Brotankery Website

I. Abstract

My senior project was to design and implement a website for the online apparel retailer, Brotankery. Brotankery specializes in selling neon tank tops for men and women. Their target demographic is 18-24 year old Americans who want tank tops for parties, raves, working out, or going to the beach. Having never done web development before, I thought this would be a good opportunity to explore web technologies and see how they interact to create a customer-facing interface for the company.

Creating the website from start to finish would take many steps. First I had to learn basic HTML and CSS to create the basic structure of the site, including the following pages: Home, Shop, Blog, FAQ, and Cart. I created an admin section where administrators could sign in to update the inventory, edit products, and analyze recent purchases. In addition I created a basic products page template that would display individual products and allow users to add them to their cart. I created a shopping cart using session variables in PHP and a Cart page to display that cart. Finally I used Google Checkout’s API to direct the user to Google Checkout’s checkout process, receive order notifications, and receive order updates.

Technologies used included: HTML, CSS, JavaScript, PHP, and MySQL. HTML provided the content, CSS provided the presentation, and JavaScript provided some interactive elements for the user. PHP was the server-side scripting language used, and it interacted with a MySQL database to store inventory, process transactions, and record restocks. Additionally I used WordPress for the blog section, and Google Checkout’s API for the checkout process and order confirmation/notification/update process.

II. Final product

A. Structure
   a. Home
   b. Shop
      i. Neon Green Tank
      ii. Neon Yellow Tank
      iii. Neon Salmon Tank
      iv. Neon Blue Tank
   c. Blog (Wordpress)
   d. FAQ
   e. Cart
   f. Store admin
      i. Inventory_list
      ii. Record_restock
      iii. Add_transaction
      iv. Edit_transaction
B. Screenshots
   a. Home page

   ![Home page screenshot](image1)

   b. Shop page

   ![Shop page screenshot](image2)

   c. Shop page (blacklight)

   ![Shop page (blacklight) screenshot](image3)
d. Neon Green Tank (same for other colors)

![Image of Neon Green Tank]

- High-quality, American-made tank
- Cute, colorful design
- Comfortable fit
- Available in various colors

Price: $18.99
Size: XS-L

Add to Cart

Blog

Hello! Welcome to Bro Tankery!

This week, we've been busy working on our new website. We've updated our front page to better serve our customers. The changes are behind the scenes, so you'll only see a few new features.

We've also modified the layout of our shop page, so the tanks are displayed in a new way.

You'll notice a small light bulb when you click on it! The neon tanks feature a fun design under a blacklight.

Check it out →

In addition, the CART page now displays shipping and handling charges up front for greater transparency. It really helps when shopping online!

Lastly, we'd like to inform everyone that all transactions are secured by Google Checkout. We do not store or save any of your sensitive data (like credit card information) on our servers.

FAQ

Who are you?

We are four friends who met in high school and played lead roles through college. Despite each of us going our separate ways, we still remain friends. We've been able to stay in touch and share our interests and aspirations.

How do you make your tanks?

- Where are you located?
- What is your shipping and handling policy?
- Do you have a money-back guarantee?
- Is it safe shopping with you?
- My organization (religion, society, team, club) wants to use your tanks. Do you offer bulk discounts or volume pricing?
- How can I contact you?

Cart

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dry Tank - Neon Green (2)</td>
<td>1</td>
<td>$14.99</td>
</tr>
</tbody>
</table>

Subtotal: $14.99
Shipping: UPS Priority Mail $1.99 (total $16.98)

Shipping: UPS Ground $9.99 (total $24.98)

Total (excluding CA tax): $18.99

Add to Cart
C. Description of how the website works
   a. Website UI
      The website is displayed to the user using HTML, CSS, and JavaScript. The structure is described in part A above, and screenshots are provided in part B. The server runs PHP and interacts with the database to display products in the Shop and Product (green, yellow, etc.) page.
   b. Cart
      When the user adds an item to the cart, the item’s unique id is added to an associative PHP array stored as a session variable. This cart array associates unique product ID’s with quantities. The default session variable timeout is 24 minutes, or when the browser is closed. When the cart page is displayed, items in the cart array are shown along with their corresponding quantities.
   c. Checkout
      The cart is then translated into an HTML form, structured so that Google Checkout can understand it. The html form is passed to Google Checkout, and Google Checkout takes care of the checkout process. User data like credit card information is handled by Google Checkout, not our website.
      Once Google Checkout receives a new order, it notifies our response handler code (“responsehandler.php”) by sending it a new order notification. Our response handler is set up to handle notifications from Google Checkout by recording new transactions in the database, updating existing transactions (like financial status or cancellation status), updating the inventory in the database, and sending ACKs back to Google Checkout.
   d. Database
      A MySQL database stores the following tables: admin, products, inventory, restocks, transactions, and transaction_items. The main purpose of the database is to maintain products, update inventory, and record transactions.
   e. Hosting
      The website is hosted by inMotionHosting and costs approximately $80/year.

III. Lessons learned
   a. I learned how to write code in HTML, CSS, JavaScript, PHP, and MySQL.
   b. I learned how the PHP scripting language could be executed on a server by browsers or by Google Checkout API calls to responseHandler.php.
   c. I learned to watch out for security pitfalls (like SQL injections) and to protect my website against common security flaws.
   d. I practiced writing extensible code that could be adapted for different products and updated easily for additional functionality.
   e. I learned how to use tools like Dreamweaver and PHPMyAdmin.