**ISCG7420**

**Web Application Development**

**Assignment 2: PHP**

**Roland Askew**

**AskewR04**

**1461498**

Table of Contents

[1. Business Specification 5](#_Toc463340351)

[1. Original Requirements 5](#_Toc463340352)

[2. Expanded Requirements 6](#_Toc463340353)

[3. Site URL 7](#_Toc463340354)

[2. User Interface Design 8](#_Toc463340355)

[1. Site Map 8](#_Toc463340356)

[2. Wire-Frames 9](#_Toc463340357)

[1. Home Page 9](#_Toc463340358)

[2. Product Detail 10](#_Toc463340359)

[3. Registration 11](#_Toc463340360)

[4. Contact page 11](#_Toc463340361)

[5. Checkout 12](#_Toc463340362)

[6. Post Checkout / Profile Page 13](#_Toc463340363)

[7. Customer Login 14](#_Toc463340364)

[8. Customer Post-Login 15](#_Toc463340365)

[9. Administrator Administration 15](#_Toc463340366)

[10. Category Administration 16](#_Toc463340367)

[11. Supplier Administration 17](#_Toc463340369)

[12. Customer Administration 17](#_Toc463340370)

[13. Caps Administration 18](#_Toc463340371)

[14. Orders Administration 18](#_Toc463340372)

[15. Colours Administration 19](#_Toc463340373)

[3. Database Design 20](#_Toc463340374)

[1. ERD 20](#_Toc463340375)

[4. System Design Rationale 21](#_Toc463340376)

[1. Client Side Techniques 21](#_Toc463340377)

[2. Server Side Techniques 21](#_Toc463340378)

[5. Test Plan with Results 22](#_Toc463340379)

# Summary

|  |  |
| --- | --- |
| Site Url: | http://dochyper.unitec.ac.nz/AskewR04/PHP\_Assignment |
| Database: | AskewR04MySql3 (MySql) |
| Admin Login: | test\_admin |
| Admin Password: | test\_\_password |
| Testing Customer Login: | test\_customer |
| Testing Customer Password: | test\_\_password |
| Disabled Customer Login: | disabled\_customer |
| Disabled Customer Password: | test\_\_password |

# 2. Business Specification & Requirements

## General Purpose

The Web Application is for Quality Caps Ltd, a business selling Caps and similar hats, and is for the purpose of online purchasing, ordering and related e-commerce.

The Web Application is to show caps categories to a visiting customer, allow the customer to choose caps to buy and to place an order.

Customers have to be registered with the website, and have to have logged in, before they may place orders. A registered customer, upon logging in, can also view and change their registered details, change their password, and also view the Orders previously placed.

A visiting (unregistered) customer can view categories and caps, and select caps to buy, but cannot place an order without logging in. However a visitor may register then login and place an order.

## Shopping Process

A visitor or customer can conduct shopping activities from the main (home) page. Initially they are presented with a list of categories with associated caps and a list of all caps for sale. These lists appear in pages, and a customer can use buttons at the bottom of each page to change pages, and see which page they are currently viewing. Selecting a Category will load a list of caps for that category. Selecting a Cap will show further details of that cap, with the option to enter a quantity for purchase. The customer may then add this quantity to the cart or return to viewing the list of caps.

Only Categories with Caps assigned to them are shown in the list of Categories. Only Caps assigned to a Category are shown in the initial list, or in the Cap lists specific to each category.

A Shopping Cart is used to retain caps selected for purchase. This Cart can be used by both visitors and registered customers. Details are given on the total costs of the currently selected caps, and costs broken down into quantities and subtotals per cap. A customer may remove an item from the cart by pressing the red X button next to it. The quantity for a cap within the cart cannot be modified, but a customer can remove that cap and quantity, and add the cap again with the desired quantity. The Cart also appears in pages, with similar buttons at the bottom for changing pages. The Cart may be cleared at any time.

When a logged in customer is ready, they may click the checkout button to review the order before placing it. From there they may Place the order, clear the current order, or remove selected caps and associated quantities. Currently there is no option to modify a quantity for a cap, only to remove it. Once an order is placed, the customer is redirected to the Orders screen where they may review the new order and previous orders.

## Registration and Profiles

A customer may register through the registration page. The registration page requires the full name, an email, a login name, a password, at least one contact number (home, work or cell-phone), and a shipping address comprised of a street address, suburb and a city.

Home and work numbers must be landline numbers (8 to 10 digits, with a leading “0”). Cell numbers must be a local cellphone format (9 to 11 digits, with a leading “0”). The email and login must be unique from other customers. Login Names must be alphanumeric and may contain underscores (“\_”). Other Names, Suburbs and Cities must only have letters and whitespace. Street Addresses must consist of a street number, followed by a street name (letters and whitespace only), and a suffix.

Once registered, the customer should receive an email with their login and password, which they will need to remember. The customer is then redirected to the login screen. They may enter their login name and password, and click the login button to proceed.

If for some reason a customer account has been disabled, then when trying to log in the customer will instead see a warning message.

A logged in customer can edit their profile. They will not be able to change their email or login, which is permanent. They can edit any other details (as long as they meet the above requirements). Also they can change their password, but they cannot see their old password.

## Admin

An administrator may login at the standard login page. From there they are redirected to a special Admin section.

The Administrator may upload and delete files, Add or delete customers (which have no orders placed), disable customers, change an order from waiting to shipped, delete an order, Add or delete a cap (if it is not part of an existing order), “Retire” a Cap (remove it from the list of caps for sale), Add or delete a category (if no cap is assigned to it), Add or delete a supplier (if no cap is assigned to it). Also under each page the Administrator may view existing items for that type of item.

For the files page, a list of files is shown. Clicking on a file name loads the image of that file and selects it for deletion. Clicking delete when a file is selected deletes that file. Note it is possible to delete an image file even if used by a Cap. It is also possible to upload files. Only one file can be uploaded at any one time. Only image (PNG, JPG) files less than 120KB in size may be uploaded. When uploaded, the image file name is changed to a unique hash based on the image content – this means that it is not possible to store duplicate images, but it is possible to upload images with the same original filename.

For other admin pages, there is a list at left showing the existing entities (Caps, Categories, etc), and a form in the middle. The form will show different fields depending on which entity. For example, the Caps page shows a list of caps (by id and name), and a caps form. Clicking on a Cap at left will load that Cap into the form. From there, the Administrator may click Delete to remove that Cap from the system (If the Cap is not referenced by an existing Order). The Admin may also click the button Add… to clear the form, fill out the required fields, then click Save to save the new Cap.

All Admin pages have a delete button to delete entities. An entity must be loaded into the current admin form before it can be deleted. Most Admin pages have an Add… button for adding new entities, except for Orders. The Orders page has a special button, Ship, to change an Order Status from ‘waiting’ to ‘shipped’. The Caps page has a special button, Remove Category, which sets the Cap category to NULL.

## Other

From the edit profile page, a customer may visit the Orders Page. There they will see a list of all orders placed.

There is a general contact page, available to all visitors.

All pages show a footer, which shows the date and time, a copyright, and a message for visitors or logged in users.

Most pages have a Header, which displays a logo and a menu of options. The Visitor header has links to login, registration and contact pages. The Customer header has links to logout, editing the profile and contact pages. The Admin Header has links to admin pages for each entity, and to the files admin page. The Logo acts as a link to the home page, except for Administration pages, where it performs a logout.



# 2. Site Map



# 

# 3. User Interface Design

## 1. Wire-Frames

Note that all Wireframes are approximate prototypes – The final web forms may have minor differences in appearance.

### 1. Home Page



The Site will use Black headers and footers. The overall colour theme will be blue, with a dark blue background and light blue sections. This black and blue colour theme is compatible with the bootstrap default element themes (Blue buttons, black labels, etc). Writing will by default be black, except for themed buttons, labels and controls. The colour theme avoids red-green colour blindness issues and is more accessible (readable) for users of all ages.

The header and footer technique (Logo and top menu as header, Copyright bar as footer) will be a template common to all public pages, as previously specified.

### 

### 2. Product Detail



### 3. Registration



All data fields contain placeholder hints advising what type of data entry (alphanumeric, phone number, etc). If required fields are left empty, or have invalid data, a warning notice appears when trying to Register / submit.

### 4. Contact page



### 

### 

### 5. Checkout



### 

### 6. Post Checkout / Profile Page

The Profile page matches the Registration page, but is filled with the details of the customer. The user must click Edit to be able to make changes to their profile, and may click reset to undo any changes before submitting them. The password field is changed to a “Change Password” Button. Clicking it allows the user to enter a new password. Independently of other changes to the profile. To record the new password, and other changes, the user must click the Save / Submit button. If no new password is given, the old password is kept. The user is always emailed after making a profile change, even if the login and password details are the same.  
  
An additional button is present, for Orders. Clicking it shows a list of the orders placed, and their status.

### 7. Customer Login



If the account is disabled, a notice appears at the bottom of the middle section.

### 8. Customer Post-Login

After logging in, the customer is redirected to the home page.

### 9. Administration Pages

### 

Administration currently supports adding and deleting entities, and a few additional functions for specific entities (see Orders, Image files and Caps).

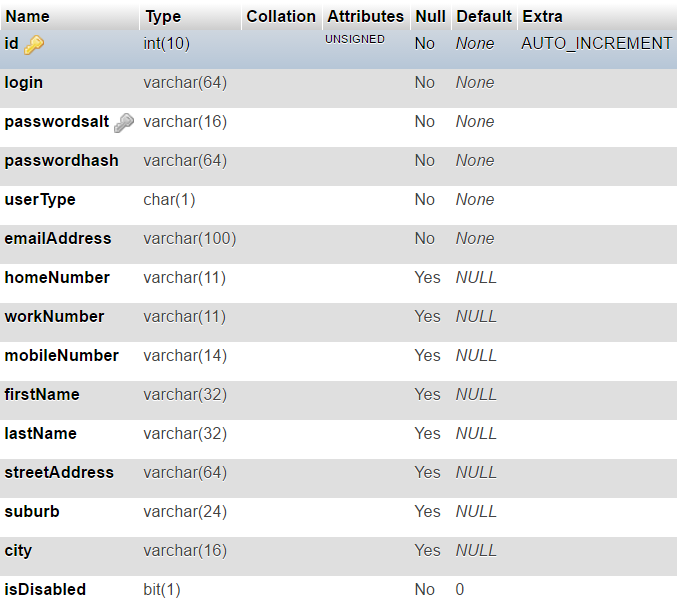
### 2. Database Design

## 1. ERD

­­­

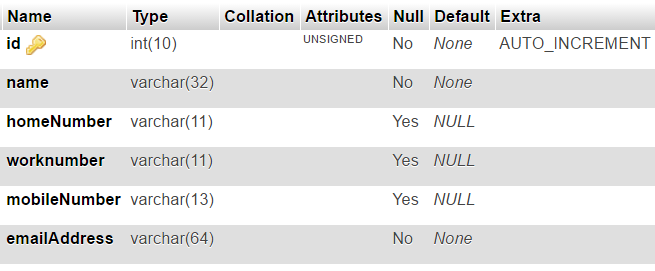
### 2. Data Dictionary

SiteUser



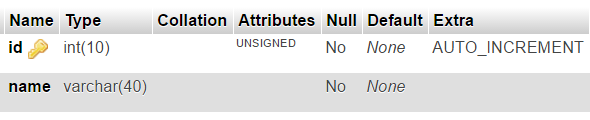
Stores both Administrators and Customers. Admins require only the first 6 fields. Customers should have at least one contact number, and require all other fields. It is the responsibility to the Web Application to make sure this requirement is met. isDisabled defines if a customer may login or not.

Supplier

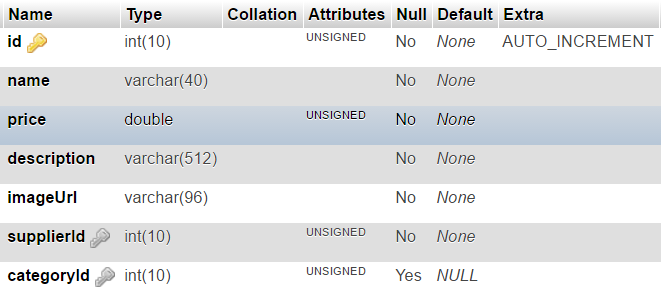


At least one contact number is required. It is the responsibility to the Web Application to make sure this requirement is met.

Category

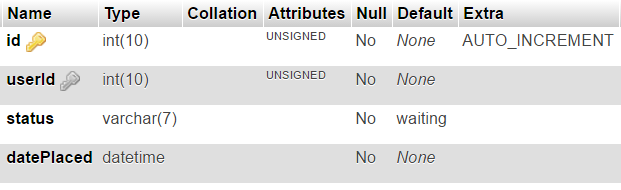


Cap

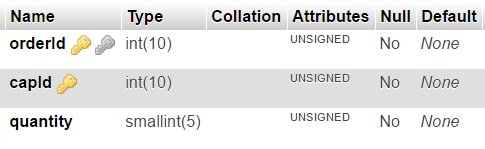


A cap may have the categoryId set to NULL. This indicates the cap is not for sale.

CustomerOrder



OrderItem



Bridging table between a customer’s order, and the caps placed in that order.

# 4. System Design Rationale

## 1. Client Side Techniques

HTML / CSS / JavaScript will be used. HTML5 will be used for input validation and restriction, and JavaScript will be used for some validation.

For Registration and profile updates, HTML5 will be used to screen input data. Placeholder hints will appear in empty fields, and all data entry will be screened using regex patterns (with the ‘pattern’ tag) and input types (email, number, text, password). Forms will be used for managing submission of entered data, and form restrictions on input elements (using the ‘required’, ‘min’ and ‘minlength’ tags) will be used to prevent invalid data entry.

Data validation for administration pages will use javascript to check that fields have not accidentially been left empty, and HTML5 used to restrict data entry to specific types were appropriate.

Bootstrap and some customised CSS will be used for appearance, layout management, and responsiveness to user actions. Javascript will be used for some event management and responding to user actions. The main window area will allow vertical scrolling to allow users to see all content should it be too big for the screen. Bootstrap containers, rows and col classes will be used to manage layout and allow layout changes between screens of different generic sizes.

Bootstrap provides many useful CSS classes and interactive javascript, and provides layout management responsive to different device screen sizes, so using it is simpler than reinventing the wheel and designing custom layout and responsiveness.

HTML5 and Javascript respond quicker than server side scripts to user interaction, and can directly manipulate frontend elements at the client end, whereas server side scripts can only replace content or reload pages. So using HTML5 and Javascript can reduce response time and create a more satisfying user experience.

Javascript is also used in combination with server-side techniques for pagination. Specifically, Javascript is used to manage AJAX pagination requests, and to prepare pagination controls with correct values and responses.

JQuery is frequently used as an extension of JavaScript, as JQuery already contains effective functions for manipulating elements and handling AJAX requests.

## 2. Server Side Techniques

A database is used to store long term persistent data (such as returning customer details, placed orders, available caps and more), and PHP scripting is used to manage server side interaction with the database and populating content with information from the Database.

For reuse of functionality, PHP objects are used, and organised into layers (via namespaces).

The Business Layer contains several objects, to manage specific requests for each entity (Caps, Categories, Customers, and Orders). It also contains an Administration object to manage all requests for administration purposes. This allows different privilege levels to be set within the code base, for different users. An example is allowing a customer to see only products available for sale, while an Administrator can see both available and withdrawn products.

The DataLayer has one object, for managing Database Interaction and queries. This is designed for sanitising input data, and to manage adding, updating and reading data. It contains all DDL / DML code required, and checks for and replaces missing tables each time it is created. There are no generic functions – all requests are confined to specific actions. This prevents any user accidentally requesting an action which could corrupt or damage the database.

AJAX requests are used to speed up some interaction, by limiting requested content to existing sections of a webpage. This is faster than replacing a whole webpage.

The MySQL extended SQL command, LIMIT, is used for pagination purposes, in combination with AJAX and Javascript. When a new page of data is requested, a database scan (of a table) is required, but is limited to the section representing the page. This significantly speeds up the response time of webpages, reduced reliance on cookies, and gets around issues with maintaining storage of entity data in the window / browser (updates to the database take immediate effect, whereas using browser memory storage requires synchronising with the database at regular intervals). AJAX is used to limit content changes to the section used to display the changed page. As mentioned above, Javascript is required to make the AJAX calls.

Some templates are used, but for simplicity this is limited to the headers and footers (using HTML5 and PHP scripting) which are included in webpages as appropriate.

As part of the server side persistence and security, all stored user passwords are hashed with a random salt, which is also stored at the server. All salts are kept unique. This makes it harder to hack passwords, as the same password will produce a different hash, due to having a different salt.

A Session is contained server side, per user, to store persistent data over the duration of the user’s visit. The Session is used to store the shopping cart, identify a logged in user, and identify if a user is an admin. In this way, different user types can be recognised, and orders can be prepared before being placed. The shopping cart persists regardless of if a user is logged in, and this leads to a better shopping experience because a prepared order is not lost on logout, login or while doing other things.

For security, sessions are regenerated when a user logs out, to reduce the risk of impersonation by a hacker.

# 5. Test Plan with Results

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Actions | Expected outcome | Actual Outcome |
| Initial Site Visit. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Home Page, Top Menu, Visitor. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Examine the Top Menu. | Top Menu shows the logo at left, the links Contact Us, Register and Login. | Top Menu shows what is described in expected outcome. |
|  | Click “Contact Us” | Contact Us page is shown. | Contact Us page is shown. |
|  | Click “Register” | Register page is shown. | Register page is shown. |
|  | Click “Login” | Login page is shown. | Login page is shown. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Home page, Categories Section. | Enter the site URL into the browser address bar and press enter. | The home page appears.  The Categories section shows 3 categories. | The home page appears.  The Categories section shows 3 categories. |
|  | Click the picture for the category “Women’s Caps” in the Categories section at left. | The top of the Products section at centre, says Women’s Caps. | The top of the Products section at centre, says Women’s Caps. |
|  | Click the picture for the category “Men’s Caps” in the Categories section at left. | A Grid of 4 caps is shown. Page number is 1. | A Grid of 4 caps is shown. Page number is 1. |
|  | Click the “next” button at the bottom of the Category section. | New categories are shown in the categories section. | New categories are shown in the categories section |
|  | Click the picture for the category “Business Caps” in the Categories section at left. | A Grid of 4 caps is shown. Page number is 1. Some of the caps are different to before. | A Grid of 4 caps is shown. Page number is 1. |
|  | Click the “previous” button at the bottom of the Category section. | The Categories section shows 3 categories. | Matches expected outcome. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Home Page, Products and Shopping Cart. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click the picture for the category “Men’s Caps” in the Categories section at left. | A Grid of 4 caps is shown. Page number is 1. | Matches expected outcome. |
|  | Click the picture for the category “Top Peak” in the Products section at centre. | New Section appears with following details: Top Peak, $22.90. | Matches expected outcome. |
|  | Click Cancel. | The previous Grid of caps is shown again. | Matches expected outcome. |
|  | Click “Next” at the bottom of the Products section. | A new Grid of caps is shown. Page number is now 2. | Matches expected outcome. |
|  | Click Return. | The previous Grid of caps is shown again. | Matches expected outcome. |
|  | Click the product “Side Squat” in the Products section at centre. | Section appears with following details: Side Squat, $29.40. | Matches expected outcome. |
|  | Change the Quantity to 2. | The Quantity textbox now contains 2. | The Quantity textbox now contains 2. |
|  | Click “Add To Cart”. | A new item appears in the shopping cart:  “Side Squat  $29.40 X 2” | Matches expected outcome. |
|  | Click the picture for the category “Children’s Caps” in the Categories section at left. | A Grid of 4 caps is shown. Page number is 1. | Matches expected outcome. |
|  | Click the next button, at the bottom of the products section, twice. | A Grid of 4 caps is shown. Page number is 3. | Matches expected outcome. |
|  | Click the cap “Dour Peak”. | Section appears with following details: Dour Peak, $29.40. | Matches expected outcome. |
|  | Change the Quantity to 3. | The Quantity textbox now contains 3. | The Quantity textbox now contains 3. |
|  | Click “Add To Cart”. | A new item appears in the shopping cart:  “Dour Peak, $29.40  $29.40 X 3” | Matches expected outcome. |
|  | In the Cart, next to “ID: 13”, click the big red X. | Item 13, “Side Squat” is deleted from the cart. | Matches expected outcome. |
|  | Click “Clear” at the bottom of the cart. | The Cart is empty. | Matches expected outcome. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Home Page, Logo. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click the “contact Us” link. | The Contact Us page appears. | The Contact Us page appears. |
|  | Click the Logo. | The home page appears. | The home page appears. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Visitor Restrictions. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click Logout. | The home page appears. | The home page appears. |
|  | Verify Checkout is not enabled. | Checkout button is faded and cannot be clicked. | Matches expected outcome. |
|  | Enter the URL “http://dochyper.unitec.ac.nz/AskewR04/PHP\_Assignment/Pages/checkout.php” in the address bar and press enter. | You are redirected to the home page. | Matches expected outcome. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Contact Us Page. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click Contact Us link in top menu. | The contact Us page appears. | The contact Us page appears. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Registration. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click the link “Register” in the top menu. | The Registration page appears. | The Registration page appears. |
|  | Click Register | A message appears, to fill out one of the fields. | Matches expected outcome |
|  | Type “Simon!” in the First Name field. | The first name field contains “Simon!” | Matches expected outcome |
|  | Click Register | A message appears, to match the requested format. | Matches expected outcome |
|  | Change First Name field to “Simon” | First name field contains “Simon” | Matches expected outcome |
|  | Type “Joe” in the last Name field. | Last name field contains “Joe” | Matches expected outcome |
|  | Type “AskewR04@myunitec.ac.nz” in the email field. | Email field contains “AskewR04@myunitec.ac.nz” | Matches expected outcome |
|  | Type “Customer111” in the login field. | Login field contains “Customer111” | Matches expected outcome |
|  | Type “123456” in the password field. | Password field contains “123456” | Matches expected outcome |
|  | Type “123 Carly Road” in the Street Address field. | Street Address field contains “123 Carly Road” | Matches expected outcome |
|  | Type “Mt Albert” in the Suburb field. | Suburb field contains “Mt Albert” | Matches expected outcome |
|  | Type “Auckland” in the City field. | City field contains “Auckland” | Matches expected outcome |
|  | Click Register | A message appears, to lengthen the field to 10 characters. The Caret moves to the password field. | Matches expected outcome |
|  | Change password to “ThisIsAPassword” | Password field changed to “ThisIsAPassword” | Matches expected outcome |
|  | Change Login to “CustomerTesting” | Login field changed to “CustomerTesting” | Matches expected outcome |
|  | Change Email. Use an email you can access at this time. | Email field has been changed. | Email field has been changed. |
|  | Click Register | A message appears at the bottom, in red, to include at least one phone number. | Matches expected outcome |
|  |  |  |  |
|  | Type “022445555” into Home Contact Number | - | - |
|  | Type “1114445555” into Mobile Contact Number | - | - |
|  | Click Register | A message appears, to match the requested format. The Caret has shifted to the Mobile Phone field | Matches expected outcome |
|  | Change Mobile Number to “0114445555” | - | - |
|  | Remove the Work Number | - | - |
|  | Click Register | Login page appears.  Message is shown: You have been successfully registered and may now log in. | Matches expected outcome |
|  | Verify Email was Received. | Email was received, containing login and password, contact numbers and address. | Matches expected outcome |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Login, Customer. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_customer” into login textbox. Type “test\_\_password” into password textbox. | Login control shows “test\_customer” and “test\_\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Home Page Appears | Matches expected outcome |
|  | Examine the Top Menu. | Register link has changed to “Profile”. | Register link has changed to “Profile”. |
|  | Click Profile. | The Profile page appears.  First and last name are “Testing”, “Customer” respectively. | Matches expected outcome |
|  | Click Orders | A page appears. At least 5 orders appear. 1 has a status of shipped, the rest are waiting. | Matches expected outcome |
|  | Click Logout button. | The home Page shows. The top menu says “Greetings Visitor!” | The home page shows as expected. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Customer, Checkout. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_customer” into login textbox. Type “test\_\_password” into password textbox. | Login control shows “test\_customer” and “test\_\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Home page appears | Home page appears |
|  | Click “Men’s Caps” picture under Categories. | A new Grid of caps is shown. Page number is now 1. | Matches expected outcome |
|  | Click Next, at bottom of Products Section | A new Grid of caps is shown. Page number is now 2. | Matches expected outcome |
|  | Click “Top Trucker” Picture. | Top Trucker cap details are shown | Matches expected outcome |
|  | Change Quantity to 2 | - | Matches expected outcome |
|  | Click Add to Cart | New item appears in shopping cart: Top Trucker $25.30  X 2 | Matches expected outcome |
|  | Click “Women’s Caps” picture under Categories. | A new Grid of caps is shown. Page number is now 1. | Matches expected outcome |
|  | Click Next, at bottom of Products Section, 3 times | A new Grid of caps is shown. Page number is now 4. | Matches expected outcome |
|  | Click “Tall Breaker” Picture. | Tall Breaker details are shown | Matches expected outcome |
|  | Change Quantity to 3 | - | Matches expected outcome |
|  | Click Add to Cart | New item appears in shopping cart: Tall Breaker $11.20  Blue X 3 | Matches expected outcome |
|  | Click Previous, at bottom of Products Section, 1 times | A new Grid of caps is shown. Page number is now 3. | Matches expected outcome |
|  | Click “Bottom Trilby” Picture. | Bottom Trilby details are shown | Matches expected outcome |
|  | Click Add to Cart | New item appears in shopping cart: Tall Breaker $16.70  Blue X 1 | Matches expected outcome |
|  | Click Checkout | Checkout page appears | Matches expected outcome |
|  | Click the Red X next to the item “Tall Breaker” | Tall Breaker is removed | Matches expected outcome |
|  | Verify Totals | Subtotal is $67.30, GST is $10.10, Total is $77.40 | Matches expected outcome |
|  | Click “Checkout”. | The Orders page appears.  Message: “**Success!** You order has been placed!”.  The order will be visible in the Orders table.  The order should have today’s date.  The order status should be waiting.  The order Qty should be 3  The order Cost should be $77.40 | Matches expected outcome |
|  | Click Logout button. | The home Page shows. | The home page shows as expected. |
|  | Close all tabs to the site. | - | - |
|  |  |  |  |
| Login as Administrator. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_admin” into login textbox. Type “test\_\_password” into password textbox. | Login control shows “test\_admin” and “test\_\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Orders Admin page appears. | Matches expected outcome |
|  | Examine the Top Menu. | Top Menu shows the logo at left, several links including “Orders”, “Caps” and “Categories”. | Matches expected outcome |
|  | Click Logout button. | - | - |
|  |  |  |  |
| Add new Category. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_admin” into login textbox. Type “test\_password” into password textbox. | Login control shows “test\_admin” and “test\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Orders Admin page appears. | Matches expected outcome |
|  | Click Categories in the top menu. | Categories page appears. | Categories page appears. |
|  | Click “New …” at top right. | The Form Clears. | Matches expected outcome |
|  | Type “Mock Caps” in the Name field. | - | - |
|  | Click “Save” | A new category “Mock Caps” appears in the list at left.  Message: “**DONE: Check list for new category.**” | Matches expected outcome |
|  | Click Logout button. | - | - |
|  |  |  |  |
| Add new cap. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_admin” into login textbox. Type “test\_password” into password textbox. | Login control shows “test\_admin” and “test\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Orders Admin page appears. | Matches expected outcome |
|  | Click Caps in the top menu. | Caps page appears. | Caps page appears. |
|  | Click “New …” at top right. | The Form Clears. | Matches expected outcome |
|  | Click Category Id and select Children’s Caps. | - | - |
|  | Change Supplier to Escobar Fabrics | - | - |
|  | Change Name to “Elevated Beckett” | - | - |
|  | Change Price to 20.00 | - | - |
|  | Change Description to “Lorum Ipsum Sum Ergo Proxy Cowboy Bebop Neon Genesis” | - | - |
|  | Click ImageUrl and from dropdown, click “bamboo-507-ivy-cap.jpg” | (the image at lower right changes to a picture of a cap) | Matches expected outcome. |
|  | Click “Save” | Message: “**DONE: Check list for new cap**” | Matches expected outcome. |
|  | Scroll to bottom of list. | A cap is present at the bottom, “Elevated Beckett”. | Matches expected outcome. |
|  | Click Logout button. | The home Page shows. The top menu says “Greetings Visitor!” | The home page shows as expected. |
|  |  |  |  |
| Change Order Status from waiting to Shipped. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | The login page appears. |
|  | Type “test\_admin” into login textbox. Type “test\_password” into password textbox. | Login control shows “test\_admin” and “test\_password” in relevant textboxes. | Login control matches the expected outcome. |
|  | Click Login button. | Orders Admin page appears. | Matches expected outcome |
|  | Click “9, waiting” in list at left | Order ID: 9 is loaded.  Message: “Ready” | Matches expected outcome |
|  | Click “Ship Order” | Status changes to “shipped”, both in the details form, and in the list at left. | Matches expected outcome |
|  | Click Logout button. | The home Page shows. The top menu says “Greetings Visitor!” | The home page shows as expected. |
|  |  |  |  |
| Disabled Account login. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click “Login” at top menu. | The login page appears. | Matches expected outcome |
|  | Type “disabled\_customer” into login textbox. Type “test\_\_password” into password textbox. | Login control shows “disabled\_customer” and “test\_\_password” in relevant textboxes. | Matches expected outcome |
|  | Click Login button. | Message, in red “This account is disabled. If you believe this is in error, contact the Admin at AskewR04@myunitec.ac.nz immediately.” Appears, at login page | Matches expected outcome |
|  |  |  |  |
| Session Timeout. | Enter the site URL into the browser address bar and press enter. | The home page appears. | The home page appears. |
|  | Click on the picture “Men’s Caps” in the Category section. | A new Grid of caps is shown. Page number is now 1. | Matches expected outcome. |
|  | Click on the picture “Top Squat” in the products section. | Section appears with following details: Top Squat, $11.90. | Matches expected outcome. |
|  | Click Add To Cart. | There is now 1 entry in the Shopping Basket. Products list appears. | Matches expected outcome. |
|  | Hold CTRL and press F5. | Browser is forced to refresh.  The home page is showing.  The Shopping cart is unchanged. | Matches expected outcome. |
|  | Wait 30 minutes. Do not do anything in the browser tab. | - | - |
|  | Hold CTRL and press F5. | Browser is forced to refresh.  The home page is showing.  The Shopping cart is empty. | Matches expected outcome. |
|  |  |  |  |

# Critique

## 1. ASP.NET / PHP Comparison

### 1. Introduction

This is a discussion to compare the ASP.NET and PHP Web Application languages.

ASP.NET is a web server-side framework for creating web applications, with C# acting as code-behind / backend scripting, and HTML, CSS and Javascript used as frontend languages. ASP.NET also provides additional Web Controls, which can be utilised alongside HTML. Note that the code-behind also supports VB.NET but for the discussion only C# will be examined.

PHP (PHP Hypertext Processor) is a general programming language, but designed for website and web application development [Gutmans, Bakken, & Rethans (2004)]. It has a unique syntax and grammar, with similarities to C and Python.

### 2. Database Techniques

ASP.NET can connect to a range of database systems including ODBC, Oracle and MySql based systems, but is most compatible with Microsoft provisioned database systems such as SQL Server and Access.

ASP.NET can connect through the ADO.NET framework, which provides objects to manage connections, data translation, and other requirements. Additionally there is available the Entity Framework ORM, by which database interaction can occur using business objects which represent database entities. ADO.NET abstracts the developer from the lower level DB management issues, and Entity Framework can abstract further so little coding of a DB interaction system is required, and more focus is given to modelling business objects. Connections to web services as data sources is also possible.

There are a range of techniques for coding Database Sources and Data presentation through ASP.NET, including specific web controls and code-behind objects and techniques. A developer is free to choose between them and combine them as desired. Web Controls specific to data presentation can be bound directly to data sources. Code Behind techniques can include direct execution of SQL (encapsulated in an ADO.NET object), the use of LINQ queries, or business object / ORM actions.

PHP has available database extensions to allow interaction with MySQL, MariaDB, SyBase, SQL Server, DB2/Derby, Oracle and many others. Additionally PHP (as of version 5.1) has special extensions for ODBC data-access abstraction, and a generic data-access extension (PHP Data Objects, or PDO) for accessing databases in general (PDO sometimes requires additional database drivers specific to the database being used) [PHP Database Extensions. (2016)]. This allows PHP to connect to any database.

PHP database interaction can be different for each extension. For example, the PostgreSQL extension consists mostly of functions, with a connection object and query result object used as parameters or returned objects. Compare this to the Mysql extension (mysqli) which uses multiple objects for connections, prepared statements, and more, and interacts through methods invoked on these objects. The PDO extension offers a consistent interface used regardless of the underlying database system.

No special frontend controls are provided for data sourcing, data presentation and binding.

ASP.Net seems to connect to fewer databases but can connect to the more common enterprise-level systems. It offers a more consistent interfacing through ADO.NET and Entity Framework. There are specialised controls available for data sourcing, binding and presentation, but more experience is required of these controls to use them effectively. PHP can connect to more databases, can offer specific extensions or a consistent interface through PDO, and lacks specialised data interaction controls but does not have the experience issue.

### 3. Client Techniques

ASP.NET provides several web controls alongside native HTML controls. In many cases a developer simply needs to introduce a web control and then tune it via properties.

ASP.NET provides ways to manage AJAX requests and responses. A developer may not need to write additional scripts or code files, and may not need to write Javascript for AJAX call management. In some cases a ScriptManager control is required, and Update Panels are required for sections which will be re-rendered using AJAX. This simplifies AJAX development, but requires more knowledge of these additional controls and techniques.

ASP.NET allows client-side scripting in languages other than javascript, such as VB and C#. These scripts can be utilised in the same way as Javascript scripts, including hooking scripted functions to control event callbacks. An example is writing a C# function called ‘Button\_Click’ to write content to the Response, and then adding ‘onclick=”Button\_Click”’ to the properties of an input button. This allows a developer to create client-side scripts even if they know more C# or VB than Javascript [ASP.NET Recommended Resources. (2016)].

ASP.NET permits extending existing client-side controls to create new controls.

ASP.NET provides web controls for storing data on the client such as HiddenField. Also it provides a viewstate to store data across the HTTP Response – Request cycle. Data stored in viewstate will be available to retrieve after a postback of the same page (reloading after a form submission, for example). There are techniques to use cookies stored at the client.

Some third-party client scripts and CSS libraries have to be loaded through extension packages (via NuGet) to work properly, especially if AJAX is used in a page.

A developer will require good knowledge of ASP.NET web controls to use then effectively. Additionally, ASP.NET has to re-render the provided web controls into native HTML before sending response, which takes additional processing time. Using third party client libraries can require extra effort and knowledge.

PHP does not provide additional web controls or the option to extend controls. A PHP developer must have good experience with native HTML, CSS and Javascript, and PHP development. AJAX calls require Javascript coding and PHP scripting to handle callbacks and content rendering. The client side scripts supported depend on the client browser capabilities, but are otherwise not affected by PHP. Third-party scripts and CSS libraries (such as bootstrap) can be used independently of PHP.

PHP can use cookies stored on the client, or store information in the DOM window via javascript.

Less knowledge overall is required to understand Client side techniques for PHP, and using third party client code is much simpler.

### 4. Software Architecture

Both PHP (as of version 5) and ASP.NET (discussing C#) allow good code reuse, through namespaces and object oriented techniques to organise code and manage application behaviour.

Both PHP and ASP.NET permit additional packages and extension libraries. ASP.NET uses the NuGet package system maintained by Microsoft, while PHP uses PEAR which is maintained by the open-source community [Gutmans et al (2004)].

ASP.NET requires Windows and IIS servers to run. PHP can be run with IIS, Apache, and other web server systems, and on Windows, Linux, Mac OSX and other Operating systems.

ASP.NET uses a combination of front-end controls and back-end coding. This can be separated into different files (.ASPX / ASPX.CS for web pages / code-behind respectively, and .CS for independent back-end code). Additional files are needed for configuring the base Web Application and managing global application state. PHP simply has backend code which can be used alongside frontend elements, or written independently (always in .PHP files).

The VB and C# languages used for ASP.NET are strongly typed – The system needs each variable and object to have a specific type declared. PHP variables and objects are weakly typed by comparison, though there is the option to force a particular type if needed.

ASP.NET offers a global application state shared by all clients. PHP has a global state defined at the web server level. Both offer a semi-global session state per client.

Both offer “super global variables” which represent request and response states, postback data, query string data, and more (for example, Response and Request for ASP.NET, and $\_GET, $\_POST, $\_SERVER for PHP) [Gutmans et al (2004)].

PHP has magic methods beginning with ‘\_\_’. Some of these replicate the required constructors, destructors, etc for class coding, and others permit additional functionality, such as \_\_clone to define how objects are copied. C# lacks magic methods but has some required methods which match the functionality described above.

PHP permits coding in procedures, objects or a combination of both. C# lacks prodecural coding but allows static objects which can store pseudo-procedures as static methods.

ASP.NET has multiple technologies, being ASP.NET MVC, ASP.NET Webforms, and ASP.NET Web Pages (also called Razor). Each can require different techniques of design and coding, for example MVC requires Model objects representing database entities, while Webforms requires each webpage to contain a form element. PHP does not have such restrictions, and developers are free to code the frontend and back-end as desired. Frameworks for PHP MVC exist, such as CakePHP, but are not required for web application development.

PHP can be developed in a range of tools, including NetBeans, DreamWeaver and more recently PHP-Storm [Best PHP IDEs, Survey Results (2014)]. ASP.NET development is limited to Visual Studio.

PHP is open source and free. ASP.NET can be used free for community / non-enterprise level purposes, but enterprise level is costly. Additionally, the main ASP.NET development tool, Visual Studio, is costly to licence for enterprise purposes.

While all OSes and web server systems have vulnerabilities, Windows and IIS systems tend to be more vulnerable due to how user privileges are assigned, and the presence of a ‘monoculture’, meaning that different forms of IIS and Windows systems have the same underlying fundamental backbone, and thus the same risks present in this backbone. This means ASP.NET, while not necessarily insecure in itself, has a higher security risk by being confined to IIS / Windows servers. PHP can be used on non-windows systems, so security risks are more likely from how the application is implemented than the underlying system.

### 5. Summary

ASP.NET allows more rapid development through the use of front-end web controls and client techniques. It has useful controls dedicated to sourcing and presenting data from databases. It provides a range of techniques for database interaction, and can connect to most enterprise level DB systems. ASP.NET requires C# or VB experience, object oriented development experience, and specific knowledge of the web controls provided and techniques for client scripting and AJAX management. ASP.NET is limited to Windows and IIS systems, and is less secure due to issues with these systems. ASP.NET can be expensive to use at enterprise level.

PHP can connect to many more database systems, and provides both dedicated extensions and a general purpose interface through PDO. It can be implemented cross-platform on many systems, does not require any specialised experience apart from frontend technologies, is low to no cost to acquire and use, can be used in object oriented or procedural form, and security risks are limited to the application development.

### 6. Preference

I would prefer PHP due to simplicity, cross platform capabilities especially with Linux systems, and being open source and no cost, and additionally from previous experience with PHP development, I prefer to use frameworks such as CakePHP and content management systems such as Wordpress and Magento.   
All the same, I enjoyed experimenting with Entity Framework and the challenges of C# coding / using data based web controls, and I may be open to making ASP.NET MVC web applications using these technologies, in future.

### References

Gutmans, Bakken, & Rethans (2004). PHP 5 Power Programming. Hagerstown, Maryland: Prentice Hall.

ASP.NET Recommended Resources. (2016). Retrieved from <https://www.asp.net/whitepapers/aspnet-data-access-content-map>

PHP Database Extensions. (2016). Retrieved from <http://php.net/manual/en/refs.database.php>

Best PHP IDEs, Survey Results. (2014). Retrieved from   
<https://www.sitepoint.com/best-php-ide-2014-survey-results/>

*ASP.NET vs PHP.* (n.d.). Retrieved from Orient: <http://www.orientsoftware.net/technologies/microsoft-net/aspnet-vs-php>