Design

I wrote the XHTML structure first with appropriately name Div IDs and Classes, leaving spaces for the addition of the PHP code.
The layout, colours, font choices, background images, mouse hover behaviour, and navigation indicators are all controlled using Cascading Style Sheets (CSS). The Cookery Pages have their own additional CSS style sheets to give extra emphasis in this area.

Programming Language Used.

I had to make a choice early on as to which programming language to use that would communicate with the MySQL database.
The two choices were PERL or PHP. In researching PERL I found that it is quick to implement with small amounts of code (PERL Golf) producing the functionality requirements of the bookshop website. However, the way that PHP can be compartmentalised into different functions and integrated within existing static HTML pages meant that the tasks could be separated into obvious divisions. Also I could write code that is W3C compliant, accessible and Search Engine Optimised.
PHP has been used to issue MySQL commands to the database. The most important of those commands we used are: -
INSERT for adding the book data, SELECT for viewing the book data, UPDATE for editing the book data and DELETE for removing a row of book data.
When the user would like to view a book or genre type the GET command is attached to the submission button/sub navigation, which sends the value via the URL to the page that controls the viewing of that book(s). For example the value in the URL is used as the constraining condition for viewing the data i.e. the last part of the SELECT command.
When more data is sent via a HTML form POST is used to send the data to the remaining MySQL commands, which are either INSERT, UPDATE or DELETE.
When a list of data is being displayed in a number of rows I have used a WHILE loop that fetches the contents of one array row at a time until the condition is met.

Shopping Cart

I have used PHP Sessions to hold the array values in a hash table. The unique book IDs are used to index the contents of the hash table. Providing the browser remains open the user is able to view as many books and pages throughout the website are return the to shopping cart to view any that he/she has added.

Form Validation

The validate user input via the HTML forms we have used JavaScript. This is client side to very fast and does not require the user to submit the form to the server. For example a pop up message is displayed if a user leaves an input box blank.
The JavaScript is stored in it’s own files and called in when needed leaving the HTML coded pages uncluttered.
However in the future I would most likely use PHP to validate input boxes form with a form that posts to itself, stores user errors in an array and uses inline warnings that appear right next to the offending input box.

Testing

All the pages have been tested throughout for W3C validation XHTML 1.0 Strict and have passed. We have also tested performed usability testing of all the forms, navigation, and general functionality.