```
      <div id="side">

      <div id="cart">
        <%= render(:partial => "cart", :object => @cart) %>
      </div>

        <a href="http://www....">Home</a><br />
        <a href="http://www..../faq">Questions</a><br />
        <a href="http://www..../news">News</a><br />
        <a href="http://www..../contact">Contact</a><br />
      </div>
      <div id="main">
      <% if flash[:notice] -%>
        <div id="notice"><%= flash[:notice] %></div>
      <% end -%>
        <%= yield :layout %>
      </div>
    </div>
  </body>
  </html>
```

Now we have to make a small change to the store controller. We're invoking the layout while looking at the store's index action, and that action doesn't currently set @cart. That's easy enough to remedy.

Download depot_j/app/controllers/store_controller.rb

```
def index
  @products = Product.find_products_for_sale
  @cart = find_cart
end
```

If you display the catalog after adding something to your cart, you should see something like Figure 9.1, on the next page.[2] Let's just wait for the Webby Award nomination.

Changing the Flow

Now that we're displaying the cart in the sidebar, we can change the way that the `Add to Cart` button works. Rather than displaying a separate cart page, all it has to do is refresh the main index page. The change is pretty simple: at the end of the add_to_cart action, we simply redirect the browser back to the index.

Download depot_k/app/controllers/store_controller.rb

```
def add_to_cart
  begin
    product = Product.find(params[:id])
  rescue ActiveRecord::RecordNotFound
    logger.error("Attempt to access invalid product #{params[:id]}")
    redirect_to_index("Invalid product")
```

2. And if you've updated your CSS appropriately.... See the listing on page 680 for our CSS.