\\n";
    $header .= "Reply-To: orders@yourdomain.com"."\\r\\n";
    $header .= "MIME-Version: 1.0"."\\r\\n";
    $header .= "Content-Type: text/plain; charset=utf-8"."\\r\\n";
    $header .= "Content-Transfer-Encoding: 8bit"."\\r\\n";
    $header .= "X-Mailer: PHP v".phpversion();

    $message = $title."\\r\\n"."\\r\\n";
    $message .= "Order ID: PS".$orderid."\\r\\n";
    $message .= "Order Date: ".$timestamp."\\r\\n"."\\r\\n";

    $message .= "Customer info:"."\\r\\n";
    $message .= "Firstname: ".$_SESSION['firstname']."\\r\\n";
    $message .= "Lastname: ".$_SESSION['lastname']."\\r\\n";
    $message .= "Email: ".$_SESSION['email']."\\r\\n";
    $message .= "Address: ".$_SESSION['address']." -
".$_SESSION['address2']."\\r\\n";
    $message .= "City: ".$_SESSION['city']."\\r\\n";
    $message .= "Zip: ".$_SESSION['zip']."\\r\\n";
    $message .= "State: ".$_SESSION['state']."\\r\\n";
    $message .= "Country: ".$_SESSION['country']."\\r\\n";
    $message .= "Phone: ".$_SESSION['phone']."\\r\\n"."\\r\\n";

    $message .= "Shipping address:"."\\r\\n";
    $message .= "Name: ".$_SESSION['shipname']."\\r\\n";
    $message .= "Address: ".$_SESSION['shipaddress']."\\r\\n";
    $message .= "City: ".$_SESSION['shipcity']."\\r\\n";
    $message .= "Zip: ".$_SESSION['shipzip']."\\r\\n";

```

```

$message .= "State: ".$_SESSION['shipstate']."\r\n";
$message .= "Country: ".$_SESSION['shipcountry']."\r\n";

$message .= "Payment info:"."\r\n";
$message .= "Type: ".$_SESSION['paymenttype']."\r\n";
if ($maskcardno)
{
    $message .= "Credit Number:
".substr($_SESSION['cardno'],0,4)."..."."\r\n";
}
else
{
    $message .= "Credit Number: ".$_SESSION['cardno']."\r\n";
}
$message .= "Card Name: ".$_SESSION['cardname']."\r\n";
$message .= "Card Expires:
".$_SESSION['cardmonth']."/".$_SESSION['cardyear']."\r\n"."
\r\n";

$message .= "Ordered items:"."\r\n";
$message .= "Code"."\\t"."Item"."\\t"."No"."\\t"."Price"."
\r\n";
$message .=
"=====
\r\n";

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

$total = 0;
for ($i=0; $i<$itemcount; $i++)
{
    $message .= $cart[PRODUCTCODE][$i]."\\t";
    $message .= $cart[PRODUCTNAME][$i]."\\t";
    $message .= $cart[QUANTITY][$i]."\\t";
    $message .=
"$".number_format($cart[PRICE][$i]*$cart[QUANTITY][$i],2)."
\r\n";
    $total = $total + ($cart[PRICE][$i]*$cart[QUANTITY][$i]);
}

$message .=
"=====
\r\n";
$message .= "Order Grand Total: $".$total."\r\n";
$message .= "\\r\\n"."
\r\n";

mail($mailto, $subject, stripslashes($message), $header);
}
?>

```

### Important note:

Replace all instances of [orders@yourdomain.com](mailto:orders@yourdomain.com) with your own email address!

### Step 2

- Click Menu->Page->Page HTML
- Click **End of Body**
- Copy/paste the following PHP code

```

<?php
// cleanup order details
$_SESSION = array();
session_destroy();
?>

```

## Adding PayPal Support

To add support for PayPal payments all you have to do is forward the shopping cart variables to the PayPal website.

Add a **HTML object** to the page. Copy/Paste following PHP code:

```
<?php
// display PayPal checkout?
if ($_SESSION['paymenttype'] == "PayPal")
{
    echo "<font style=\"font-size:13px\" color=\"#000000\" face=\"Times New Roman\">Please click the PayPal button below to securely pay for your order.<br><br></font>\n";
    echo "<form target=\"PayPal\" action=\"https://www.paypal.com/cgi-bin/webscr\" method=\"post\">\n";
    echo "<input type=\"hidden\" name=\"cmd\" value=\"_cart\">\n";
    echo "<input type=\"hidden\" name=\"business\" value=\"orders@yourdomain.com\">\n";
    echo "<input type=\"hidden\" name=\"receiver_email\" value=\"orders@yourdomain.com\">\n";
    echo "<input type=\"hidden\" name=\"upload\" value=\"1\">\n";

    $itemcount = $_SESSION['itemcount'];
    $scart = $_SESSION['cart'];

    for ($i=0; $i<$itemcount; $i++)
    {
        $item = $i+1;
        echo "<input type=\"hidden\" name=\"item_name_\" . $item . \"\" value=\"\" . $scart[PRODUCTNAME][$i] . \"\">\n";
        echo "<input type=\"hidden\" name=\"item_number_\" . $item . \"\" value=\"\" . $scart[PRODUCTCODE][$i] . \"\">\n";
        echo "<input type=\"hidden\" name=\"amount_\" . $item . \"\" value=\"\" . number_format($scart[PRICE][$i]*$scart[QUANTITY][$i],2) . \"\">\n";
    }
    echo "<input type=\"hidden\" name=\"currency_code\" value=\"USD\">\n";
    echo "<input type=\"hidden\" name=\"no_shipping\" value=\"1\">\n";
    echo "<input type=\"hidden\" name=\"no_note\" value=\"1\">\n";
    echo "<input type=\"image\" name=\"submit\" src=\"http://images.paypal.com/images/x-click-but6.gif\" alt=\"Make payments with PayPal, it's fast, free, and secure!\">\n";
    echo "</form>\n";
}
?>
```

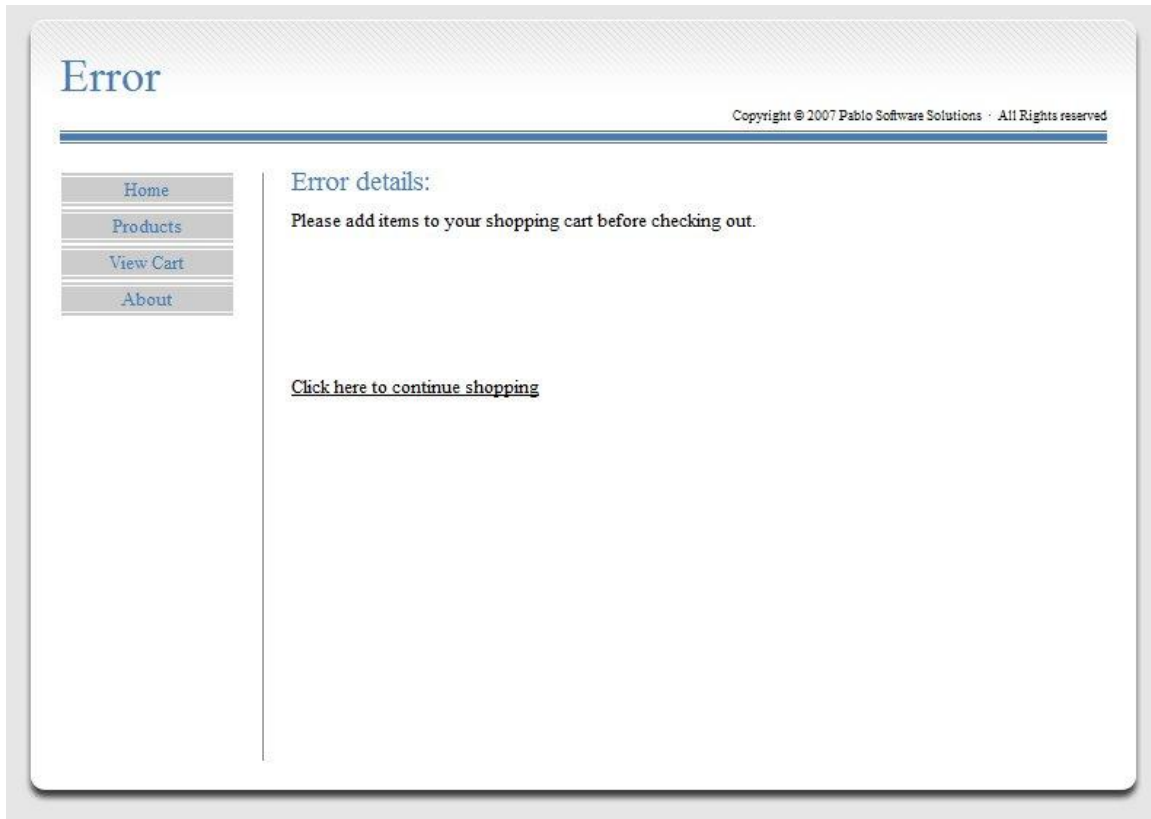
If the paymenttype is set to 'PayPal' this code dynamically generates a form with all the required PayPal variables.

More details about this PayPal integration is available here:

[http://www.paypal.com/cgi-bin/webscr?cmd=\\_pdn\\_howto\\_checkout\\_outside](http://www.paypal.com/cgi-bin/webscr?cmd=_pdn_howto_checkout_outside)

## Error Page

The error page was added to introduce a nice way to display error messages. It basically takes the **msg** value from URL and displays it in a HTML object.



### Important note:

Make sure you set the file extension in Page Properties to **php**!

### Step 1

Add a HTML object to the page and copy/paste this code:

```
<font style="font-size:13px" color="#000000" face="Times New Roman">
<?php
echo $_GET['msg'];
?>
</font>
```

## **Cancel Page**

The cancel page was added to introduce a method to empty the shopping cart.



### **Important note:**

Make sure you set the file extension in Page Properties to **php**!

### **Step 1**

- a) Click Menu->Page->Page HTML
- b) Click Start of Page
- c) Copy/paste the following PHP code:

```
<?php
session_start();
$_SESSION = array();
session_destroy();

?>
```

If you have any comments, suggestions or improvements to this tutorial, please discuss them on the forum:

<http://www.wysiwygwebbuilder.com/forum/>

Note this solution is not a replacement for a professional (database based) ecommerce solution like oscommerce. However you can easily integrate this lightweight solution into your Web Builder website and you will have full control over the look and feel.

We are planning to integrate a shopping cart component in WYSIWYG Web Builder in the near future, but first we'd like to know if we are on the right track. So let us know what you think;)

This tutorial was created by Pablo Software Solutions with special thanks to Kees Versluis.

Copyright 2007 Pablo Software Solutions

<http://www.pablosoftwaresolutions.com>

<http://www.wysiwygwebbuilder.com>