

The Algrech Photo Store

Design Document

Dec 2, 2016

(names removed)

Table of Contents

Introduction	1
Mission Statement	1
Executive Summary	1
System Functions	2
Entity Relationship Model	4
Description of Web Interface	5
Sitemaps	6
System and Feature Description	8
Site Limitations	12

Introduction

The Algrech Photo Store is the seventh most popular camera store exclusive to Canada. In recent years, it's become painfully obvious that their lack of online presence is causing their business to become stagnant. For this reason, we were contacted and asked to build an online store to give people convenient access to information about their products and to allow them to shop effortlessly from home.

This report outlines our plans to fulfill this request. It describes the desired behavior of the website and provides details regarding the design of the underlying database, including an ER diagram and the relational schema.

Mission Statement

Our mission is to build a simple and pleasant online shopping experience for customers of The Algrech Photo Store. Guests to the site must be able to search and browse through all available products, have the ability to order products quickly and easily, and be able to view their order history. Managers must be able to add and edit available products (including inventory levels) as well as get sales statistics and demographics.

Executive Summary

The website is simple and easily accessible, enabling all guests to search and browse through the available products and add them to a shopping cart without needing to register for an account. If a guest wants to purchase the items in their cart, they then need to register for a user account or login to their account if they have already registered. When logged in, users also have the ability to view their order status, as well as look at orders they have placed in the past.

When registering, users are required to supply their name, phone number, email address and shipping address. They also need to set a password to use alongside their email address when logging in. In case a user forgets their password, they are able to reset by having a reset link sent to the email address on their account. Users have the ability edit their address, phone number and password at any time.

Each product on the site is listed with relevant details as well as at least one photo of the product itself.

The website also has a private section, accessible only by store and warehouse managers with special manager accounts. This section of the site enables managers to edit product details and inventory levels, add new products, view past orders, edit current orders, and generate reports with sales records, statistics, and demographics.

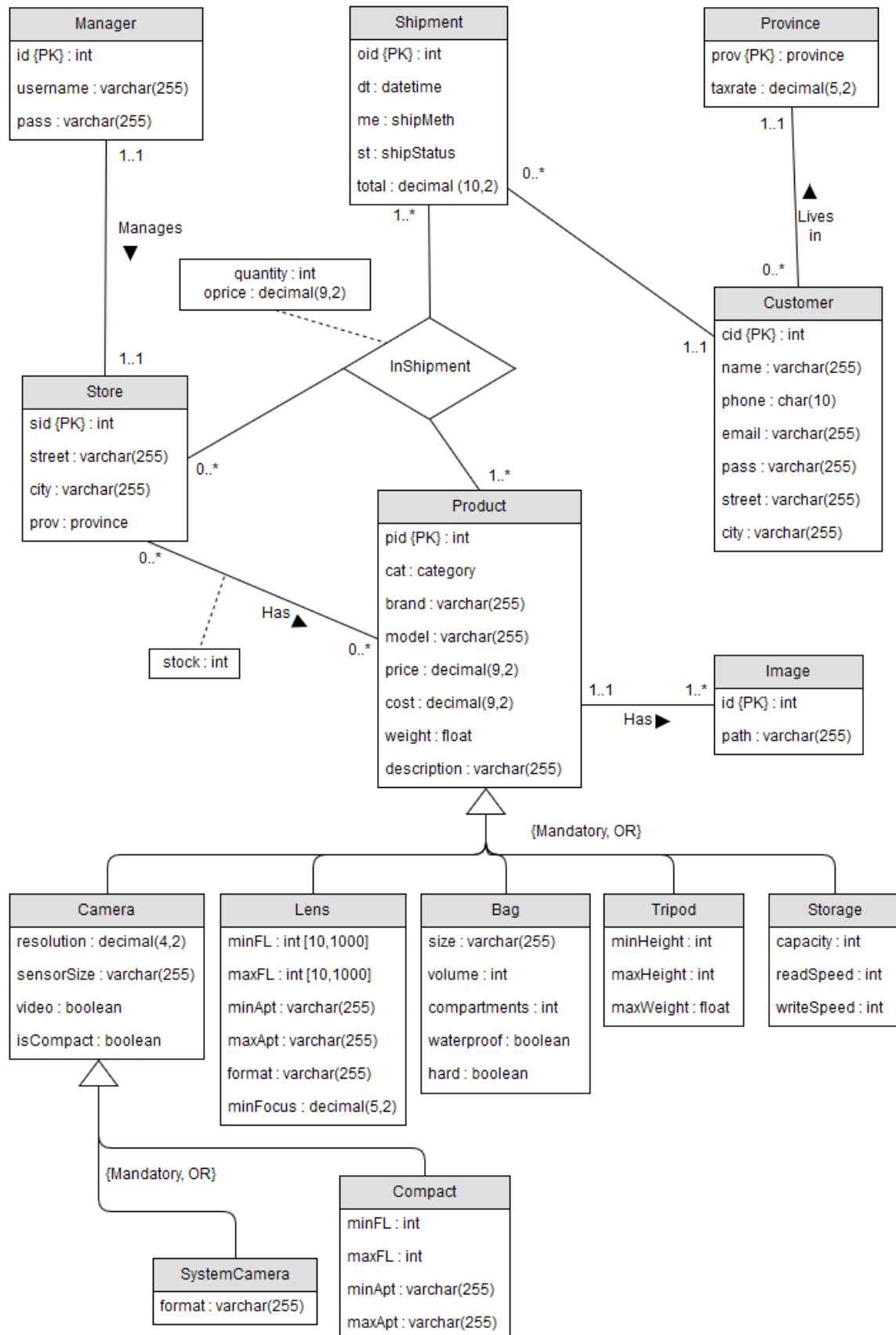
System Functions

DEFINITIONS		
Shall	Something the site/system must have or do at all times.	
Should	Something the site/system could do or contain.	
Must not	Something the site/system must not do.	
User	A general user of the system without credentials.	
Customer	A user with customer credentials and privileges that has created a customer account.	
Manager	A person working for the company that is that either runs the warehouse or a store, has manager credentials.	
Administrator	An person working for the company that is responsible for system and site functionality, security, and maintainence and has administration credentials.	
Critical	Functions that are required to be implemented for the first release of the system and required for operation.	
High	Functionality that will be implemented for the first release of the system.	
Med	Functions which some of may be implemented for the first release of the system with the remainder in the second version.	
Low	Functions which may be implemented in the first release but will be handled in the later versions.	
NUMBER	FUNCTIONAL REQUIREMENTS	PRIORITY
PUBLIC FUNCTIONS		
1	Users shall be able to browse for items for sale without registering.	Critical
2	Users shall be able to search for items for sale by keyword without registering.	Critical
3	Users shall be able to register at the site by providing their name, e-mail, phone number and address.	Critical
4	Users shall be able to add items in their shopping cart while browsing/searching.	Critical
5	Users shall be able to remove an item in their shopping cart while browsing/searching.	Critical
6	Users shall be able to update items in their shopping cart while browsing/searching.	Critical
7	Users shall be able to search for items for sale by category/subcategory	High

CUSTOMER FUNCTIONS		
8	Customers shall be able to login by providing user id and password.	Critical
9	Customers shall be able to checkout and pay for items in their shopping cart.	Critical
10	A Customer that has placed an order shall be able to see order status and shipment information after order has been placed.	Critical
11	A customer shall be able to view their order history that can be accessed at any time.	High
12	A customer shall be able to retrieve their forgotten user id/password using their e-mail address.	High
MANAGER FUNCTIONS		
13	A manager shall be able to generate simple sales reports including frequent items, sales volume, etc.	Critical
14	The warehouse manager shall be able to add an item for sale on the site	Critical
15	A manager shall have the ability to upload a photo and add to and existing product	High
SYSTEM REQUIREMENTS		
16	The system shall enable items to have at least one picture.	Critical
17	The system shall track inventory levels of items in central warehouse.	Critical
18	The system shall only allow item sales given sufficient inventory.	Critical
19	Products should be able to have multiple pictures and extended information.	High
20	The system shall have constraints on the data and check to ensure that it is valid.	High
21	The system shall integrate internet distribution warehouse with store inventory.	Med
22	Calculate shipping cost and taxes based on item weight and customer location.	Med
23	The system shall send e-mail notification on registering, password reset.	Med
24	The system shall allow simulated advanced payment options including PayPal or credit card transactions (These will not be real transactions)	Low

ADMIN FUNCTIONS		
25	The administrator shall be able to recover the system by using a (SQL) script to create tables and load initial data.	Critical
SECURITY		
26	All administrator/manager functions shall be secured by user id and password.	Critical

Entity Relationship Model



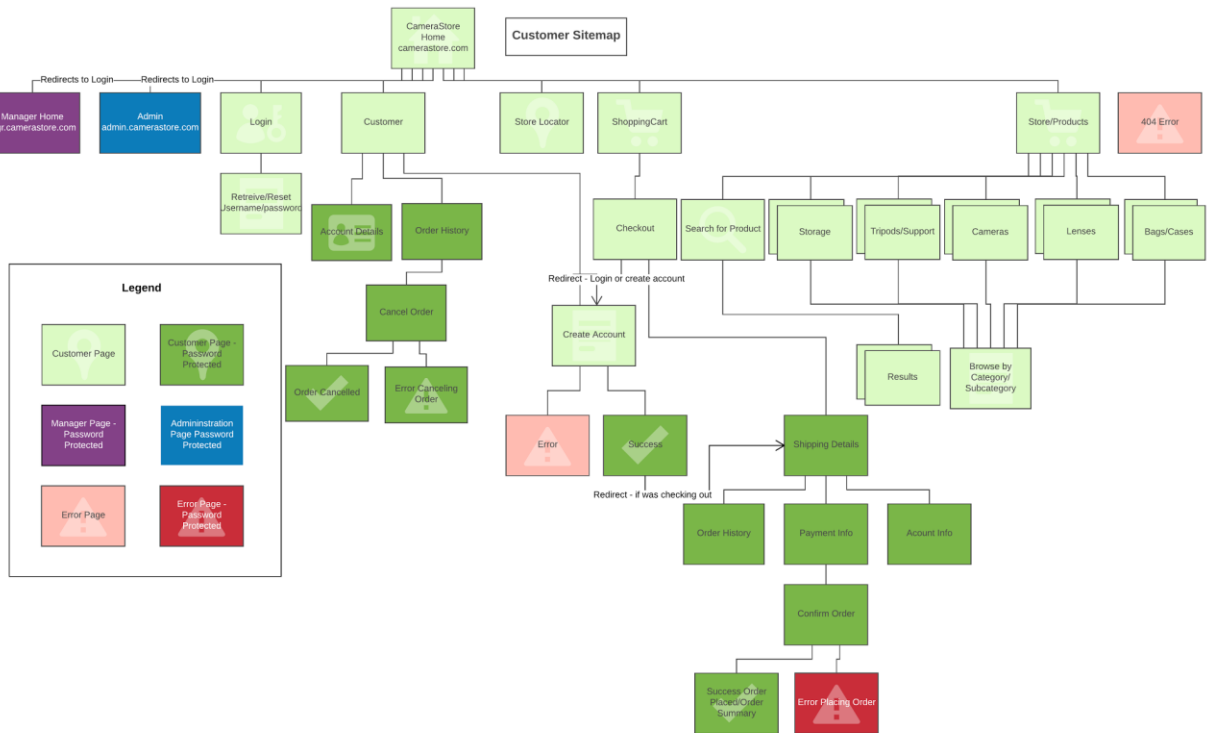
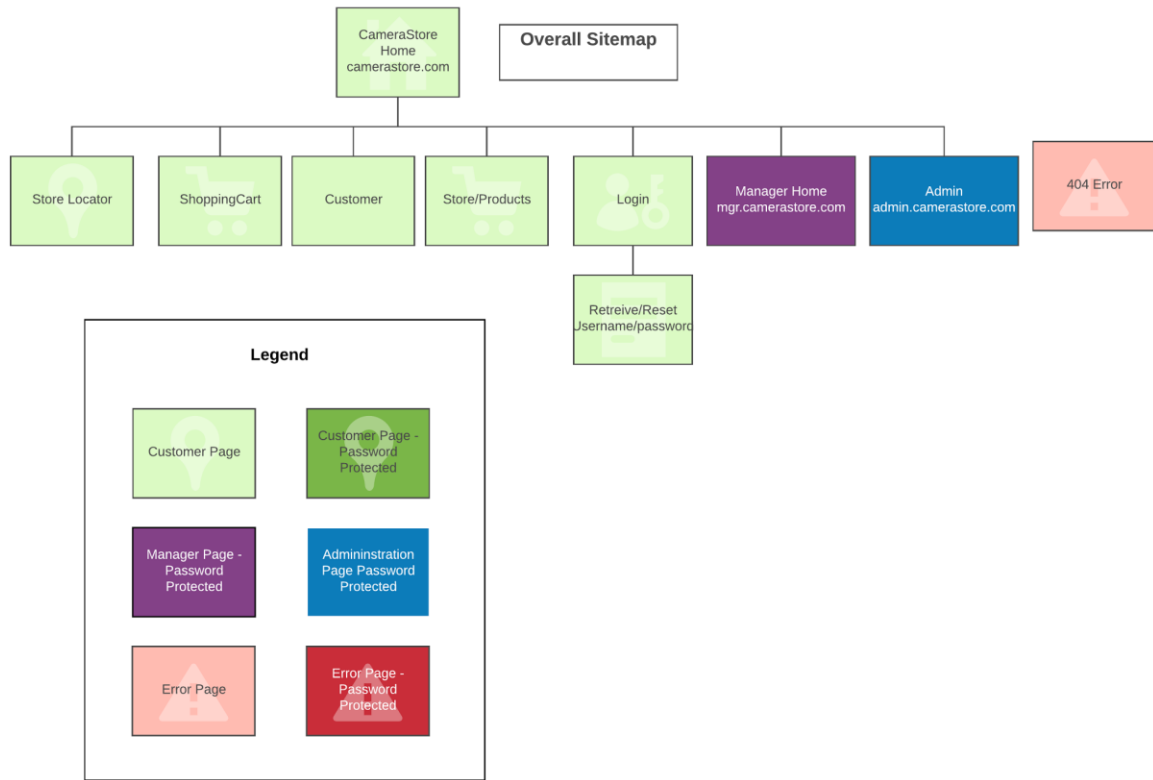
Description of Web Interface

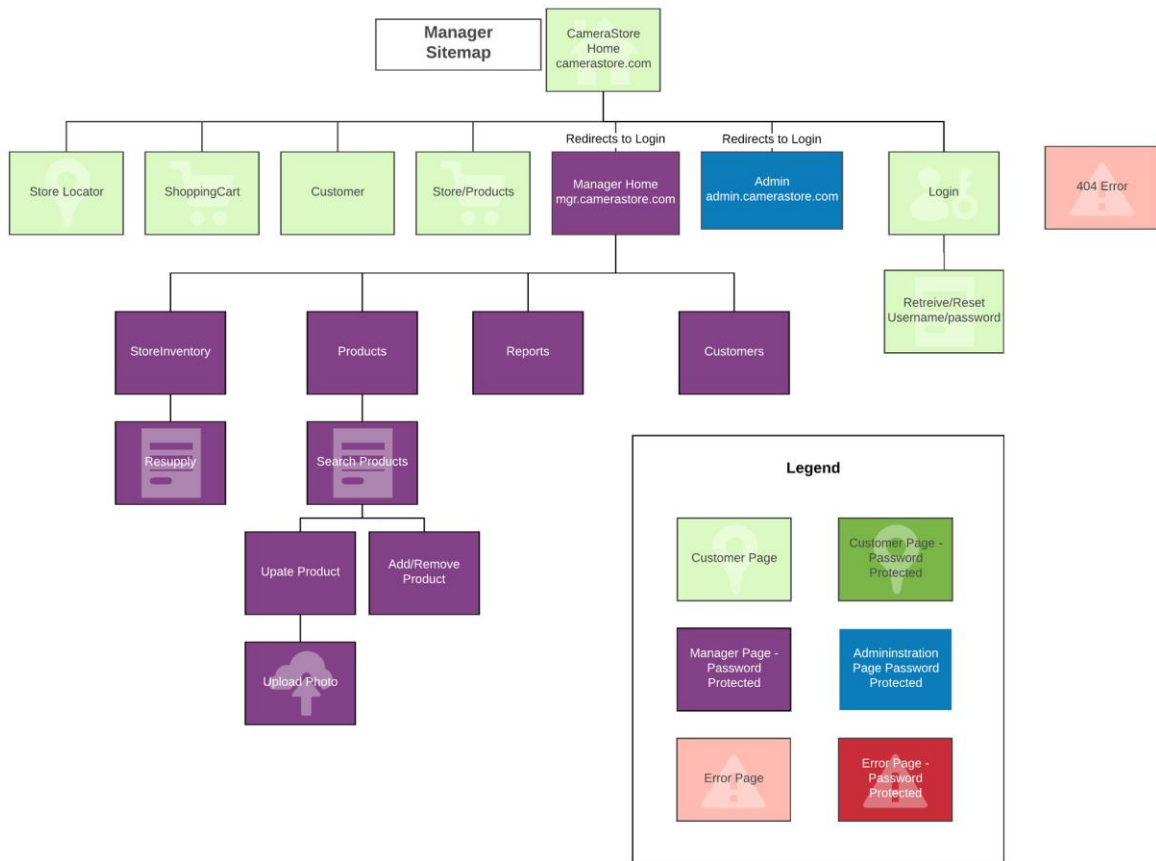
The website for The Algrech Photo Store will have 4 main areas, a public portion of the site devoted to users without any credentials, a customer area of the site, a manager section of the site and an administrator section of the site. Each of these areas will have a simple and clean interface and easy to navigate menus that allows all users to accomplish their goals with the site. The customer, manager, and administrator portions of the site are password protected and only the users with correct privileges will be able to access each area of the site.

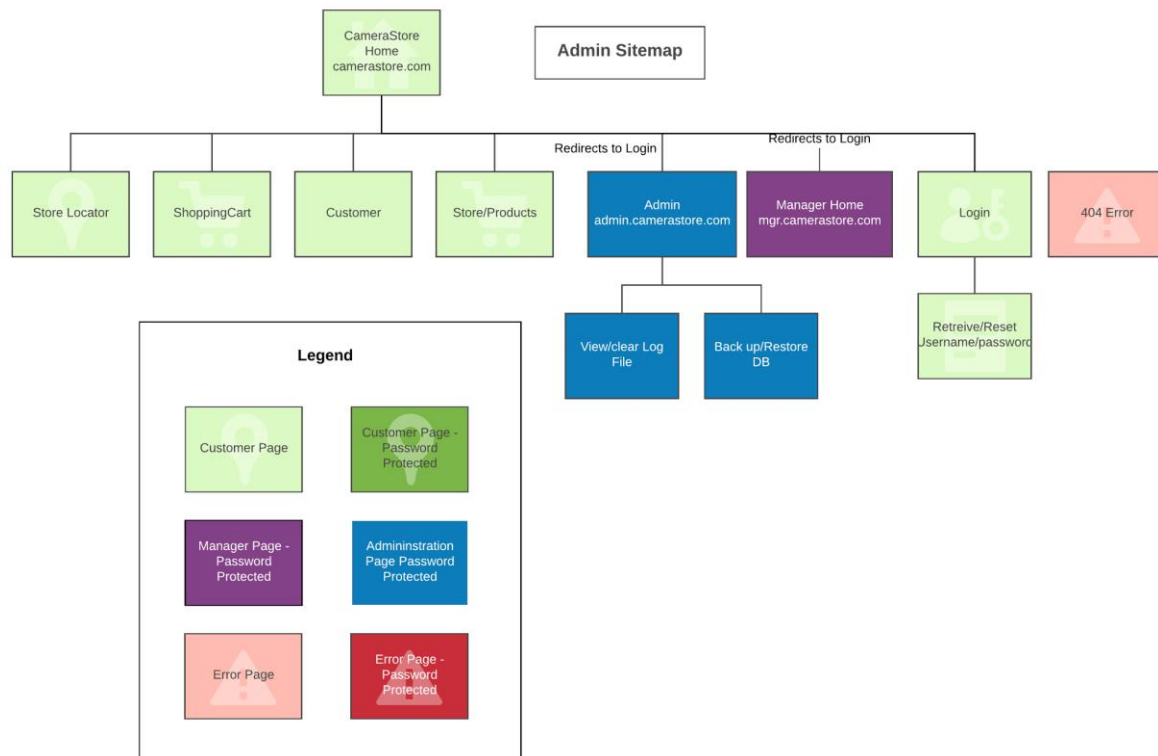
The public area of the website will enable users to browse for products, search for products, add items to shopping cart checkout and create an account. If a user is not logged in when checking out they'll be redirected to a page to either login or create an account and then continue on with the purchase logged in with customer credentials. Customers will be able to view order history and status.

The admin and manager pages will not be linked to off of the main public pages but will be accessible directly by URL (using 'admin.' and 'mgr.' URL prefixes, respectively) and or when logging in with correct credentials. This can be done off the public pages and managers will be redirected to the manager site and administrators will be redirected to admin site. Managers will be able perform store and warehouse functions through the website and administrators can perform administration duties.

Sitemaps







System and Feature Description

Our System uses a thin client three tier architecture with seven components to it, A mariaDB database, apache web server, the mail server that cosc304 uses, our backend code written in PHP, our pages written in a combination of PHP and HTML, using CSS for styling and some java script (using jQuery) for a few nice UI touches. For the CSS we used UIKit and some custom styling. Our core system is built with PHP, uses a mix of object oriented programming and procedural abstraction for the scripts. We abstracted Products, Session, Database, and Cart into classes as well as two static classes User and Mail. Data was passed between pages using the Session class which acted as a wrapper class for the super global `$_SESSION` to make interacting with it easier.. We used the Session to track if users were logged in, they type of user they were, when they logged in and to hold the shopping cart.

index.php: This is the store's homepage. The user can search for products by category using a dropdown or search using keywords or both. They can also reset the search. Each product is displayed depending on the search filter. Each product is a clickable row that directs to the product.php page.

login.php: The user can choose to enter already existing login information and login or create a new account. They can also reset their password or choose to go through the "forgot your password?" process. Once logged in they are sent to the relevant page. If they were checking out with their cart, they would be sent to viewcart.php.

The session recognizes that a user is not logged in. Managers also login from this page with relevant credentials.

logout.php: The user is redirected back to the index page after logging out. The session reflects that one is not logged in.

product.php: All available information for a product is displayed as well as every picture. The user can choose to “search again” (back to index.php) or “add to cart” (directed to viewcart.php). Relevant product information is retrieved from the database. To get the image(s), an associated file name(s) for the product is taken from the database then dynamically appended to the path of the file from the document root and .

viewcart.php: When an item is added to a user’s cart, they are automatically directed to this page. The page has a table describing each product in the cart and its quantity. Product info is retrieved from the database based on the session’s cart object. A subtotal is displayed at the bottom of the table. A user can update the quantity and remove items entirely. The user can click “continue shopping” or “proceed to checkout”. If the user clicks “proceed to checkout” and is not logged in they will be directed to login.php or register.php. Once they are logged into their account they proceed to the checkout automatically.

checkout1.php: After a user is satisfied with what is in their cart they can proceed to this page. The user’s current shipping details are displayed as retrieved from the database using the user’s id. The user selects their shipping method. They have the option to pick up at any store or ship standard or express. A flat rate is set for standard or express shipping based only on product weight. A user can proceed to pay for their items from here. The shipping type is sent to checkout2.php.

checkout2.php: After a user selects their shipping preference, they see a page with basic order details. The number of unique items, subtotal, shipping costs, sales tax (based on their residing province) and their final total. These have all been stored as session variables at this stage. If they are satisfied with this they can fill out their credit card details in a form. The user can proceed to pay for their items from here.

theprocess.php: After the user has confirmed their payment summary information for their order is displayed which is inserted into the Shipment table. A summary of items purchased is also displayed. This information is inserted into the InShipment table for the associated order number and store number.

orderhistory.php: A user sees all of their previous orders on this page. Included is all information about the order and some summary product information for each order.

admin/dbrestore.php: An admin can click the link available and restore the original database. This version of the database does not include new information added to the database for inventory and order history.

includes/constants.php: This file contains database connection credentials and document roots. It was created so that database connection credentials did not have to be typed out every time a connection was made. Since we all used different document roots on local machines, this streamlined checking out the project from version control.

includes/init.php: This is an initialization file. It was mainly used to include relevant class files and the constant files. Every page needed those files. This allows us to not type out each included and required file.

includes/class/Cart.php: This file is for the Cart class. The session has a cart object with an array containing each product id and quantity for each product in the cart. All of the relevant functionality is in this class. It has a removal function and quantities can be updated.

includes/class/Database.php: This file is for the Database class. Constructing a Database object connects to the database. It can be used to execute queries.

includes/class/Mail.php: This file contains the Mail class. It has functions for sending relevant email for registering and recovering user passwords.

includes/class/Product.php: This file contains the Product class and its subclasses. A product can be constructed as well as any subproduct (based on categories). Product and each subproduct has getter methods for each fields

includes/class/Session.php: This file contains a session class most of the commonly stored classes and variables in the session. The Database, Cart, and User class information can be retrieved easily using this.

includes/class/User.php: This file contains the User class. It has functions for adding a new user, and retrieving user information.

includes/functions/functions.php: This file contains a miscellaneous set of functions that had to be used at various stages of designing the site. Many form validation functions are contained within as well as functions for formatting values.

public/about.php: This file contains basic information about the people involved with the development of the site.

public/addtocart.php: This file is used specifically to add items to a cart and redirect to the viewcart.php page. It is not viewable from the user's perspective.

public/forgotpassword.php: If a user forgets their password they have the ability to reset it. This page allows them to enter the relevant details to get a link for resetting their password.

public/register.php: This page contains a form for entering relevant information for making an account with the site. The registration asks for the user's shipping details as well as login information. Form validation done for each field using PHP once a user submits the form.

public/resetlink.php: This file is not viewable by the user. It sends to the user's email a unique link to reset their password.

manager/index.php: This file is where the user is redirected after logging in with a manager account. It presents the user with links to the various extra tasks that customers and visitors do not have, including the ability to view a sales report on the store run by that manager, add a new product, upload a photo for a product, and restoring the database to a known-functional state.

manager/addproduct.php: This file displays a form for a manager to complete in order to add a new product to the catalogue. Upon completion, the manager is then taken to addtoproduct.php in which they will see another category specific form with which they can add category specific details about the product. That file then sends the form through addproductdetails.php to be processed, in which the product and its details are added to the database.

manager/uploadPhoto.php: This file allows a manager to add a photo to any product currently found on the database.

manager/reports.php: This file shows a report of the total sales from the store run by the currently logged in manager.

Features

User Authentication

Passwords are stored in the databases and authentication of users uses one-way encryption

Send Mail on Events (register, reset password, password reset)

This feature uses the mail() function of php which utilizes the sendmail program of UNIX/Linux. Functions were written to send varying messages depending on the need which called a general function to interact with mail().

Forgot Password

This feature is implemented by sending links to customer who have registered with the site. They entering their email address they registered with (which is checked to ensure it is in the database) then a string is built composed of the time() and the time + 3 hrs along with the user id. This is encrypted and sent as a GET variable in a link to the users email address. Upon clicking the link the page that processes the GET request decrypts the variable, extracts each part, compares the current time() with the expiration time set when the reset link was sent. There is also a check that the two times set in the original link string are $time2 = time1 + 3 \text{ hours}$. If these are not equal then the link was modified before being decrypted and it is invalid. This is not the most secure way of doing the password reset link but we did not want to alter the database at all when we decided to implement this feature and a work around was found that works for demonstration purposes but not actual use.

Restore Database

The database restore currently uses a mysqli function to execute multiple SQL commands in a sql file. It drops all tables if they exist and creates them and re-inserts the original data. This had originally used drop database and create database statements but our accounts did not have the privileges to do so on the cosc server

Upload Photo

This used a form to upload a photo for an existing product and php was used to move the file from the tmp upload directory to the correct directory for the website, and the file name was added to the Image tables in the database.

Site Limitations

- You cannot create a manager account on the site.
- The site does not actually collect payment from a customer, nor will placed orders ever be fulfilled.