**IE4727 Web Application Design - Project Report**

**Design project group number:** F36-DG17

**Team members:**

1. Muhammad Nabil Bin Mohamad Ali (U2222720J)
2. Ahmadul Matin Bin Ahmad Kamal (U2121585C)

**Project Title:** Element Electronics

### Summary of Project:

This project aims to develop a comprehensive e-commerce website that serves as a one-stop-shop for users to purchase electronic goods. The website provides a seamless user experience, where customers can browse through a wide range of electronic goods such as laptops, monitors and accessories. Furthermore, customisation options such as upgrading storage and memory will also be made available for products that offer such customisations. Users will have the ability to create their own user accounts, so as to store checkout details such as their address and payment options.

The shopping experience will be further enhanced with the “cart” page where users can review their items, adjust quantities and proceed to checkout. During checkout, users can either enter new payment and shipping details, or select the ones that are already saved to their account. This ensures that users have a seamless shopping experience, reducing the hassle of checking out. Upon successfully completing the purchase, the system will automatically send a confirmation email to the user, providing them a receipt and details of the order.

The website aims to offer a user-friendly interface that simplifies the process of shopping for electronics, while also providing essential e-commerce features like account management, a product catalogue and email confirmation all in one platform.

# **Application Requirements and Specifications**

This application is designed to provide a one-stop-shop for users to purchase various electronic goods online. It allows users to browse through a wide range of electronic products, manage their cart and receive an email confirmation for their purchases. It also provides users with a seamless experience throughout their use of the website, from account creation to checkout.

#### Application Requirements:

* **R1:** The application must present an extensive catalogue of electronic goods in neat categories.
* **R2:** The product catalogue must be filterable by categories, brands and price range.
* **R3:** Users must be able to view detailed product information on individual product pages.
* **R4:** Customisation options must be available for products that offer such options.
* **R5:** Users must allow users to create and update accounts.
* **R6:** The application must provide a user login and authentication system.
* **R7:** A “cart” feature where users can add, remove and modify their potential purchases.
* **R8:** User are able to make payment and checkout their items
* **R9:** An email confirmation containing the purchased items and receipt must be sent to the user once a purchase has been completed.
* **R10:** A search function will be implemented for users to quickly search for a product.

# **Functional Requirements and Specifications**

#### F1: Product Catalogue Browsing (R1)

* Product catalogue pages will display products that are grouped into different categories such as laptops, tablets, monitors etc. This allows easy navigation for users who are looking for a class of items.

#### F2: Sorting and Filtering (R2)

* Within each category page, users will be able to filter products by categories, brands and price range. Filters (which will be presented in check boxes) will be present on the left sidebar on the product catalogue page.
* Sorting options such as “Price: Low to High”, “Price: High to Low” or “Latest First” will be available.

#### F3: Product Details and Specifications Pages (R3)

* Each product listed will have its own page displaying details like specifications, dimensions, reviews and product images.
* Users will have the option to add the product to their shopping cart directly from this page.

#### F4: Customisation Options (R4)

* For products that offer customisation options (eg laptops), users will be able to click on the different options in the product details page.
* Depending on the choices selected, the final price of the product will be updated after each click.

#### F5: Account Creation and Update (R5)

* Users can register an account by providing an email and a password.
* After creation, they can later log in to update their personal information such as their shipping details and payment details. This information will be used to provide a seamless checkout upon purchase.

#### F6: Login and Authentication (R6)

* The application will provide a secure login system. Sessions will be maintained for logged-in users to enhance user experience.

#### F7: Virtual Cart (R7)

* Users will be able to add products into a virtual cart. When the cart web page is accessed, users can review their items, adjust quantities and delete items.
* The cart page will display the total price of all the items in the cart and allows users to proceed to the checkout page.

#### F8: Payment and Checkout (R8)

* A secure checkout page will be provided for users to enter their payment details and shipping information. Contact details such as email address and phone number may also be included.
* For users with accounts, they can select the saved shipping information and payment details.

#### F9: Purchase Confirmation Email (R9)

* After a successful purchase, the application will send an email to the user’s email address. This email address is obtained through the contact details they have put in in the checkout page.
* Email will include the purchased item list and a receipt.

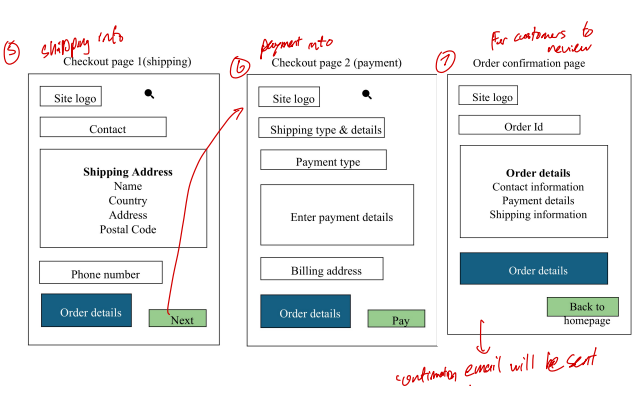
#### F10: Search Function (R10)

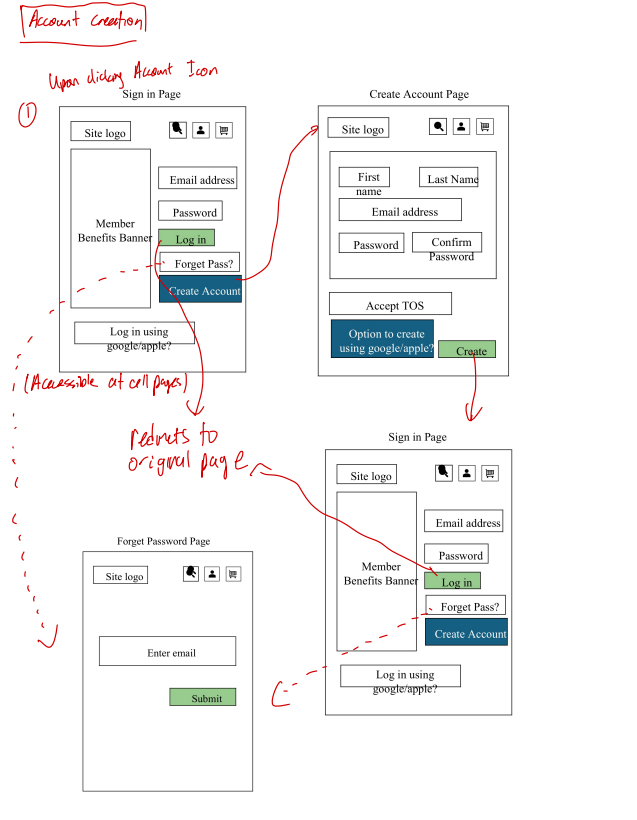
* A search function will be prominently displayed on the website, allowing users to quickly search for products by entering keywords or product names. A dropdown of the 3 most relevant items will be shown, and the user will be redirected to a page of all relevant items if the users click the “search” button.

# **Design of the web application**

### 3.1 Site map

### 3.2 Storyboard





### 3.3 Wireframe

### 3.4 Database

For an e-commerce website such as this one, a robust database will be required to handle all facets of the application, such as the users, products, orders and transactions. Below is an overview of the key components of the database design

#### User Database

This database stores information about users, such as their username, email, shipping and billing address as well as their contact details

#### Product Database

This database stores information about the products, customisation options and categories.

#### Orders Database

This database stores information about user’s orders, the contents within their carts as well as payment details

#### Payment Database

This database is meant to create a ledger of the payments made and stores information such as the payment methods, transaction dates and the total amount.

### 3.5 Web application testing plan

The aim of this testing plan is to ensure that the web application we built is functional and user-friendly. We are required to test across different devices, browsers and platforms with different screen sizes as different devices have different compatibility.

#### 3.5.1 Testing phases

Testing of our web application will be carried out in the order as shown below. We will conduct 4 different types of tests to identify bugs, triage them and ensure that the application works as intended.

#### Functional Testing

Functional testing is an essential form of testing in Software Development that helps to verify if the feature of the web application developed matches the acceptance criteria that had been specified. For what we are developing, we have to ensure that the main features are stable and working as expected. The features include User login & registration, product search, viewing, adding of items to the cart, payment process, order confirmation and user profile management.

#### UI/UX Testing

This type of testing verifies the User Interface layout and its design elements, by checking whether it matches the design specifications. It is an important method of testing as we are able to evaluate the User Experience or friendliness of our application.

#### Regression Testing

Regression makes sure that existing functionalities are still working even after new codes are being introduced. It ensures that the whole application remains stable and function as expected. Usually performed to verify that there are no new defects in the existing functionalities whenever there are changes in the application or bug fixes. This is the most important type of test, especially before every build release.

#### User Acceptance Testing (UAT)

This is the final type of test that we intend to carry out. End-users, which are consumers of electronic products, will be testing our application and identify issues within the application. They will be carrying out the test cases that we have planned and provide their feedback or experience. We will then decide whether to improve the application based on their test case results and feedback.

#### 3.5.2 Test cases

Test cases are a specification of actions that needs to be executed whether manually or by automation. They check the functionality of a particular feature and verify if the objectives or expected results are met.

For our project, we need to create specific test cases for each of these features: User login & registration, product search, viewing, adding of items to the cart, payment process, order confirmation and user profile management. Simple test cases that we have drafted are as follows.

* **User login & registration**
  + Verify that user is able sign up using a new email address with a correct format
  + Verify that user is able login using an existing email address
  + Check that there is an error message if incorrect email/password is entered
* **Product search**
  + Verify that when user queries on the search bar, top 3 relevant products will be shown in a dropdown
  + Verify that when user submits their search, they will be redirected to the search page and relevant products are displayed
* **Viewing of products**
  + Verify that products are displayed according to the category selected
  + Verify that when user clicks on a product, they will be redirected to the product detail page and they are able to customise their product for order
* **Cart**
  + Verify that if user clicks on the “Add to cart” button, then the product will be added to cart
  + Verify that user is able to proceed to the checkout page if they click the checkout button from cart page
* **Payment**
  + Verify that user is able to successfully pay using different payment methods
  + Verify that if required fields are not entered, user is unable to successfully order their item
* **Order confirmation**
  + Verify that after payment, the user is directed to the order confirmation page where they can see their shipping address, payment and order details.
  + Verify that a confirmation email is sent to their email address
* **Account management**
  + Verify that user is able to edit their profile (name,e-mail, password)
  + Verify that their order history is displayed with the correct details

#### 3.5.3 Bug Triage process

After testing has been completed, bugs will be identified. Triage is the process of evaluating the bugs and prioritising them, in order of the criticality of the bug or how severely the bug impacts our web application and its important features. We will fix the bugs according to its priority, after the triage process.

# Implementation

Throughout the project, we had to ensure consistency of the code so that all the webpages follows the same layout, so that our webpages functions similarly throughout the entire website, for a cohesive and intuitive user interface. Having a uniform layout, maintaining design elements and coding practices throughout the pages, users can easily navigate the site and interact with the implemented features without confusion. It is also important for us to be consistent with styling choices such as uniform color schemes, fonts, headers and footers, so that the website appears reliable.

More importantly, it is important for us to maintain a consistent coding practices so that we can collaborate more effectively. Having consistent coding practices reduces the risk of errors and streamlines future collaborative efforts.

### 4.1 Coding of Web Page Template

The most important aspect of ensuring the consistency across all the webpages is the use of a common template structure. This includes the header, navigation bar, footer and stylesheet being consistent throughout the pages. These parts of the webpages are coded in a consistent manner and reused across various pages of our website.

Template Components:

1. Head
   * The head portion of each page provides a consistency of the font and stylesheet used. It also allows for the consistency of the code

A black screen with orange text

Description automatically generated

1. Header:
   * The header of our webpage includes the company logo, the navigation bar, as well as the search, account and cart functions. The use of the navigation bar is to ensure that users are able to browse through important webpages easily and the access is available readily.
   * We hardcoded those aspects of the website in HTML so that there is little chance of bugging. It also allows us to easily fix the positions of those components so that it remains consistent.

A screen shot of a computer program

Description automatically generated

1. Footer
   * In our website, the footer provides some basic information such as copyright details as well as a subscription form for users to enter their email addresses to subscribe to the newsletter.

A computer screen with text

Description automatically generated

1. Stylesheet
   * The styles.css file is used to maintain a consistent appearance throughout the website. It handles layout, sizing, spacing and the responses that the website gives the user.

### 4.2 Coding of the Pages

The website itself has many different components and pages. While embarking on this project, we had to prioritise the creation of important pages such as the home, catalogue, products and checkout pages first, as those pages are the ones that creates the user experience and is essential to the sales of the electronic products. After those parts are created, then we moved on to the creation of pages that are not essential to business continuity, such as the contact page and the about us page.

* + - 1. **Home Page (homepage.php)**

The homepage serves as the landing page for the user, introducing them to the brand that is Element Electronics as well as providing quick link to featured products as well as the product catalog.

**Fetching User and Cart Data**

A computer code with colorful text

Description automatically generated

The webpage will first check if the user is logged in by verifying session data. If the user is logged in, the user ID is retrieved. Otherwise, it defaults to 0.

**Carousel**

A screen shot of a computer program

Description automatically generated

We added a carousel in the home page which displays promotional images and text in a sliding format. We chose to implement a carousel in the home page to add a layer of interactivity, as well as a way of displaying featured products and links without cluttering the outlook of the page.

**Displaying Featured Products**

A screen shot of a computer code

Description automatically generated

Below the carousel, we added in a section which features several products. The product loop iterates over the array of featured products and displays the products with their images, names, and price. There is also an “Add to Cart” button for each product, which is linked to a JavaScript function that adds the product to the user’s cart.

**JavaScript “Add to Cart” Function**



A screen shot of a computer screen

Description automatically generated

The function will verify whether the user is logged into the site. If the user is not logged in, an alert will be shown and the user will be redirected to the login page.

Upon clicking “Add to Cart”, the webpage sends a POST request with product details such as the product ID and configuration options that the user has selected to add\_to\_cart.php. There will also be an alert to display whether the product was successfully added to the cart or if there was an error.

**Updating Cart Quantity**

A screen shot of a computer code

Description automatically generated

The webpage will send a request to cart\_quantity\_fetch.php to retrieve the current cart quantity. The cart quantity as displayed in the header will then be updated based on the response.

* + - 1. **Catalog Page (catalog.php)**

The catalog page lists all the available products. The products are dynamically fetched from the products table of our database using PHP and MySQL. Each product will have its own image and “square” on the catalog page. Details such as its name, description and price will also be displayed.

A black background with different colored text

Description automatically generated

The first part of the code is the pagination setup. This defines the number of products per page and determines the current page based on the “page” parameter in the URL.

A computer code on a black background

Description automatically generated

The next part is the Category and Brand fetching. This portion queries the database to fetch distinct categories and brands for filtering options. It also allows for users to filter products by categories and brands using checkboxes.

A screen shot of a computer program

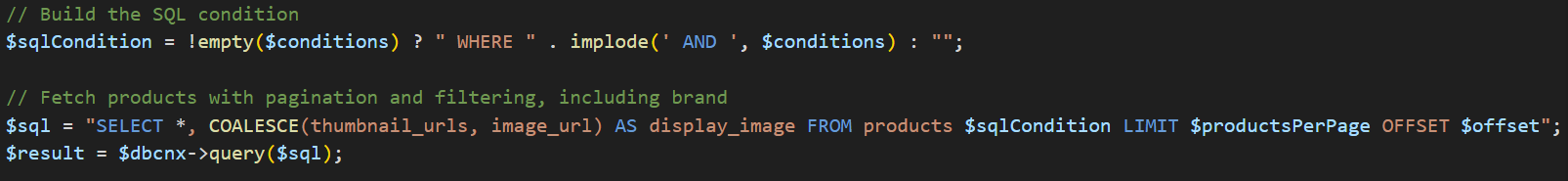
Description automatically generated

This portion is to create filters for the users. If categories are selected via the checkboxes in the sidebar, the code adds them to the “conditions” array for the SQL query. The same goes for the brand if the user chooses that.

A screen shot of a computer program

Description automatically generated

The next portion of the code is a price filter which allows users to set a minimum and maximum price range. We also ensured that the provided prices are numeric to avoid any invalid inputs.



This portion constructs the SQL query by combining all filter conditions. It then fetches product data from the “products” table based on the conditions that the user has inputted.

A screen shot of a computer code

Description automatically generated

The products will then be displayed in the HTML. We used a while loop to iterate over the fetched products and display them. As mentioned, the product’s image, name, description. Price and a “View Details” will be displayed within the “square” of the product.

A screen shot of a computer code

Description automatically generated

Lastly, this part of the code is an “Add to Cart” button. Upon clicking the “Add to Cart” button, the website sends a “post” request to add\_to\_cart.php with the product details. If the request is successful, the cart’s quantity will be updated with an alert saying, “Product added to cart!”. If not, there will be an error alert saying, “Failed to add product to cart.”

* + - 1. **Product Page (product.php)**

The product.php is aims to display detailed information about a product. It retrieves product data from the database based on the product ID provided in the URL and showcases features such as images of the product, product description, pricing and configuration options (eg colour and storage). When users navigate to a product page, they will be informed about the visuals and specifications of the product and will be informed enough to decide whether to add the product to cart.

**Fetch Product Details**

A screen shot of a computer program

Description automatically generated

The first part of the code retrieves the product ID from the URL. When user clicks “View Details” on a product at the catalog.php page, it creates a GET request type. The webpage will also fetch product details based on the product\_id.

**Handle Product Data**

A screen shot of a computer code

Description automatically generated

The next section of the code is to handle the product data based on what was fetched from the database. Firstly, it verifies if the product exists in the database. After that, the code assigns product details such as the name, description, category and price to variables for display. It also uses the first image from thumbnail\_urls array as the main product image that the user sees.

**User Cart Quantity Checker**

A computer code on a black background

Description automatically generated

Then, the webpage will determine if the user is logged in. If he is, it then fetches the total quantity of the user's cart.

**Display**

A computer code on a black background

Description automatically generated

This portion displays clickable thumbnails using a loop through thumbnailUrls. It shows the main product image, which then updates when a thumbnail is clicked.

**Product Configuration Options**

A screen shot of a computer code

Description automatically generated

The next part of the code serves as the configuration options that users can select to configure the electronic device that they are viewing. It provides dropdown menus for selecting colour and storage options, and its only shown for configurable devices such as phones, laptops and tablets.

**Dynamic Price Update**

A computer screen shot of text

Description automatically generated

The next important feature is the dynamic pricing update in the product page. This JavaScript function called updatePrice adjusts the final price of the product based on the selected storage option and quantity. It calculates the final price using a multiplier for each storage option as well.

**Add To Cart**

A screen shot of a computer code

Description automatically generated

When the user clicks the Add to Cart button, the webpage sends a POST request with product details such as the product ID and configuration options that the user has selected to add\_to\_cart.php. If successful, the alert “Product added to cart!” will be returned. If not successful, the website will alert the error message “Failed to add product to cart.” This will provide clear alerts for success or failure.

* + - 1. **Checkout Process**