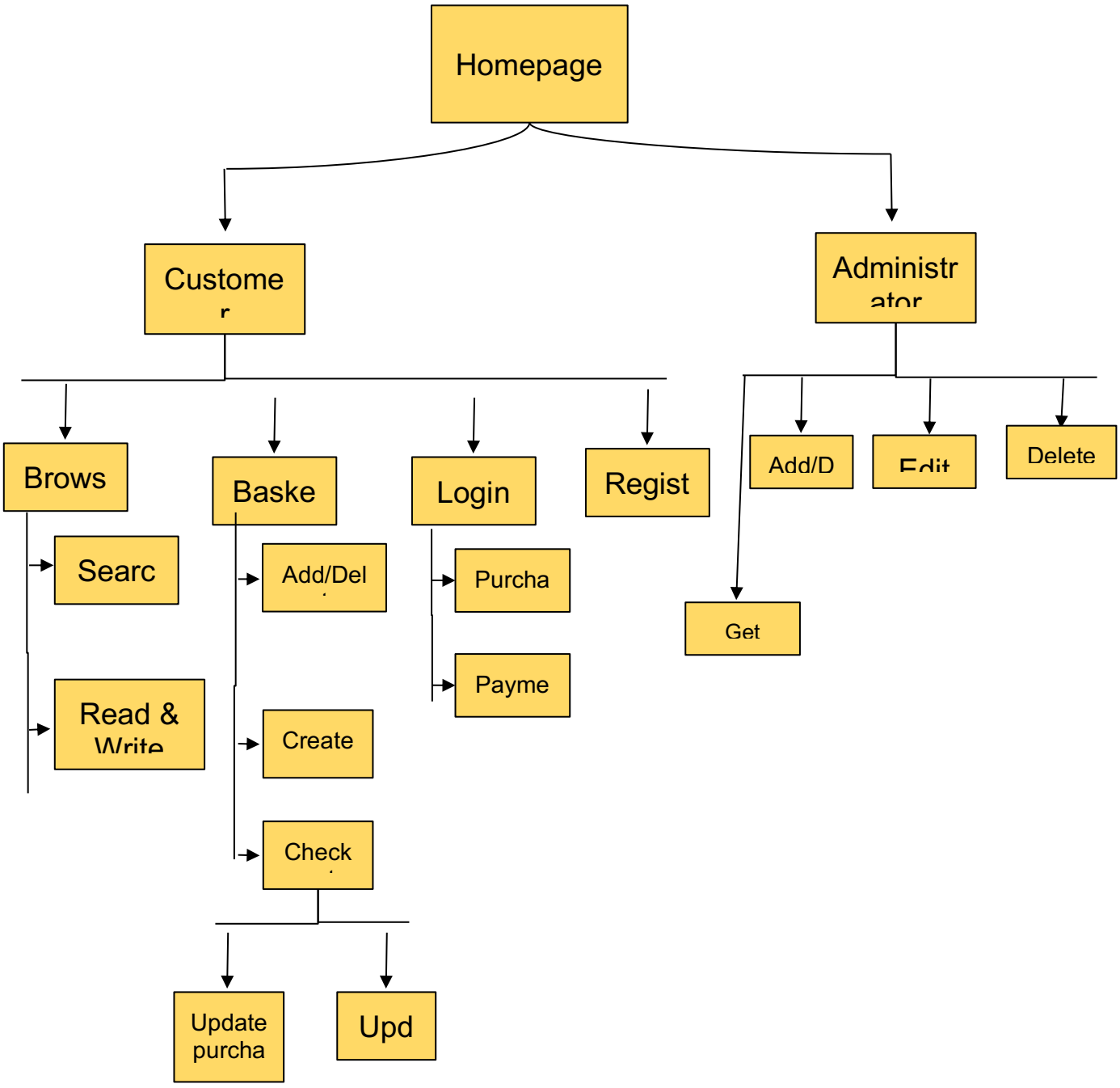
Design

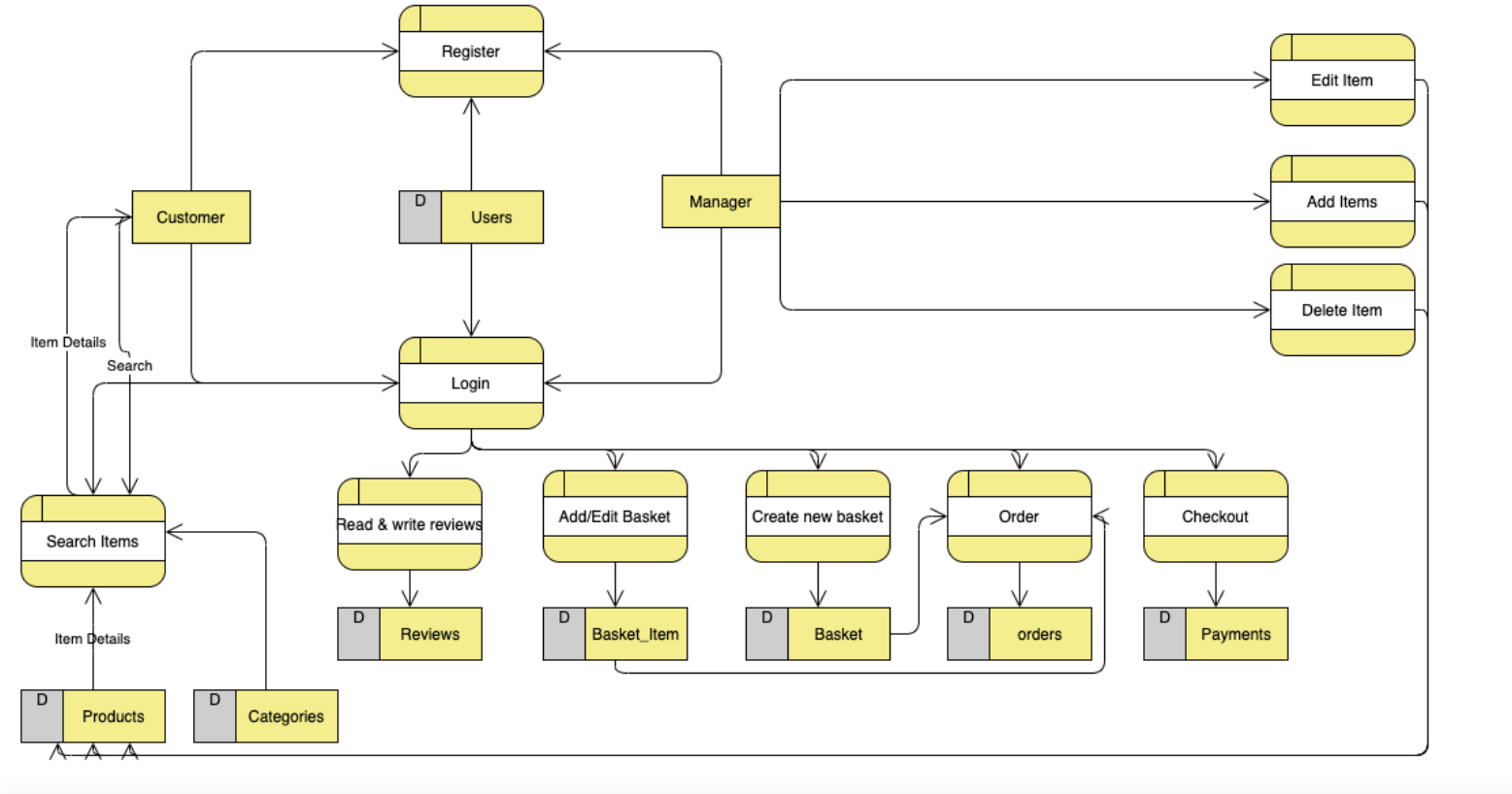
Modular Diagram:

A modular diagram is used to show the different blocks of code in a program and how each block of code can interact with another. A modular Diagram is similar to a flowchart as it contains arrows to show each module that interacts with another. This helps the developer get a clear understanding of the structure of the program, so when it comes to coding, the developer can use this diagram to identify how each module interacts with another and to determine how the user can access each module.



This modular diagram will help me when I start coding the program as it:

- displays the main modules of my website so I can identify what functions I have to build.
- Shows how each module can be accessed.
- Shows how data should travel between different modules.



This diagram will allow me to understand how the website should be built and in what order, also it allows me to identify what research I should do to implement the different modules.

Data Flow Diagram (DFD):

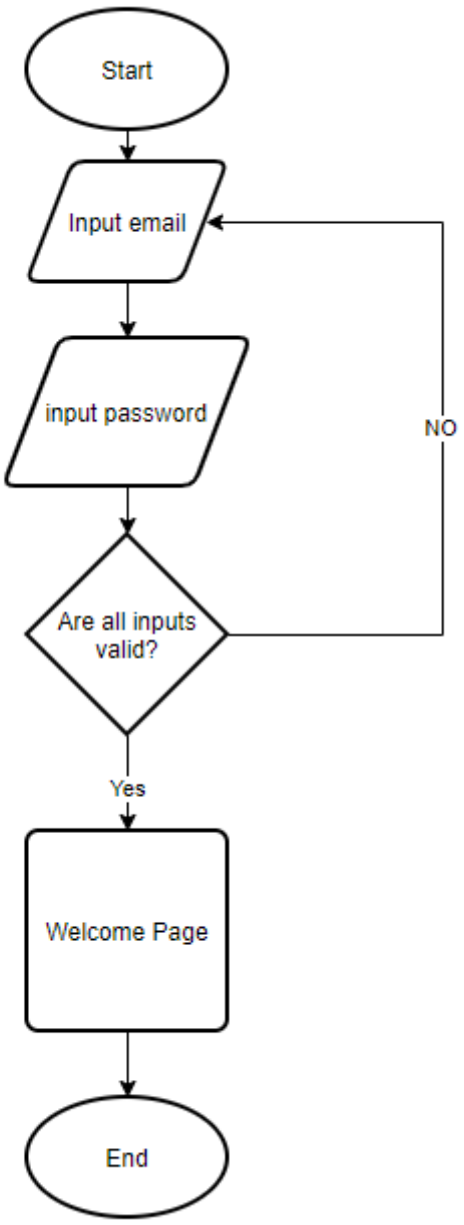
Data flow diagram allows you to have a better understanding of the system and the kinds of processes there are such as registration, login etc. The diagram displays what data is being retrieved and passed from and to each database table. Therefore, it helps us identify who can access which table.
The above diagram is the data flow diagram for my project which is an e-commerce website. As we can see there will be 2 external entities which are the Customer and the Manager. Customers will have different types of processes they can do after they log in(e.g. Read and write reviews, interact with the basket, order and pay for products), but they can only do one process if they haven't logged in which is search for an item. On the other hand, managers will be able to update stock by adding, deleting or editing a product and therefore we can see they will only be able to access the products table.

This diagram will help me identify which specific database tables I need for each processor or feature I want to implement. In addition, the diagram shows me what data has to be passed to and what data has to be retrieved from the database when a user commits an action. Therefore, when it comes to coding, this will allow me to break down every feature into a series of steps which makes the development more efficient and more accurate.

FlowCharts:

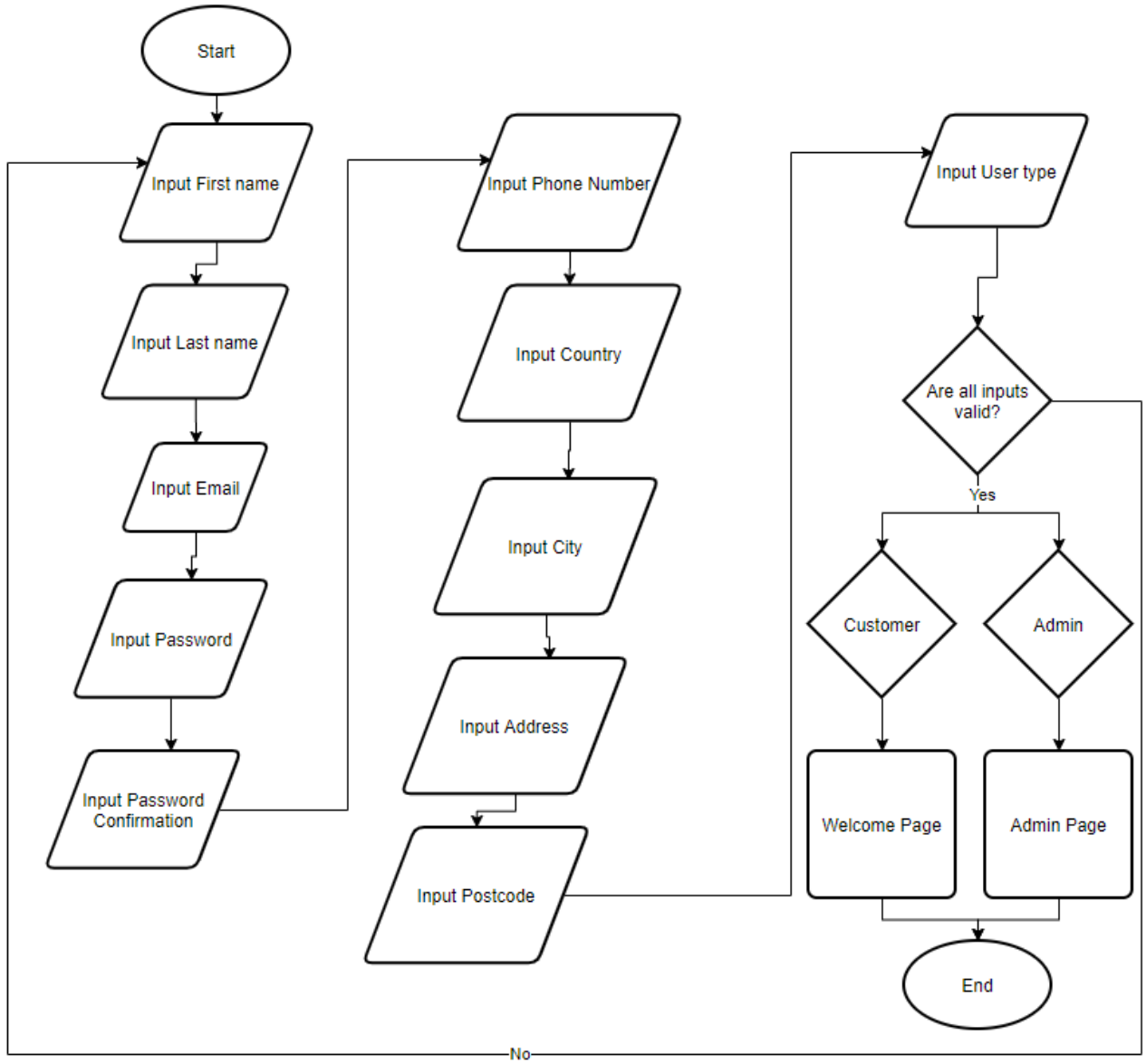
A flowchart is a diagrammatic representation of an algorithm. A flowchart diagram shows a step-by-step process to solve a problem. The diagram helps a developer understand how to approach a problem, therefore, it will allow them to make better decisions within the system. In addition, the flowchart will help them identify any inputs or IF statements they will need to use when developing.

Login Flowchart:



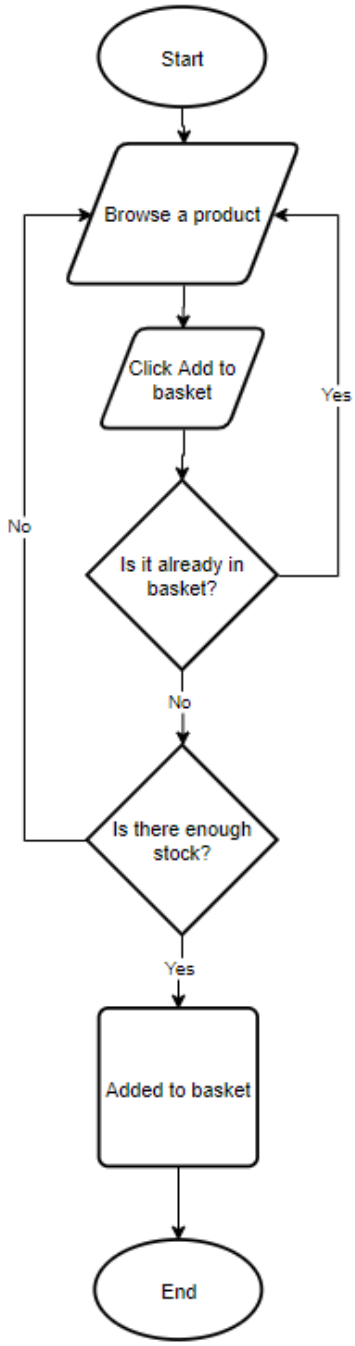
This flowchart shows what should occur when a user tries to log in. The user inputs email and password, then the system checks if the inputs are valid, if yes then the user gets access, if not then it redirects the user back to the login page.

Register Flowchart:



This registration diagram is similar to the login flowchart diagram, however, the only difference is that it has more inputs, and at the end after checking if inputs are valid, it does one additional check to verify the user type and based on that the user gets access to the correct page.

Basket Flowchart:



Conclusion:
The 3 diagrams above show how the features Login, Register and Basket should be approached and what kind of decisions I need to make when it comes to coding these features. This will help me in the development process as it provides me with a clear understanding of the inputs (variables) and IF statements I will need to use. For example, when I made the flowchart for the Basket, I realised that when a user clicks “Add to basket”, the system has to check whether the item is already in the basket or is there enough stock etc. These validations will provide the user with a better experience and it makes the system more efficient and complex.

Pseudocode:

Pseudocode is a description of the source code of a computer program or an algorithm in a language easily understood by humans. It uses the general structure of a programming language but ignores details that are required by machines to execute the code. Pseudocodes are important to create at the design stage as it allows developers to understand the logic of an algorithm and how it works without being distracted by programming language syntax, also they can be easily translated into an actual programming language as it is similar to a programming language.

Search bar:

```
If searchBar == "" AND searchButton == clicked:
   print("Product Not Found")
Else:
   productQuery = Select * From Products Where productName Like '%what has been
entered in the search bar%'
END-IF

If productQuery.rows > 1:
        Output productInformation
Else:
        Print(“Product Not Found”)
END-IF
```

Checkout:

```
If user != loggedIn AND checkoutButton == clicked then
        print(“You must log in to proceed to checkout”)
Else:
        If POST_DATA['bill_info'] is set then
          Foreach field in POST_DATA['fname', 'lname', 'address', 'city', 'cc_type', 'state', \
                        'postcode', 'cc_num', 'phone', 'cc_expmonth', 'cc_expyear']
         If field is empty:
           Output "[field] is a required field."
         Else if fields == valid:
                Redirect to confirmation page
         [End If]
      [End Foreach]
```

Deals and Discounts Section:

```
discountedProducts = []
```

```
productQuery = Select * From Products Where Sale == True
discountedProducts.append(productQuery)
For each product in discountedProducts:
        OUTPUT Name
        OUTPUT Picture
        OUTPUT Details
END Foreach
If viewAllProducts == clicked:
        Redirect to Discounts Page
END-IF
```

UI Design & Usability Features:

Header:

This sidebar nav will be represented by an equal sign but with 3 dashes. This will allow the customer to have a better look over all the sections that are in the website as well as all the different categories, therefore they can navigate through the website easily which saves time and make the UI more friendly instead of showing all the sections under the header which might look unprofessional and make it confusing. I put it in this position as the sidebar will be animated from left to right which means it can be collapsible. The reason I will make it collapsible is because I will not make a separate mobile app for the my project, therefore, it allows the customer to use the website in any device which increase efficiency and provide better user experience.

Cate gory Sele ctor

This button will have a drop down menu which allows the user to see what they can access in their account. A drop down menu will make it more efficient and user friendly as its quicker than typing the page they want to access manually (e.g. spelling errors) or redirecting them to another page which can be time consuming.

The font sizes of these buttons will be roughly around 22. This is because after a few tests I discover ed it's the most suitable size for most devices. In addition, each button will have an icon that represen ts it next to it, also when you hover over the button they will change colour to orange.

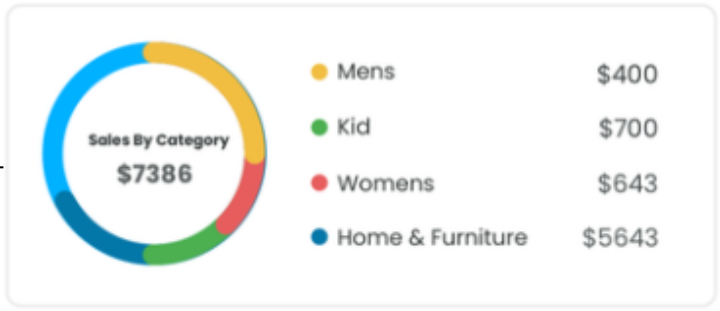
One usability feature in the header is the category selector. I decided to make it a dropdown menu instead of a data entry box where the user has to type the category they want manually. A dropdown menu will reduce the chances of any spelling mistakes, also the user might not be able to identify in which category they want to search, so the dropdown menu will provide them with a list of categories that they can select. Therefore, this will make it more user friendly and increases efficiency and provides the user with a good experience.

Home Page:

This section will display any items that have sales or discounts on the them with more details about the item(e.g. Name, Picture, Description). The size of this section will be relatively large(approximately 500px * 1000px), so the pictures of the products can be seen clearly as well as having a brief description of the product underneath it, to give the customer a rough idea about the product. There will be another 3 sections similar to this section, but they will display different content such as Big Brands, Bestsellers, categories.

On the home page, one usability feature is the sidebar navigation. This is because it will allow the end-user to browse through the website more easily and efficiently, without having to go back to the homepage every single time they want to navigate through a different category. They can simply open up the sidebar navigation and they will be able to see all the links and sections on the website. The sidebar navigation will be collapsible, this means that it can be opened and closed. This is important because the sidebar navigation will not take up a lot of space on the page, especially in a mobile phone where the screen is relatively small, so having collapsible sidebar navigation will allow all the other elements in the page to be accessible in any device.

Admin Page:



I added this chart to allow business managers/admins to have better understanding of the distribution of sales across different categories, which will help them to make better decision such as supplying more or increasing prices etc. The size of the chart will be relatively large as its the main element in the page, so it must be clear and understandable. Each colour will represent a different category and it will be chosen randomly. The background colour will be white so it suits the colours of the pie charts and will make it visually appealing.

Total number of order:

Total

This box will have a basic but important information, as it gives the manager a rough idea about the progress of the business. Every single element inside will have its own icon in front of it to make it more appealing. Colours will be light grey with black text.

Add Item

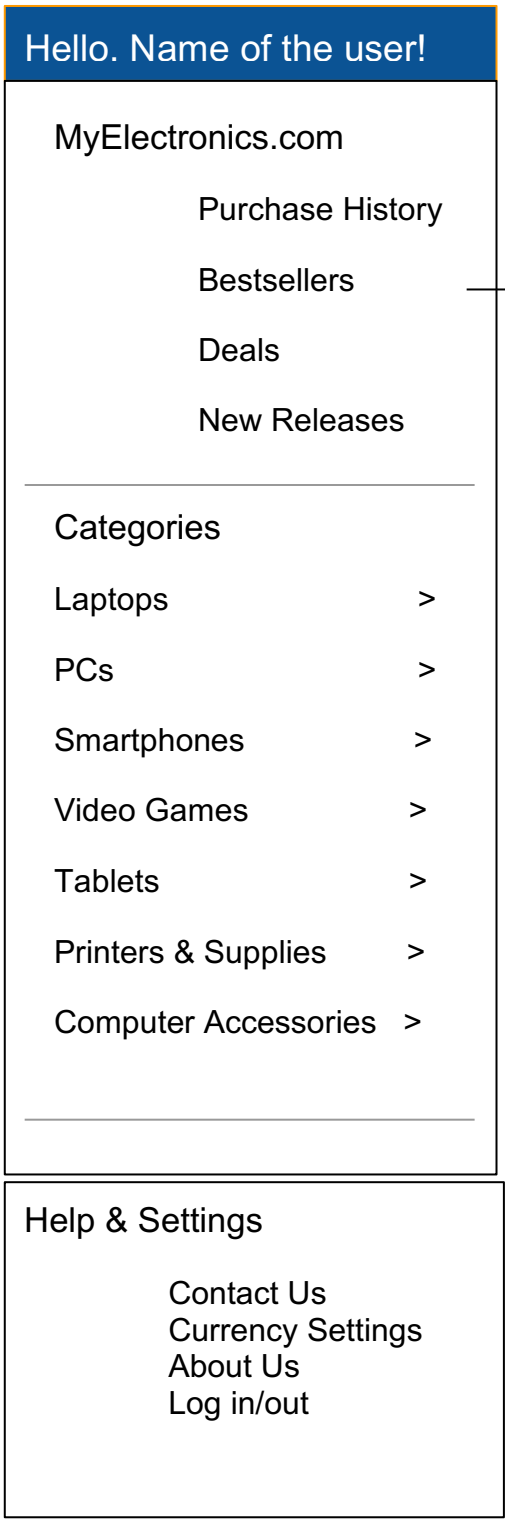
Edit Item

Delete Item

Contact Supplie

This part of the page will contain 4 buttons/links that allows the admin to implement any changes in the items or the website. The buttons/links will have a medium - large size with a clear text and a high quality icon, this is important as will make the website look more interactive and user friendly. The colours of the buttons/links will be light grey with black text and orange icons to represent the theme of the website.

Sidebar Navigation:



The first part of the sidebar navigation contains the main sections of the website, these sections can also be found in the homepage. I added this part to allow quicker access to the sections as the user don't have to always go back to the homepage to access these pages, they can simply open up the sidebar nav and access any section they want.

Data Dictionary:

Product Table:

Field Name	Data Type	Field Size	P/F	Example
ProductID	INT	11	Primary	0
ProductName	VARCHAR	100		iPhone 12
ProductySKU	VARCHAR	50		03600029145
ProductPrice	Float	-		£5.99
ProductImage	BLOB			-
ProdcutDiscription	VARCHAR	500		The phone comes in 6.1-inch size
ProductStock	Float	-		514 units
ProductCategoryID	INT	11	Foreign	4

This table will allow me to store different products so I can display them on my website in order for customers to browse through them and check out the prices, pictures and descriptions of the products.

Product Category Table:

Field Name	Data Type	Field Size	P/F	Example
CategoryID	INT	11	Primary	2
CategoryImage	BLOB			
CategoryName	VARCHAR	50		Video Games

As I mentioned above, there will be many different types of products and items, so in order to keep them organized, each product will belong to a category, therefore I need this table to store the different categories I will need in the website, so when I add a new product I can specify to which category it belongs using this table.

Customer Table:

Field Name	Data Type	Field Size	P/F	Example
CustomerID	INT	11	Primary	1
FirstName	VARCHAR	100		Abdul
LastName	VARCHAR	100		Ibrahim
Email Address	VARCHAR	100		aibraim2015@hotmail.com
Password	VARCHAR	500		1234567Aa
PhoneNumber	CHAR	11		07911123456
Counrty	VARCHAR	20		UK
City	VARCHAR	50		London
Address Line	VARCHAR	100		46 Victoria Road
Postcode	VARCHAR	12		Ha4 0ls
DateJoined	Datetime			19/03/2021

This table is one of the most important tables as it stores the account information of users, this is important because customers would not be able to purchase an item unless they have an account, therefore, this table is crucial for this purpose.

Orders Table:

Field Name	Data Type	Field Size	P/F	Example
OrderID	INT	11	Primary	5
OrderCustomerID	INT	11	Foreign	3
OrderPhoneNumber	CHAR	11		07911123456
OrderCounrty	VARCHAR	20		UK
OrderCity	VARCHAR	50		London
OrderAddress	VARCHAR	100		South Ruislip
OrderPostcode	VARCHAR	12		Ha4 0ls
OrderTrackingNumber	VARCHAR	80		4221736293
Total	Float	-		£29.50
PaymentID	INT	11	Foreign	8

When a customer submits an order, the business must identify some information in order to deliver the order to the customer, therefore, this table stores data and information about the person who ordered the product such as address and phone number etc.

Order items Table:

Field Name	Data Type	Field Size	P/F	Example
ItemIID	INT	11	Primary	3
ItemOrderID	INT	11	Foreign	5
ItemProductID	INT	11	Foreign	4
ProductName	VARCHAR	250		iPhone 12
Price	Float	-		£800
Quantity	INT	11		1

This table stores data about each individual item in an order. This will help the business determine information such as the quantity of each item and product name etc.

Payment Methods Table:

Field Name	Data Type	Field Size	P/F	Example
CardID	INT	11	Primary	2
CustomerID	INT	11	Foreign	3
OrderID	INT	11	Foreign	4
Name	VARCHAR	100		Abdul
Address Line	VARCHAR	100		24 Victoria Road
country	VARCHAR	50		UK
city	VARCHAR	20		City
county	VARCHAR	50		Middlesex
postcode	VARCHAR	12		Ha4 0ls
cc_type	VARCHAR	20		Visa
cc_num	INT	16		4916043720801609

SortCode	INT	3		123
phone	VARCHAR	11		07911123456
cc_expmonth	DateTime	19		12
cc_expyear	DateTime	19		2029

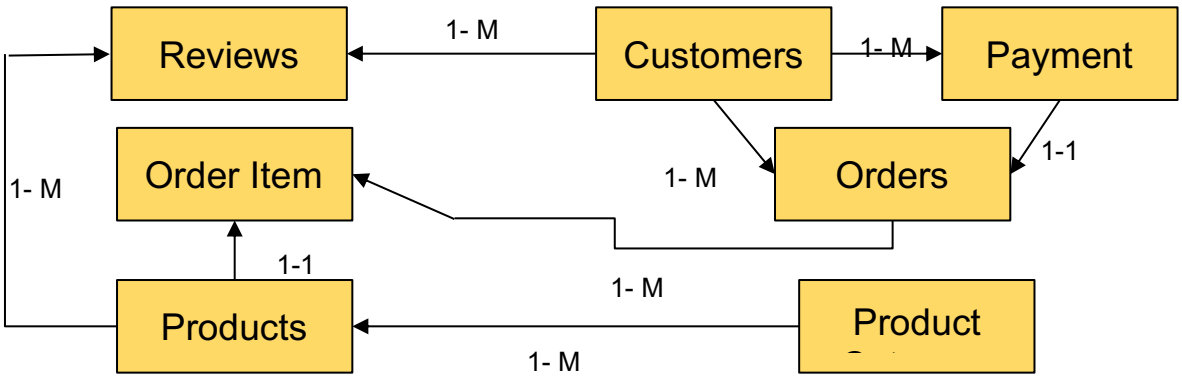
This table is important as it stores very confidential information such as credit card details. This information is crucial to the application because the user needs to enter this data in order to submit an order and then receive their order. Otherwise, they won't be able to pay for the order and therefore they won't receive it.

Reviews Table:

Field Name	Data Type	Field Size	P/F	Example
ReviewID	INT	11	Primary	5
ProductID	INT	11	Foreign	4
CustomerID	INT	11	Foreign	2
CustomerFirstName	VARCHAR	100		Abdul
CustomerLastName	VARCHAR	100		Ibrahim
Rating	Float	-		4.0
Title	VARCHAR	50		Great Product!
Description	VARCHAR	500		I found the product really useful and I recommend it.
DatePosted	DateTime	-		04/05/2021 17:02:12

The reviews table will allow me to store any reviews or comments a customer made on a specific product, and then I can display them under that product. This is important because it will make the customers trust the business more, as I would be displaying honest opinions about a product which will help customers to make their decision.

ERD Diagram:



Validation Table:

Field Name	Validation Checks	Description	Error Message
FirstName, Last Name	Presence Check Type Check	The presence check will make sure that the user has entered something in the data entry box, the type check checks the type of the data entered and makes sure its string and not any other type as names can't have numbers or symbols.	"Invalid First Name" "Invalid Last Name"
Email Address	Presence Check Format Check Database Check	The presence check will make sure that the user has entered something in the data entry box. The format check will make sure that the email entered is in the correct format which is string@string.com or .co.uk etc. The database check verifies if the email is already being used by someone else in the database.	"Invalid Email Address"
Password	Presence Check Length Check Mixed Character check	The presence check will make sure that the user has entered something in the data entry box. The length check makes sure that the password is 8 characters long. A mixed character check makes sure that the password has at least 1 number, 1 uppercase letter, 1 lowercase letter etc.	"Invalid Password"
PhoneNumber	Presence Check Length Check	The presence check will make sure that the user has entered something in the data entry box. The length check makes	"Invalid Phone Number"

		sure that the phone number is 11 digits long.	
Postcode	Presence Check Format Check	The presence check will make sure that the user has entered something in the data entry box. The format check will make sure that the postcode entered is in the correct format which is LLN NLL.	“Invalid Postcode”
Product Price	Presence Check Format Check	The presence check will make sure that the user has entered something in the data entry box. The format check will make sure that the price entered is in the correct format which must contain a currency sign (e.g. £).	“Enter a Price Only ”
Stock, Quantity	Presence Check Type Check	The presence check will make sure that the user has entered something in the data entry box. The type check makes sure that the data entered is an integer as the amount of stock can’t be a string.	“Enter the amount of Stock or the Quantity”
cc_type	Presence Check Type Check	The presence check will make sure that the user has entered something in the data entry box. The type check makes sure that the data entered is a string (e.g. VISA, Master Card, American Express).	“Select the Credit Card Type”
cc_num	Presence Check Length Check	The presence check will make sure that the user has entered something in the data entry box. The length check makes sure that the credit card number is 16 digits long.	“Invalid Credit Card Number”
Cc_exp month/year	Presence Check Type Check	The presence check will make sure that the user has entered something in the data entry box. The type check makes sure that the data entered is a date/time data type.	“Invalid Expiration Date”

The validation table is important because it makes sure that the system provides an excellent user experience. For example, if there was no validation for the credit card number, and the user entered an incorrect credit card number and the system did not alert them, then the user would not be able to order the product at all, which will make the whole website useless as users purchasing products is the main objective.
When developing the project, the validation table will assist me to identify the fields I need to validate, what types of checks I need to include, also what error message should I display to the user.

Object-Oriented Programming Design:

Introduction to OOP & its advantages:

Object-oriented programming is based on the concept of objects. In object-oriented programming data structures or objects are defined, each with its own properties or attributes. Each object can also contain its own procedures or methods. Software is designed by using objects that interact with one another.

Benefits:

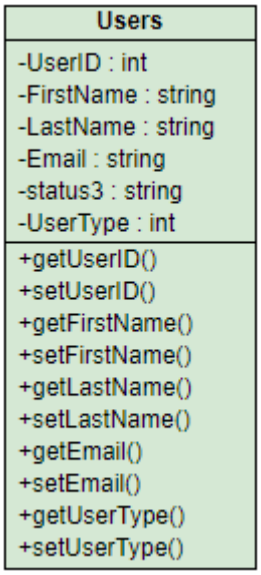
- Reusability: this means that we can reuse the code as many times as we want rather than creating them again and again, therefore, it saves time and increases efficiency.

- Easier troubleshooting: this is because the code is decomposed and broken down into smaller parts and modulus, and so it will help me identify where the problem is much faster as I know where to look for it.

- Higher productivity: as I could reuse the code and I know how to troubleshoot problems faster, I would be able to increase the output and produce a better program.

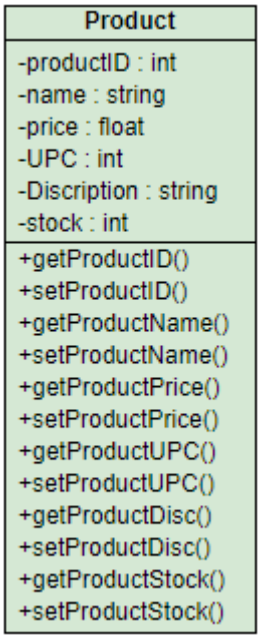
Class diagrams:

Users:



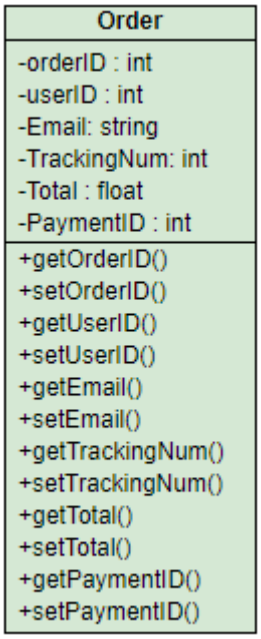
This class stores all the details about a user, this makes it easier when trying to retrieve or update any information about that user anywhere in the code.

Products:



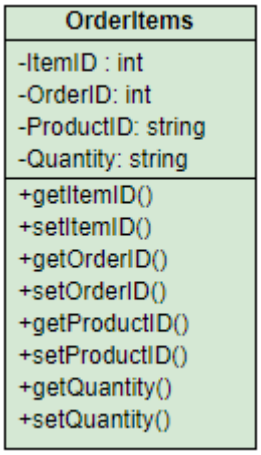
This class stores all the information about a product, this makes it easier when trying to retrieve or update any information about that product in places such as the basket or favourite items section.

Orders:



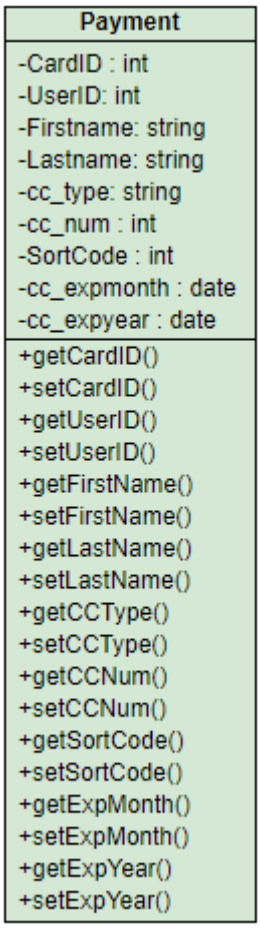
The purpose of this class is to store any order the customer has submitted by instantiating an object, so we can use the information in any place in the code, for example when dealing with the Order Items table that stores every single item in the order submitted by the customer.

Order Items:



This class stores the individual items of an order.

Payment:



This class allows me to store the details of any card used by the user to pay. This is important because it makes it easier when retrieving the payment method to store it in the orders table.

Test Plan:

Iterative Testing
Post-Development Testing

Test No.	Module	Description	Test Type	Expected Outcome	Pass/Fail
1	Register	Inputting the correct name, email and password(in the correct format) into all the inputs.	Normal	The user should be added to the database.	
2	Register	Inputting Incorrect name, email and password(in incorrect format) into all the inputs.	Erroneous	The application should output an error message underneath each input.	
3	Register	Inputting an email that already exists in the Users table in the database.	Extreme	The application should output an error message informing the user that an account with an email already exists.	
4	Login	Inputting the email and password of an existing user.	Normal	The user should be logged in and redirected to the Welcome page.	
5	Login	Inputting email and passwords that do not exist in the Users table in the database.	Extreme	The user should be redirected to the login page and an error message should be printed.	
6	Login	Inputting correct email and an incorrect password.	Erroneous	An error message should appear informing the user that one of the inputs is incorrect.	
7	Browse	Search for a product that exists in the database.	Normal	The users should be redirected to the results page where they can see the products they searched for.	
8	Browse	Search for a product that does not exist in the Products Table in the database.	Erroneous	The users should be redirected to the results page and a message should appear saying no items found.	
9	Browse	Search for a product in a specific category with the product existing in that category.	Normal	The users should be redirected to the results page where they can see the products they searched for.	
10	Browse	Search for a product in a specific category with the product not existing in that category.	Extreme	The users should be redirected to the results page and a message should appear saying no items found.	
11	Browse	Rate and Write a review for a	Normal	The review should be added	

		product.		to the database and it should appear with the rating in the correct position in the product details page.	
12	Browse - Administrator	Reply to a review of a product.	Normal	The reply must be displayed under the review that has been replied to, also it should be added to the database.	
13	Basket	Add a product to the basket that has stock.	Normal	The product should be added and displayed in the basket as well as in the Basket Table in the database.	
14	Basket	Add a product that already exists in the basket.	Extreme	The user should receive a message informing them that the product already exists in the basket.	
15	Basket	Increase the quantity of a product.	Normal	The total price should increase.	
16	Basket	Decrease the quantity of a product.	Normal	The total price should decrease.	
17	Basket	Delete an item from the basket.	Normal	The user should no longer see the item in the basket and the total price should decrease by the price of the deleted item.	
18	Basket	Adding/Deleting items to/from the basket while the user is logged in.	Normal	The basket should be updated and no errors should appear.	
19	Basket	Adding items to the basket while the user is not logged in.	Erroneous	The basket should not be updated and the user should receive an error message informing them that they have to log in.	
20	Basket	Create a new basket.	Normal	The user should see 2 different baskets in the Basket Page	
21	Checkout	Inputting valid credit card number, sort code, dates and billing address.	Normal	The user should be redirected to the confirmation page.	
22	Checkout	Inputting invalid credit card number, sort code, dates and billing address.	Erroneous	An error message should appear to the user stating the invalid inputs.	
23	Checkout	Ticking the box to allow the	Normal	The payment method should	

		system to store the payment method in the database.		be saved and stored in the payment table in the database.	
24	Checkout	Not allowing the system to store the payment method by not ticking the box.	Normal	The payment method should not be saved and stored in the payment table in the database.	
25	Purchase History	Adding items to the basket and checking out with valid credit card numbers, sort code, dates and billing address.	Normal	The purchase history should be updated and the user should be able to see the items he/she purchased.	
26	Administrator - Delete a user	Delete a user that exists in the database by typing the first name and the ID in the search bar	Normal	The user should be deleted from the system.	
27	Administrator - Delete a user	Try and delete a user that doesn't exist in the database by searching the first name and the ID	Erroneous	An error message should appear informing the admin that the user doesn't exist.	
28	Delete a user	Login with the account that got deleted	Erroneous	The user should not be able to log in as the account got deleted from the database.	
29	Stock	Update the amount of stock when the user adds a product to the basket.	Normal	The amount of stock should be decreased based on the quantity the user has purchased.	
30	Admin - Stock	Add stock for a certain product.	Normal	The product table should be updated and stock should increase based on the quantity added by the admin.	
31	Stock	Add an out of stock product to the basket.	Extreme	The users should receive an error message informing them that the product ran out of stock.	
32	Links	Clicking on the basket link in the header.	Normal	The user should be redirected to the basket page.	
33	Links	Clicking on the My Items link in the header.	Normal	The user should be redirected to the My Items page.	
34	Links - Admin	Clicking on the Admin link in the header.	Normal	The administrator should be redirected to the admin page.	
35	Favourite Items	Adding a product to favourite items.	Normal	Favourite items table should be updated, however, the stock of the product should	

				not change.	
36	Favourite Items	Adding an out of stock product to Favourite items.	Normal	The user should receive a message saying the product has run out of stock, but the product still gets added to favourite items.	
37	Favourite Items.	Adding items to favourite items while the user is not logged in.	Erroneous	The user should receive an error message saying they have to either log in or register to add items.	
38	Favourite Items	Remove an item	Normal	The item should get removed from favourite items.	
39	Admin - Edit Items	Edit the name of a product.	Normal	The name of the product should be updated in the products table.	
40	Admin - Edit Items	Increase the price of a product.	Normal	The price of the product should be updated in the products table.	
41	Admin - Edit Items	Set a discount on a product.	Normal	The system must calculate the discount and decrease the price, and the discount percentage should appear next to the price on the product page.	
42	Admin - Edit Items	Edit the description of a product and has to be between 50 - 500 words	Normal	The description of the product should be updated in the products table.	
43	Admin - Edit Items	Edit the description of a product and make it less than 50 words.	Extreme	The user should receive an error message informing them the minimum amount of words is 50.	
44	Admin - Edit Items	Edit the description of a product and make it more than 500 words.	Extreme	The user should receive an error message informing them the maximum amount of words is 500.	
45	Admin - Edit Items	Edit the image of a product.	Normal	The image of the product should be updated in the products table.	
46	Discounts	Set a buy 1 get 1 free offer on a product.	Normal	If the user gets 2 of that product, then he/she must only be charged for one of them and the subtotal must appear in the basket.	

47	Discounts	Use a promotion code that exists in the database.	Normal	The promotion code must be fetched from the database and calculated and the price should be decreased.	
48	Discounts	Use a promotion code that does not exist in the database.	Erroneous	The user must receive an error message saying the code is invalid.	
49	Basket	Add an item to the basket and check the total price.	Normal	The VAT of 20% must be added to the price of the product and the total price should appear at the bottom of the basket.	
50	Admin - Adding items	Add a new item to the store with the correct name and description.	Normal	The product should be added to the database.	
51	Adding items	Search for the new item in the correct category	Normal	The user should be able to see the product on the results page.	
52	Adding items	Search for the new item in the incorrect category	Erroneous	The user should get an error message informing them the product is not found.	
53	Admin - Adding items	Add a new item to the store with an incorrect name and description.	Erroneous	The product should not be added to the database and the admin should receive an error.	
54	Admin - Deleting items	Search for a product that exists in the database and delete it.	Normal	The product should be deleted from the database.	
55	Admin - Deleting items	Search for a product that doesn't exist in the database and delete it.	Erroneous	The admin should get an error saying the product is not found.	
56	Deleting items	Search for the deleted item without specifying the category.	Normal	The user should get a message saying the product is no longer available.	
57	Sales report	Add to the orders table and check the sales report to see the number of orders that day, most purchased product/categories and revenue.	Normal	The number of orders that day, most purchased product/categories and revenue must be updated.	
58	Sales report	Click revenue in detail.	Normal	The admin should be able to see the quantity sold and revenue for each product separately.	

59	Sales report	Go to previous months to see sales reports	Normal	The user should be able to select a month in the last 12 months and see the report	
60	Welcome Page	Test the page in a different browser.	Normal	The page should appear correctly.	
61	Welcome Page	Test the page on a mobile phone	Normal	The page should appear correctly.	
62	Welcome Page	Change the size of the page on a computer.	Normal	The elements of the page should change sizes based on the change in the size of the page.	
63	Links	Login with a user account and go into the admin page by typing the link manually.	Extreme	The user should be redirected to the welcome page.	
64	Links	Without logging in with any account, access the admin page by typing the link manually.	Extreme	The user should be redirected to the index page to log in.	
65	Browse	Search for a product and see the images.	Normal	The user should be able to see all pictures clearly and swap between them.	
66	Browse	Check the alt(alternative) text of an image.	Normal	If the image is not loading then the user should be able to see text describing the image.	
67	Browse	Enlarge the image of a product.	Normal	The user should be able to zoom in/out.	
68	Basket	Add a product to the basket and see the image.	Normal	The user should see one picture of the product with a smaller size.	
69	Browse	Search for a product and filter the results based on price.	Normal	The user should see all the products that are in the price range specified by the user.	
70	Browse	Search for a product and filter the results based on price, and make the range too little.	Erroneous	The user should receive an error message saying that no product is found in that range.	
71	Browse	Search for a product and filter the results based on colours, sizes or brands.	Normal	The user should be able to see all the filtered results.	
72	Browse	Search for a product and filter the results based on colours.	Normal	The user should be able to see all the filtered results.	
73	Browse	Search for a product and filter	Normal	The user should be able to	

		the results based on brands.		see all the filtered results.	
74	Favourite Items	Move an item from favourite items to the basket	Normal	The item should be added to the basket	
75	Sidebar Nav	Expand the sidebar navigation	Normal	The sidebar nav should be animated and expanded.	
76	Sidebar Nav	Close the sidebar navigation	Normal	The sidebar nav should be animated and closed.	
76.1	Sidebar Nav	Expand the sidebar navigation while logged in.	Normal	The sidebar nav should be animated and expanded and it should print the name of the user at the top.	
76.2	Sidebar Nav	Expand the sidebar navigation while not logged in.	Normal	The sidebar nav should be animated and expanded and it should print “Sign in” at the top.	
77	Welcome page	From another page, go to the welcome page by clicking on the logo in the navigation bar.	Normal	The user should be redirected to the welcome page.	
78	Navigation Bar	Hover over “My Account” in the navbar while not logged in.	Normal	A dropdown menu should appear giving the user the option to log in.	
79	Navigation Bar	Hover over “My Account” in the navbar while logged in.	Normal	A dropdown menu should appear giving the user the option to either go to purchase history or to log out.	
80	Navigation	Before logging in, in the index page and in the login form click on the “Click here to register” link.	Normal	The user should be redirected to the registration page to create an account.	