

PhotoArt

Showcase the beauty of our world

University of British Columbia Okanagan
COSC 304, Introduction to Databases

Team Information

Rohan Chauhan - Rohan.Chauhan@hotmail.com

Matt Carrier - Matt.Carrier183@gmail.com

Jan Reisler - agentmikster44@gmail.com

Tyler Kreway - TylerKreway@gmail.com

Jorden Roberts - jordenroberts@outlook.com

Contents

1. Introduction

- 1.1 Purpose
- 1.2 Mission Statement
- 1.3 Executive Summary

2. Implemented Features

- 2.1 Documentation Features

3. Domain Assumptions

- 3.1 Warehouse
- 3.2 Product
- 3.3 Stored at
- 3.4 Order
- 3.5 Members
- 3.6 Payment info
- 3.7 Contains
- 3.8 Shipment

4. Data Model

- 4.1 UML Diagram
- 4.2 Entity relations
 - 4.2.1 Warehouse
 - 4.2.2 Product
 - 4.2.3 Order
 - 4.2.4 Members
 - 4.2.5 PaymentInfo
 - 4.2.6 Shipment
 - 4.2.7 Contains
 - 4.2.8 StoredAt
 - 4.2.9 Admin
 - 4.2.10 Moderator
- 4.3 Entity Assumptions
 - 4.3.1 Product Being Stored at Warehouse
 - 4.3.2 Order Contains Product
 - 4.3.3 Order Ships Shipment
 - 4.3.4 Member Places Order
 - 4.3.5 Member Has Payment Info
- 4.4 Relational Schema (SQL DDL)

5. Interface

- 5.1 Visual Layouts
- 5.2 Site Maps

6. User/Admin Walkthrough

7. Known Issues

- 7.1 Incomplete Features
- 7.2 Bugs/Errors

8. Conclusion

Introduction

1.1 Purpose

This project design document describes the design and schema of the PhotoArt Website. Additionally, the document includes the UML design of our database, related assumptions, relational schema as well as many other important features.

1.2 Mission Statement

The crew here at PhotoArt seeks to provide the creatively inclined with exclusive prints of stunning photographs from our professional team of photographers.

1.3 Executive Summary

With so many stock photo websites and near unlimited access to pictures readily available on the internet, the true beauty of photography is being lost. PhotoArt still believes in the importance of photography, which is why we employ a team of highly-experienced professional photographers to deliver our customers with spectacular images.

The simple description of our business is that we sell and ship high-quality images printed on different sizes of premium canvases for reasonable prices. However, PhotoArt knows the power of a meaningful photo, which is why we don't consider ourselves just a photo-distribution service, rather a service that provides a window to your innermost dreams and desires. Fill your home, workspace or office with these beautiful escapes from your everyday life.

Art is borderless, which is why a central goal of our company is to provide a diverse view of the world all around us. For this reason, our photographers are posted all around the world, from the golden dunes of the Sahara to the snow-covered peaks of the Rocky Mountains.

The website, itself, delivers users with a seamless interface for all their print needs. It's easy to find the perfect image as our photos are divided into several categories, including landscape, nature and abstract. If customers have a specific photo in mind, they can find it with the search bar. The user also has the option of browsing our site as a guest or creating an account to review pictures or make purchases (by adding a photo to their cart). Each photo also includes image information so customers know exactly what they're getting and how the photo came to fruition. PhotoArt thinks of its customers as a community, which is why account holders can review any photo. We aim to showcase the beauty of our world with pictures. I'm sure you'll agree, the price tag is well worth the photograph quality and customer experience.

2.0 Implemented Features

2.1 System Requirements

Final Overall System Features	Progress	Code Files
Hosting		
Running on cosc304 server	100%	connection.php
Main/Home Page		
Search for a product by name	100%	PhotoArtMain.php
Browse products by category	100%	PhotoArtExplore.php
List products (by search/browse)	100%	PhotoArtSearch.php
List products with image	100%	Multiple pages including above 2
Page header with menu	100%	<i>All pages</i>
Page header shows current logged in user	100%	<i>All pages</i>
Dynamic products on page based on sales (hot item tracking)	100%	PhotoArtTopPhotos.php
Shopping Cart		
Add to shopping cart	100%	PhotoArtPictureInfo.php and addToCart.php
View shopping cart	100%	PhotoArtCart.php
Update quantity (with data validation)	100%	PhotoArtCart.php, getPrice.php, getPriceTotal.php, PhotoArtPictureInfo.php (note: this worked locally with the online database, not sure if it works on the server)
Remove item from shopping cart	100%	PhotoArtCart.php and removeItemFromCart.php
Improved formatting/UI (e.g. in header bar)	100%	PhotoArtCart.php
Cart stored in database between sessions	100%	Logout.php and PhotoArtLogin.php

Checkout		
Checkout with customer id	100%	PhotoArtCheckout.php
Checkout with payment/shipment info	100%	<i>Same as above</i>
Checkout with data validation	100%	<i>Same as above</i>
Inventory tracking per item	100%	<i>Same as above</i>

Calculate taxes/shipping (by items/stat	100%	PhotoArtCheckout.php
Product Detail Page		
Product detail page and item description	100%	PhotoArtPictureInfo.php
Product detail has an image	100%	PhotoArtPictureInfo.php
User Accounts and Login		
Create user account page	100%	
Create account with data validation	100%	
Login/logout	100%	
Register user/Forgot password email	50%	Forgot pass not done

Product Reviews		
Ability to enter a review on a product	100%	
Display product review on product detail	100%	
Restrict to one review per user on item purchased	50%	Still need to check if user purchased product, PhotoArtPaymentInfo.php
Write Review form is dynamically collapsible (collapsible item section)	100%	
Administrator Portal		
Secured by login	100%	adminPortal.php
List all customers	100%	ListAllCustomer.php
List report showing total sales/orders	100%	salesReport.php
Report with a graph	100%	<i>Same as above</i>
Add new product	100%	alterProduct.php
Update/delete product	100%	<i>Same as above</i>
Upload a photo to database for user	100%	PhotoArtEditProfile.php
Add/update warehouse, customer	50%	ListAllCustomer.php
Database System/General		
Implement some validation using triggers	100%	
Use AJAX for some pages	100%	PhotoArtPictureInfo.php, PhotoArtPaymentInfo.php

User interface and navigation/usability	100%	<i>Every page</i>
Additional Features		
Accessibility for site	100%	<i>Every page</i>
Styling flourishes	100%	<i>Multiple pages</i>
Search for user by name/email	100%	listAllCustomer.php
Enable/disable users	100%	<i>Same as above</i>
Asynchronous update for cart	100%	PhotoArtCart.php

3.0 Domain Assumptions

3.1 Warehouse

- Each one of our Warehouses will be uniquely identified by a name that can not include special characters.
- Warehouse's will be located in cities that are inside of canada..
- They will belong to a province inside of canada.

3.2 Product

- Each product will be identified by a unique UPC that may not be set to null.
- Products will have a price that will not be null.
- Products will have a link to the images available to purchase as a digital copy.
- There will be a description of each product.

3.3 Stored at

- *All products will be stored at a warehouse*
- *These products will be uniquely identified by the UPC and warehouse name*

3.4 Order

- All orders will have an order id associated with them.

3.5 Members

- All members are required to register a username.
- They will create a password that may not be null.
- They must also submit a first and last name that may not contain special characters.
- Members are required to enter a valid email address.

3.6 Payment info

- All payments will be identified by a username and card Number.
- There will be a first and last name for all payments.
- Card numbers will have an expiry date as well as a three digit CSV.
- Payment info will include a valid address, province and city.
- All customers will be shipping to inside canada.
- Postal code will have 6 slots alternating between letter and number.

3.7 Contains

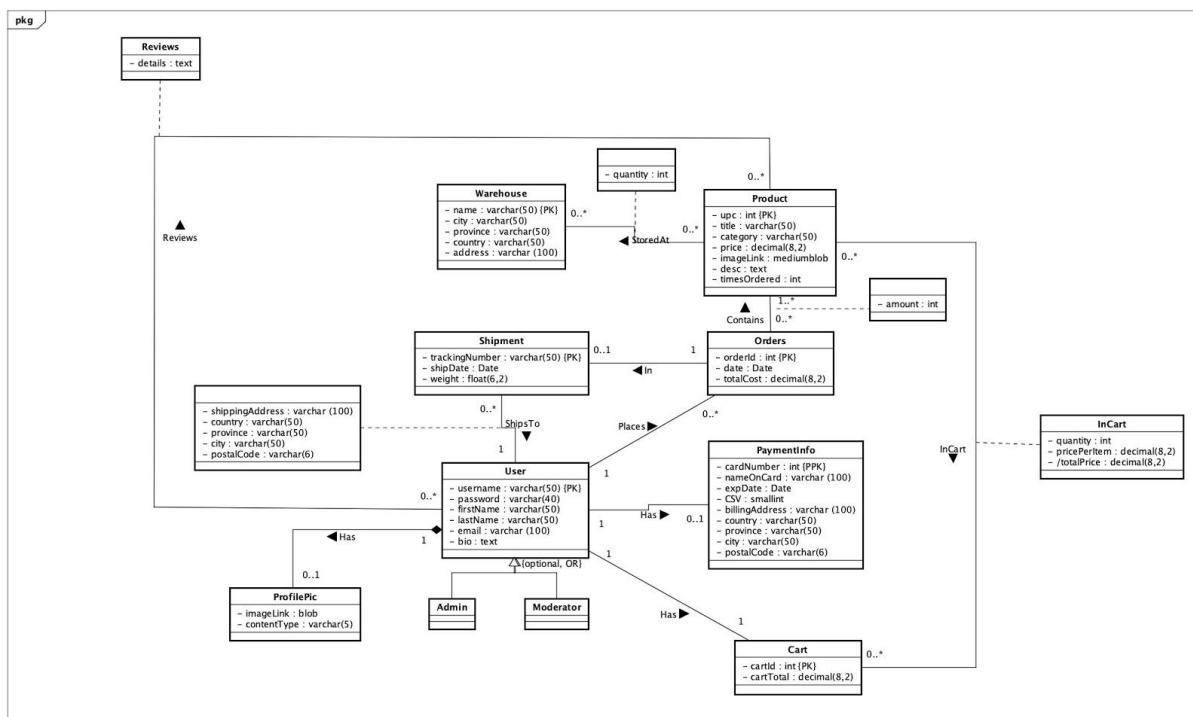
- All orders will contain a product
- The items that an order contains will be identified by both a unique orderID and a UPC too.
-

3.8 Shipment

- Each Shipment will have a unique tracking number
- There will be a shipment date associated with each shipment.
- Shipment's will be weighed and represented as a floating number.

4.0 Data Model

4.1 UML diagram



4.2 Entity Relations

4.2.1 Warehouse

Attribute	Description
<u>name</u> : varchar(50) {PK}	Unique warehouse name
city : varchar(50)	City in which the warehouse is located
province : varchar(50)	Province in which the warehouse is located
country : varchar(50)	Country in which the warehouse is located
address : varchar(100)	Address of the warehouse

4.2.2 Product

Attribute	Description
<u>upc</u> : int {PK}	Universal Product Code used to uniquely identify each product in the database
price : decimal(8,2)	Price of product
imageLink : varchar(1000)	URL leading to preview of image
description : text	Brief description of the image

4.2.3 Order

Attribute	Description
<u>orderId</u> : int {PK, auto}	Unique identification number assigned to each order processed
orderDate : Date	Date on which order was received
totalCost : decimal(8,2)	Total price of order to be charged to customer
username : varchar(50)	Username of member who placed order

4.2.4 Member

Attribute	Description
<u>username</u> : varchar(50) {PK}	Username unique to each member
password : varchar(40)	User's password required for log-in
firstName : varchar(50)	Member's first name
lastName : varchar(50)	Member's last name
email : varchar(100)	Email address associated with member's account

4.2.5 PaymentInfo

Attribute	Description
<u>cardNum</u> : int {PK}	Credit card number used by a member to make purchases
<u>username</u> : varchar {PK}	Username to whom this payment information belongs
nameOnCard : varchar(100)	First name, middle initial, and last name as displayed on credit card
expDate : Date	Expiry date
CSV : smallint	3-digit security code located on back of credit card
billingAddress : varchar(100)	Billing address for transactions made with a specific credit card
country : varchar(50)	Country in reference to billing address
province : varchar(50)	Province in reference to billing address
city : varchar(50)	City in reference to billing address
postalCode : varchar(6)	Postal code in reference to billing address

4.2.6 Shipment

Attribute	Description
<u>trackingNum</u> : varchar(50) {PK}	Tracking number assigned to shipment, giving user access to status updates on their order
shipDate : Date	Date on which shipment left the warehouse
weight : float(6,2)	Gross weight of shipment in kilograms (Note that the weight can't exceed 9,999.99 kg)
shippingAddress : varchar(100)	Address of shipment destination
country : varchar(50)	Country that shipment is sent to
province : varchar(50)	Province that shipment is being sent to
city : varchar(50)	City that shipment is being sent to
postalCode : varchar(6)	Postal code of shipment destination address
username : varchar(50)	Username of member to whom the shipment is being sent
warehouse : varchar(50)	Name of warehouse that filled the shipment
orderId : int	Refers to the order ID that called for the shipment to be made

4.2.7 Contains

Attribute	Description
<u>orderId</u> : int {PK}	Unique order ID used to identify any order placed
<u>upc</u> : int {PK}	Unique product code used to identify any product
amount : int	Amount of the specified product in the specified order

4.2.8 StoredAt

Attribute	Description
-----------	-------------

<u>warehouse</u> : varchar(50) {PK}	Unique warehouse name
<u>upc</u> : int {PK}	Unique product identification code
quantity : int	Amount of the specified product currently being stored at the specified warehouse

4.2.9 Admin

Attribute	Description
<u>username</u> : varchar(50)	Username that uniquely identifies an admin account

4.2.10 Moderator

Attribute	Description
<u>username</u> : varchar(50)	Username that uniquely identifies a moderator account

4.3 Entity Assumptions

4.3.1 Product being Stored at Warehouse

All products are stored in a warehouse but have the potential to be out of stock. A warehouse has the potential to store multiple different products or they may not have that product at all.

4.3.2 Order contains Product

All Orders must contain at least one product. However, not all products must be in a order.

4.3.3 Order ships Shipment

Every shipment will come from a its corresponding order. All orders may be canceled or they may be shipped.

4.3.4 Member places order

All Orders belong to one member but a single member may make multiple orders.

4.3.5 Member has Payment info

All Members have the option of not providing payment info but may only provide one payment method. Payment info belongs to a specific Member.

4.4 Relational Schema: SQL DDL

```
drop table if exists ProfilePic;
drop table if exists Admin;
drop table if exists Contains;
drop table if exists InCart;
drop table if exists Moderator;
drop table if exists PaymentInfo;
drop table if exists Review;
drop table if exists Shipment;
drop table if exists StoredAt;
drop table if exists Cart;
drop table if exists Orders;
drop table if exists Product;
drop table if exists User;
drop table if exists Warehouse;

create table Warehouse (
  name          varchar(50),
  address       varchar(100),
  city          varchar(50) not null,
  province      char(2),
  country       char(6),
  primary key (name)
);

create table Product (
  upc           int,
  title         varchar(50) not null,
  category      varchar(50),
  price         DECIMAL(8,2) not null,
  imageLink     mediumblob not null,
  description   text,
  timesOrdered int default 0,
  primary key (upc)
);

create table User (
  username      varchar(50),
  password      varchar(40) not null,
  firstName     varchar(50) not null,
  lastName      varchar(50) not null,
  email         varchar(100) not null,
  bio           text,
  primary key (username)
);

create table Cart (
  cartId        int auto_increment,
  username      varchar(50) not null,
  cartTotal     decimal(8,2) default 0.00,
  primary key (cartId),
  foreign key (username) references User (username)
    on delete cascade on update cascade
);

create table InCart (
  cartId        int,
  upc           int, -- price per item
  price         DECIMAL(8,2) not null,
  quantity      int,
  primary key (cartId, upc),
  foreign key (cartId) references Cart(cartId)
    on delete cascade on update cascade,
  foreign key (upc) references Product(upc)
    on delete cascade on update cascade
);

-- Add calculated field to InCart to store totalPrice of a given product in cart
alter table InCart add column totalPrice decimal(8,2) as (price*quantity);

create table Review (
  username      varchar(50),
  upc           int,
  reviewDate   date,
  details      text,
  primary key (username, upc),
  foreign key (username) references User (username)
    on delete cascade on update cascade,
  foreign key (upc) references Product (upc)
    on delete cascade on update cascade
);
```

```

create table Shipment (
    trackingNum    varchar(50),
    shipDate      Date,
    weight        float(6,2),
    shippingAddress varchar(100) not null,
    country       varchar(50),
    province      varchar(50),
    city          varchar(50),
    postalCode    varchar(50) not null,
    username      varchar(50) not null,
    warehouse     varchar(50),
    orderId       int not null,
    primary key (trackingNum),
    foreign key (username) references User (username)
        on delete no action on update cascade,
    foreign key (warehouse) references Warehouse (name)
        on delete no action on update cascade,
    foreign key (orderId) references Orders (orderId)
        on delete no action on update no action
);

create table StoredAt (
    upc          int,
    warehouse     varchar(50),
    quantity      int not null,
    primary key (upc, warehouse),
    foreign key (upc) references Product (upc)
        on delete cascade on update cascade,
    foreign key (warehouse) references Warehouse (name)
        on delete cascade on update cascade
);

create table Admin (
    username      varchar(50),
    primary key (username),
    foreign key (username) references User (username)
        on delete cascade on update cascade
);

create table Moderator (
    username      varchar(50),
    primary key (username),
    foreign key (username) references User (username)
        on delete cascade on update cascade
);

create table Orders (
    orderId       int auto_increment,
    orderDate     date,
    totalPrice    decimal(8,2),
    username      varchar(50) not null,
    primary key (orderId),
    foreign key (username) references User (username)
        on delete cascade on update cascade
);

create table Contains(
    orderId int,
    upc      int,
    amount   int,
    primary key (orderId, upc),
    foreign key (orderId) references Orders (orderId)
        on delete cascade on update cascade,
    foreign key (upc) references Product (upc)
        on delete cascade on update cascade
);

create table PaymentInfo (
    cardNum      int,
    username      varchar(50),
    nameOnCard   varchar(100) not null,
    expDate      Date not null,
    CSV          smallint not null,
    billingAddress varchar(100),
    country       varchar(50),
    province      varchar(50),
    city          varchar(50),
    postalCode    varchar(6),
    primary key (cardNum, username),
    foreign key (username) references User (username)
        on delete cascade on update cascade
);

create table ProfilePic (
    username      varchar(50),
    imageLink     blob,
    contentType   varchar(5),
    primary key (username),
    foreign key (username) references User (username) on delete cascade on update cascade
);

-- DB automatically contains one Admin
insert into User values ('Admin', '25d55ad283aa400af464c76d713c07ad', 'Admin', 'Cosc', 'cosc304@ubc.ca', nu
insert into Admin values ('Admin');
insert into ProfilePic (username) values ('Admin');

--Products cannot be deleted
CREATE TRIGGER delProd
    before delete on Product for each row
        signal sqlstate '45000' SET MESSAGE_TEXT = "Products cannot be removed from the database";

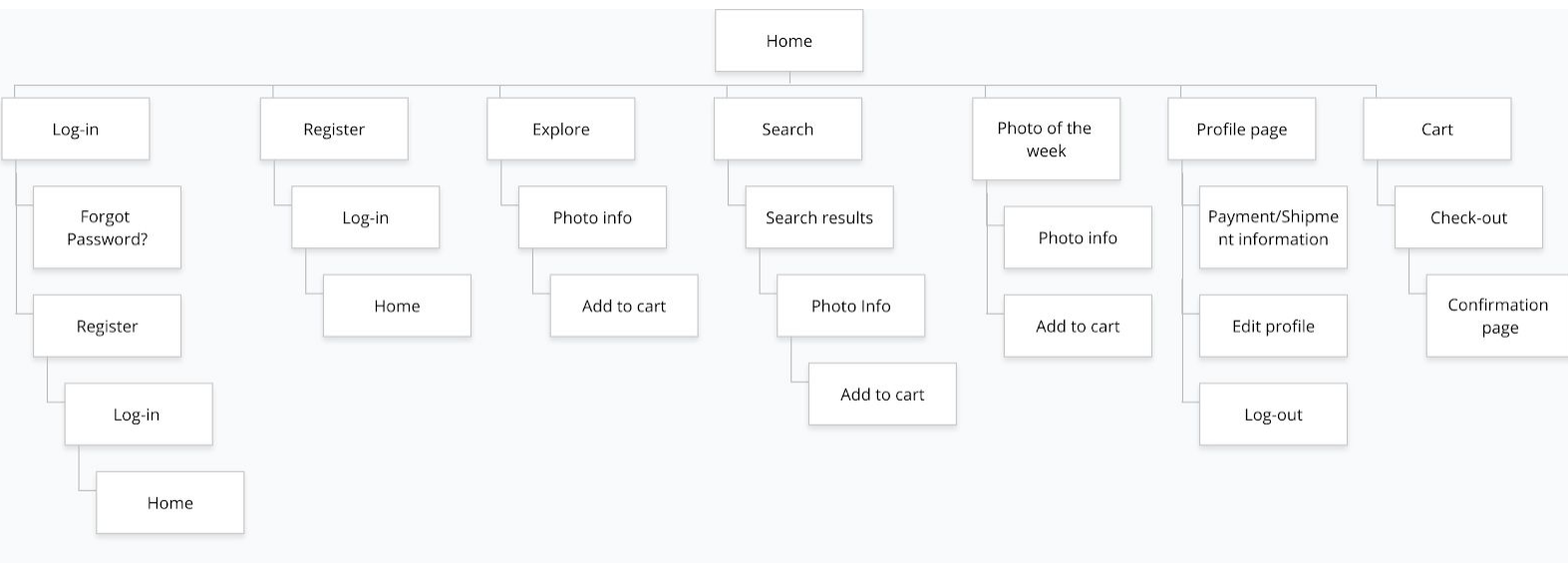
-- Update timesOrdered attribute of product each time it is purchased
CREATE TRIGGER prodPurchased
    after insert on Contains for each row
        update Product set timesOrdered = timesOrdered + new.amount
        where upc = new.upc;

-- Automatically add a new User's username to ProfilePic table
CREATE TRIGGER newUser
    after insert on User for each row
        insert into ProfilePic(username) values (new.username);

```

5.0 Organization of Pages

SITE MAP

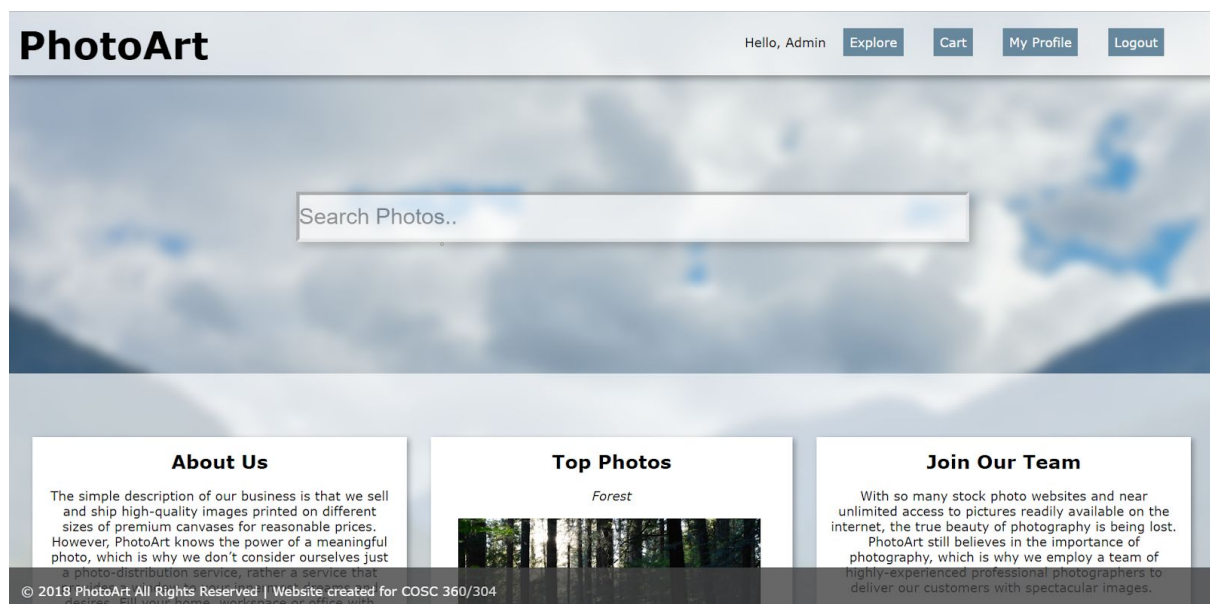


6.0 User/Admin Walkthrough

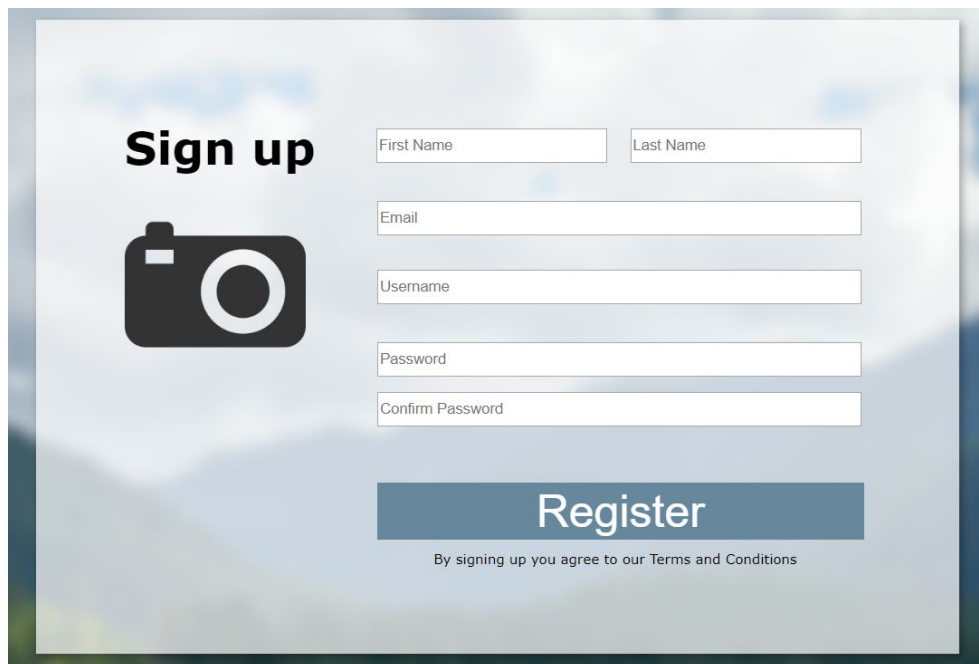
The walkthrough begins by entering the following URL into your browser,

http://cosc304.ok.ubc.ca/group12/src/client_side/PhotoArtMain.php

Where you will arrive at our beautiful main page, you can return here at any time by simply clicking the PhotoArt logo nestled in the top left of the navigation bar:



Next we will click on the 'Register' button (however, you can also browse the site without registering an account)

A registration form titled "Sign up" with a camera icon. The form includes input fields for First Name, Last Name, Email, Username, Password, and Confirm Password. A large "Register" button is at the bottom, with a note below it stating "By signing up you agree to our Terms and Conditions".

Sign up

First Name Last Name

Email

Username

Password

Confirm Password

Register

By signing up you agree to our Terms and Conditions

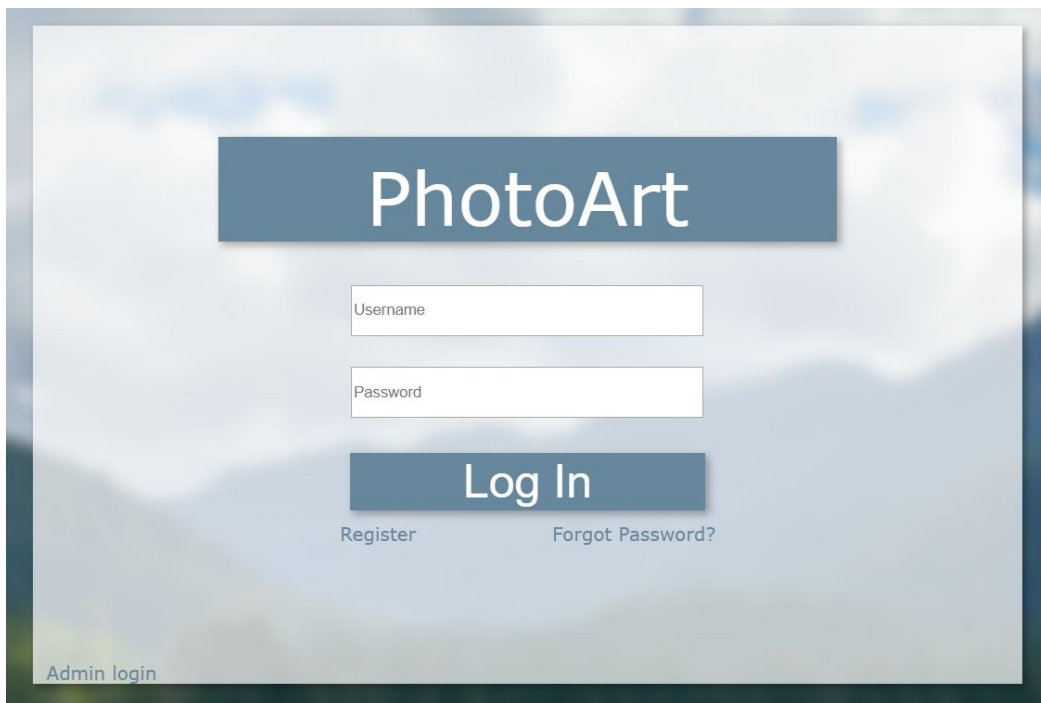
Here you can fill out the 'Register' form to create an account.

Once an account has been created, you will be brought back to the main page:



Note that the header now says 'Hello, tbrady' ('tbrady' was our chosen username) and the 'Login' tab has been replaced by a 'Logout' tab, because we are now logged in.

Or, if you already have an account you can simply logon using the 'Login' button, which brings you to the following form:



We will log in with the same username 'tbrady' and continue the walkthrough.

Next, we will use the search bar.



First we will search for a photo that is not sold on our site: 'Ramon eating ramen'. This is the result.




Now we navigate back to the main page by clicking the convenient link 'Return to the previous page'.

Now we search 's' in the search bar, which results in all pictures with 's' in the title as well as their respective descriptions:


There are 9 results!

Forest




A walk in the lush, green forest

Crops




Crops with a view of mountains in the background

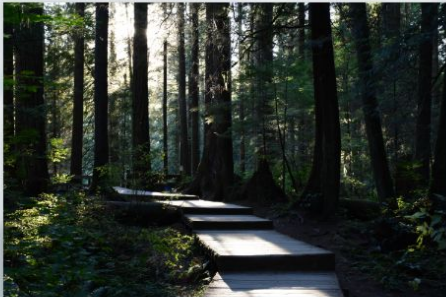
Squirrel



Trees



Suppose we want more information on the picture 'Forest', so we click on the photo, which takes us to the Image Info page.



Forest

Image Info

A walk in the lush, green forest

In Stock: 0
Quantity:

[Share](#)

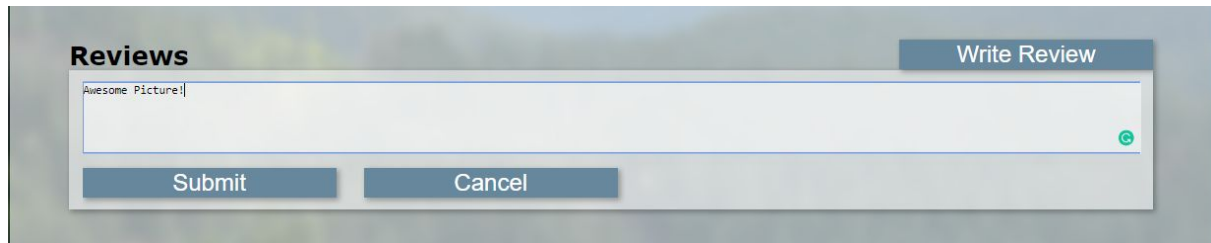
\$39.99

Add to Cart

Reviews

[Write Review](#)

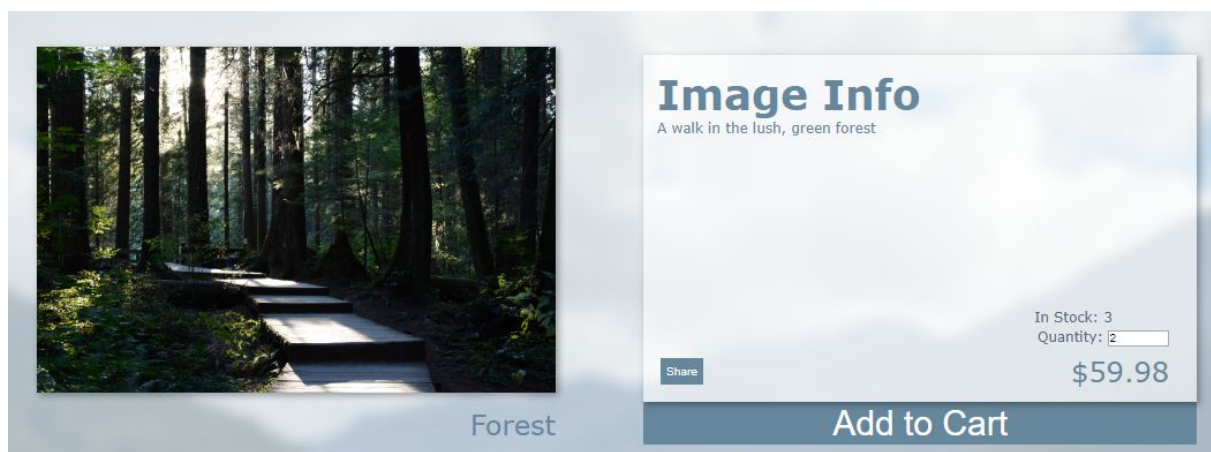
From here we can leave a review by clicking the 'Write Review' button and typing in a review beneath:



The screenshot shows a 'Reviews' section. At the top right is a 'Write Review' button. Below it is a text input field containing the text 'Awesome Picture!'. At the bottom of the input field is a small green circular icon with a 'G' inside. Below the input field are two buttons: 'Submit' and 'Cancel'.

After submitting we get a confirmation message saying 'Review has been added'

From the Image Info page we can also put an item in the cart and see how many items are in stock. Let's add 2 copies of this photo to our cart (note that it automatically updates the price when you change the quantity).



The screenshot shows the 'Image Info' page for a forest photo. On the left is a large image of a forest path. Below the image is the caption 'Forest'. To the right of the image is a white box containing the following information: 'Image Info' in large blue text, 'A walk in the lush, green forest' in smaller blue text, 'In Stock: 3' in grey text, 'Quantity: 2' in a text input field, a 'Share' button, and the price '\$59.98' in large blue text. At the bottom of the white box is a large blue button that says 'Add to Cart'.

Once we've added the item to our cart, we click on the cart tab. From here we can remove the item or checkout.



The screenshot shows the 'Shopping Cart' page. At the top left is the title 'Shopping Cart'. Below it is a white box containing the following information: a small image of the forest path, the caption 'Forest', a 'Remove Item' button, 'In Stock: 1' in grey text, 'Quantity: 2' in a text input field, and the price 'Price: \$59.98' in blue text. Below the white box is a large white box containing the text 'Subtotal: \$ 59.98'. To the right of this box is a large blue button that says 'Checkout'.

We will place the order by clicking 'Checkout'. Which takes us to the 'Payment and Shipping Information' page, which we have filled out with our information.

Payment and Shipping Information

First Name: Tom, Last Name: Brady

Shipping Address: 762 Dehart Rd.

Country: CA, Province: BC

Email: tylerkreway@gmail.com

Payment method: Visa

CardNumber: 4516214154123124, Expiry Date: January, 2020, CSV: 242

Subtotal: \$59.98
GST tax 5.00%: \$3
PST tax 7.00%: \$4.2
Total: \$67.18

Checkout

After clicking 'Checkout', here is what we can see.

Payment and Shipping Information

First Name: Tom, Last Name: Brady

Shipping Address: 154 ritohie court

Country: CA, Province: BC

Email: tbrady@gmail.com

Payment method: MasterCard

CardNumber: 3450032168372901, Expiry Date: January, 2020, CSV: 103

Subtotal: \$59.98
GST tax 5.00%: \$3
PST tax 7.00%: \$4.2
Total: \$67.18

Checkout

After filling out this information, we click 'Checkout' to confirm our order.

We receive this message to confirm that the order has been placed.

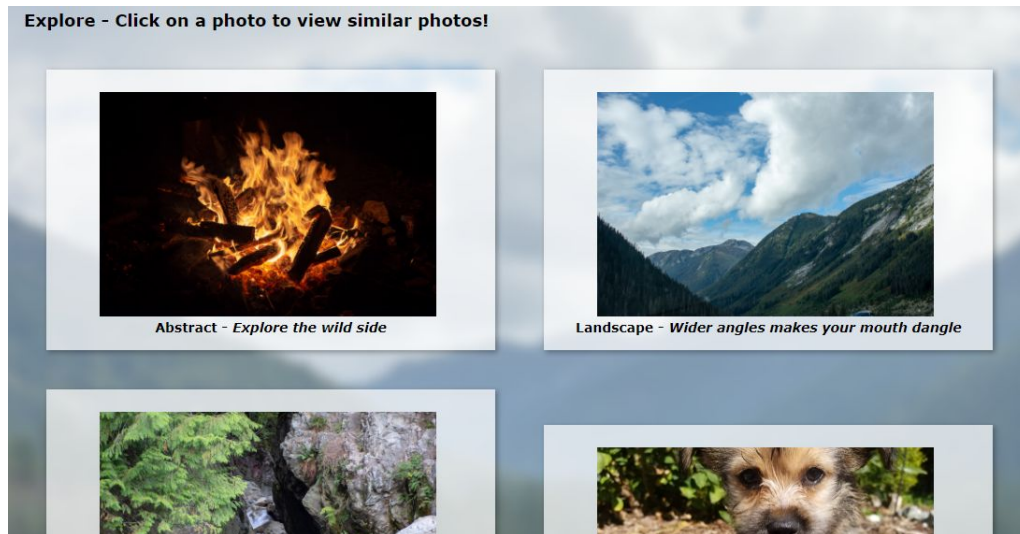
cosc304.ok.ubc.ca says

Thank you, the items have been ordered.

OK

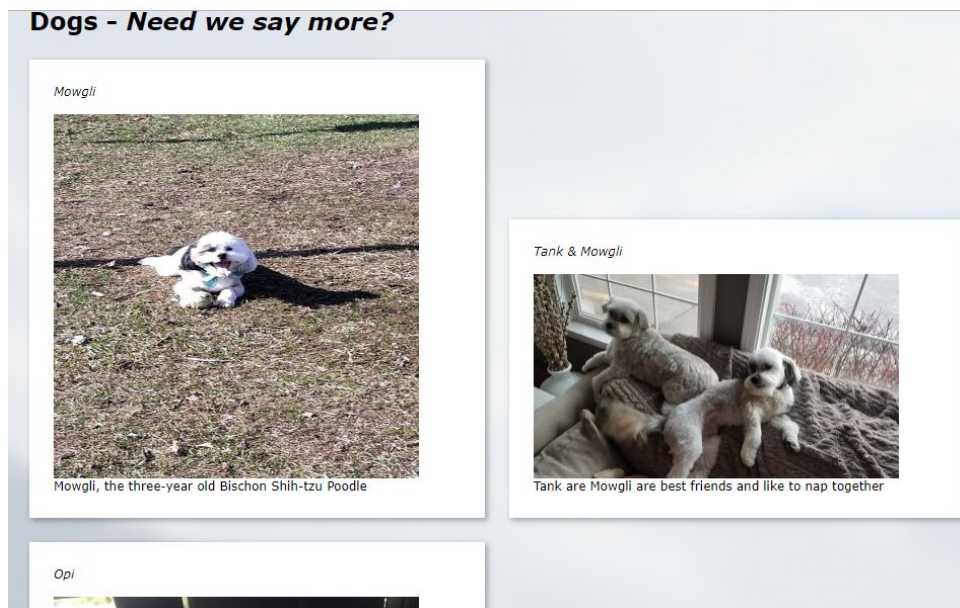
Then, we are taken back to the main page.

Now we will check out the 'Explore' tab.



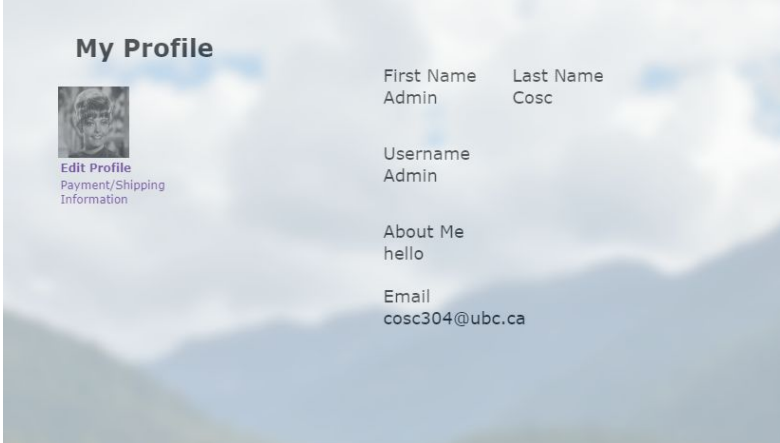
This page displays our four categories of photos. From here you can select the category you are interested in and you will see photos that qualify as this category (and are similar to the category photo).

We will click the 'Dog' category.




Clicking on any of these photos will take us to the Image Info page, which we have already covered.

Now we will click the My Profile tab
From here we can view info about our profile:



My Profile


[Edit Profile](#)
[Payment/Shipping Information](#)

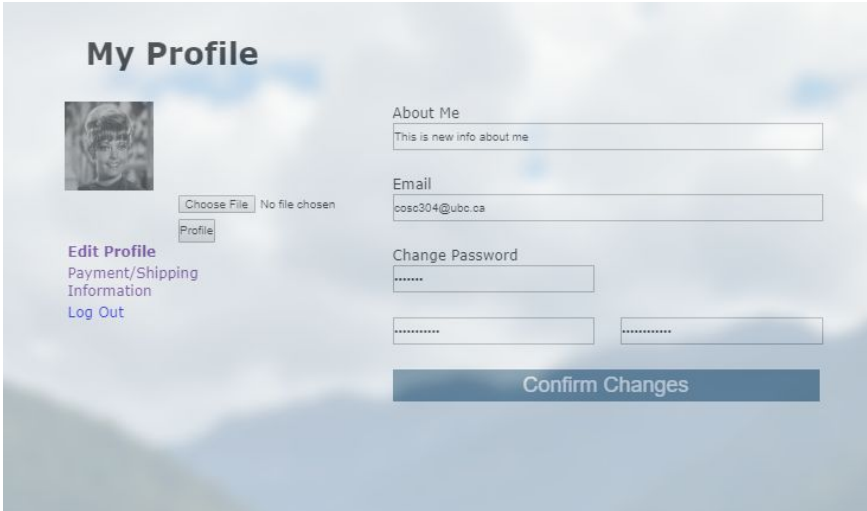
First Name	Last Name
Admin	Cosc

Username
Admin


About Me
hello

Email
cosc304@ubc.ca

Edit our profile by clicking 'Edit Profile', I updated the About Me section and changed my password. Clicking 'Confirm Pages' takes us back the 'My Profile' page and includes the updated information.



My Profile


 No file chosen

[Edit Profile](#)
[Payment/Shipping Information](#)
[Log Out](#)


About Me

Email

Change Password

Edit our payment and/or shipping information by clicking 'Payment/Shipping Information', I updated the names and credit card information. Clicking 'Save Changes' takes us back the 'My Profile' page.

Payment and Shipping Information

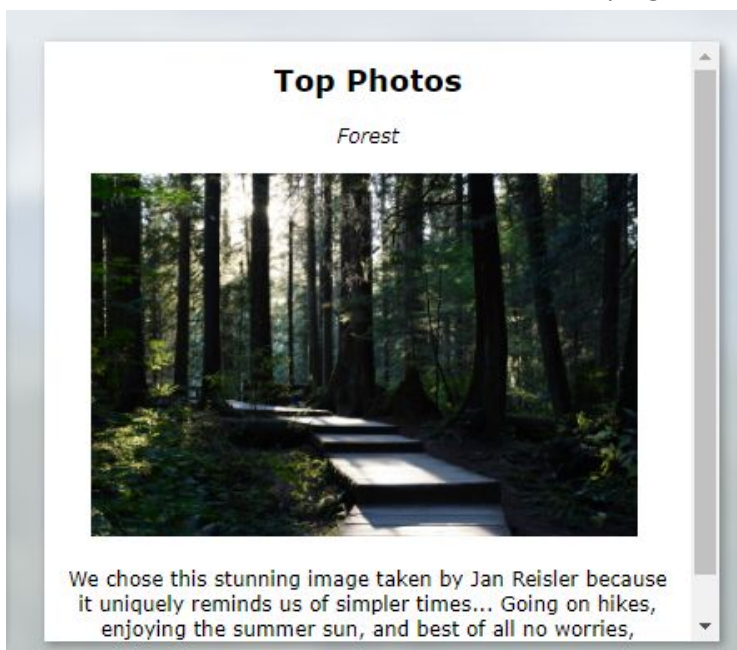


[Edit Profile](#)
[Payment/Shipping Information](#)
[Log Out](#)

First Name	Last Name	
<input type="text" value="Tom"/>	<input type="text" value="Brady"/>	
Shipping Address		
<input type="text" value="154 nitohie court"/>		
Country	Province	
<input type="text" value="CA"/>	<input type="text" value="BC"/>	
City	Postal Code	
<input type="text" value="Kelowna"/>	<input type="text" value="V1V 1Z"/>	
Payment method		
<input type="text" value="MasterCard"/>		
Card Number	Expiry Date	CSV
<input type="text" value="2147483647"/>	<input type="text" value="January, 2020"/>	<input type="text" value="103"/>

Save Changes

Our last user feature is back on the main page.



The 'Top Photos' section shows the top selling photo and a description of the photo, including where it was taken and the professional PhotoArt photographer who took it. Clicking the photo brings you to a page that shows you the top 5 highest selling photos of the week and a description of those photos. Clicking the photo brings you to the product information page.

Top 5 favourite photos of the week

1. Forest



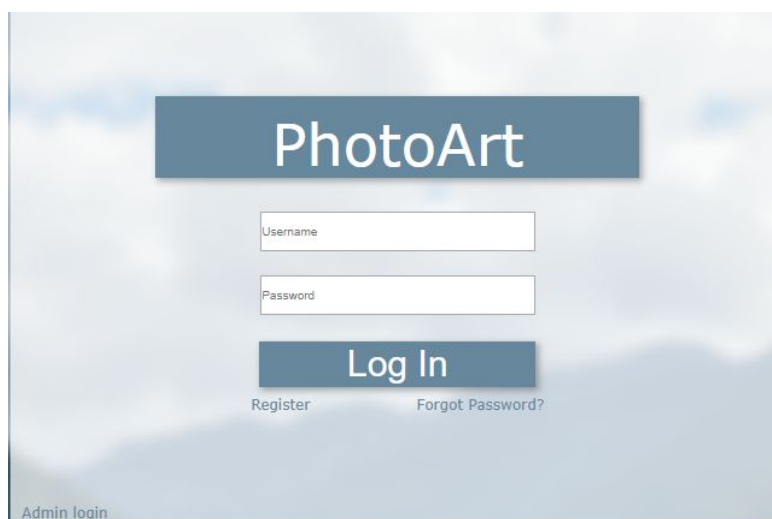
A walk in the lush, green forest

2. Creek



Lastly, we will log out using the 'Logout' tab. This concludes our *USER* walkthrough.

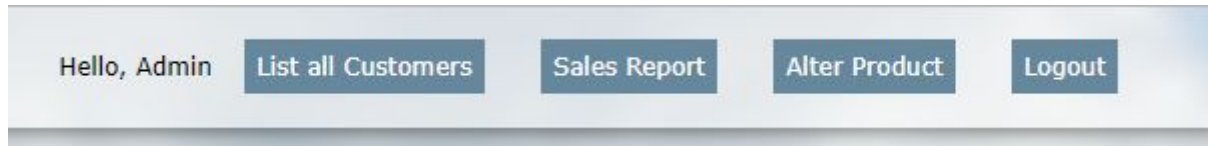
Now we will demonstrate the *ADMIN* tools. Click the login button and draw your attention to the bottom left of the form we can see the 'Admin login' button, go ahead and click it.

A screenshot of the PhotoArt login page. The page has a light blue background with a subtle cloud pattern. At the top, there is a dark blue rectangular box containing the text "PhotoArt" in white. Below this, there are two white input fields: the first is labeled "Username" and the second is labeled "Password". Below the password field is a dark blue button with the text "Log In" in white. Underneath the "Log In" button, there are two links: "Register" on the left and "Forgot Password?" on the right. In the bottom left corner of the page, there is a small link that says "Admin login".

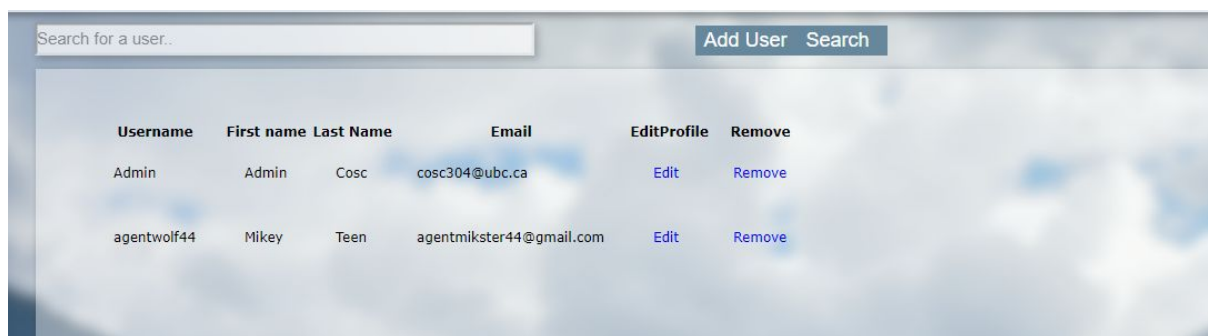
We will log in with our **admin credentials**:

Username: *Admin* Password: *12345678*

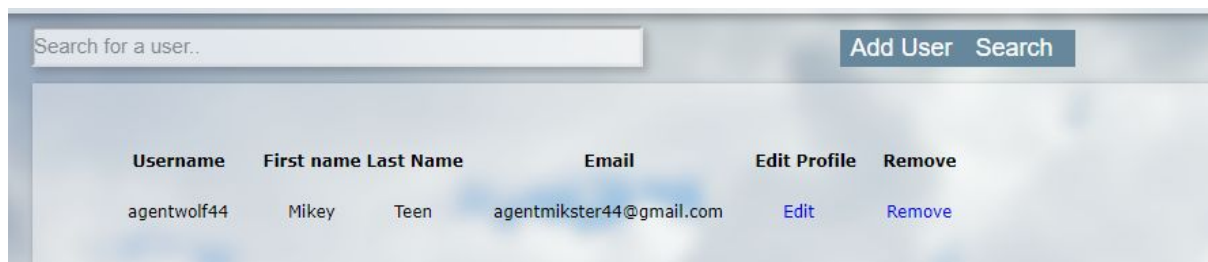
This will bring us back to the main page but now we have special tabs along the top catering to admin capabilities.



Clicking the 'List all Customers', we can see a list of all registered users.



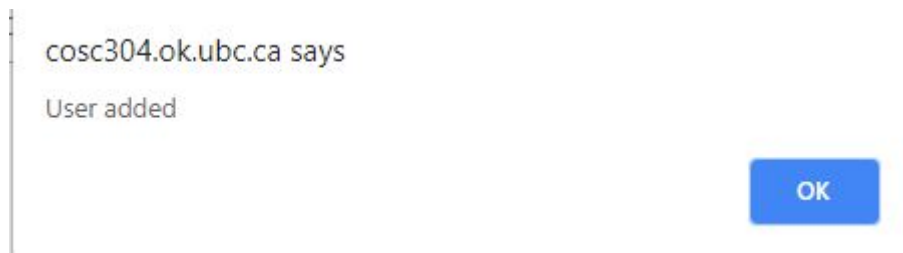
We can search through our users, searching their username, first name, last name or email by hitting 'Search' or pressing ENTER.



We can also directly add users to this list by clicking 'Add User', we then fill out the following information.

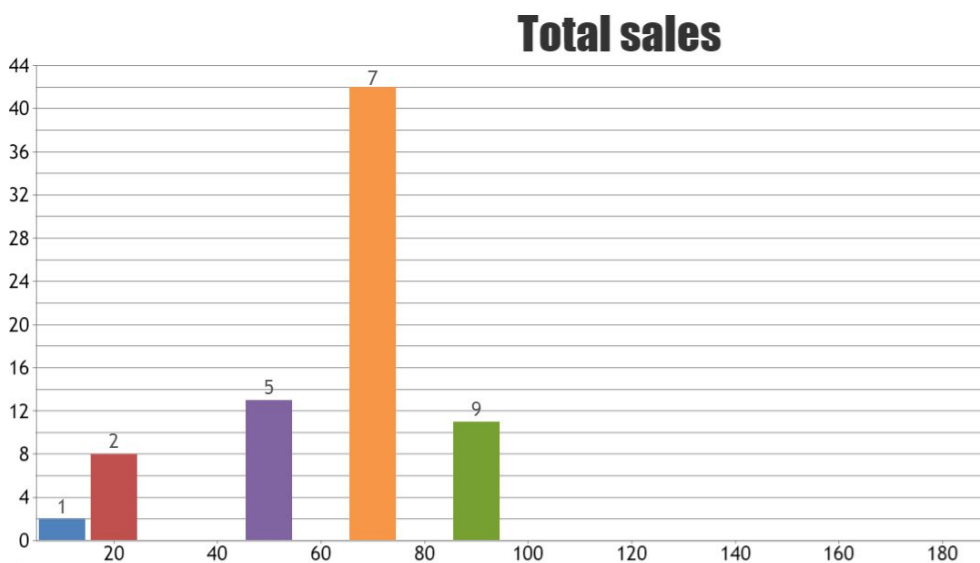
Username	First Name	Last Name	Email	Password
<input type="text" value="tbrady"/>	<input type="text" value="Tom"/>	<input type="text" value="Brady"/>	<input type="text" value="tbrady@gmail.com"/>	<input type="password" value="flatfootballs"/>
<input type="button" value="Add User"/>				

Clicking 'Add User' then gives us this confirmation message:



Then clicking 'OK' brings us back to the 'List all Customers' page, where the new user 'tbrady' has been added to the list (and the new user can then be searched for the same as anyone else).

Clicking the Sales Report tab, gives us a report on the company sales, which measures the amount of times products have been sold:



Then click 'Alter Product' tab. We can search through our users, searching their UPC (picture id), title, category, price or email by ,hitting 'Search' or pressing ENTER. We will search 'Forest' , which returns:

UPC	Title	Category	Price	Image Link	Description	edit	remove
1	Forest	Nature	39.99	../client_side/images/Nature/Forest.JPG	A walk in the lush, green forest	Edit	Remove

From here we can either remove the photo or edit the photo.

Clicking 'edit' brings us to this page, where we changed the price from \$39.99 to \$29.99.

UPC	Title	Category	Price	Imagelink
<input type="text" value="1"/>	<input type="text" value="Forest"/>	<input type="text" value="Nature"/>	<input type="text" value="29.99"/>	<input type="text" value="../client_side/images/Nature"/>
<input type="button" value="Confirm Changes"/>				

Once we click, 'Confirm Changes' we get this message

cosc304.ok.ubc.ca says

item updated!

OK

Then we are brought back to the 'Alter Product' page, where the price of Forest has been updated from \$39.99 to \$29.99.

Search for a product..				<input type="button" value="Add Product"/> <input type="button" value="Search"/>			
UPC	Title	Category	Price	Imagelink	Description		
1	Forest	Nature	29.99	../client_side/images/Nature/Forest.JPG	A walk in the lush, green forest	Edit	Remove

Note: If you log out of an admin account and want to log back into a user account you must click the 'User Login' link on the login page.

That concludes the walkthrough of our website, we hope you enjoyed it.

7.0 Known Issues

7.1 Incomplete Features

- Doesn't automatically update and can't update quantity while in cart view
- The numbers on the x-axis on Total Sales are meaningless and are only used to space out the different products.
- You can decrease the quantity of a product to put into your cart (to a minimum of 1) with the arrows but you cannot increase the quantity
- When you add a review, it is added to the database, but you can't view it under reviews

7.2 Bugs/Errors

- After submitting a review, if you try to place an order it gives you the wrong toast notification. It does add it to your cart but instead of getting the desired message 'Item(s) added to your cart', you get the message 'Review has been added' again. Refreshing the page fixes this issue.
- On the 'Alter Product' tab the 'Add Product' and 'Search' buttons are overlapping.
- When you try to 'Save Changes' when editing 'Payment/Shipping Information, the site crashes

8.0 Conclusions

While this document focused primarily on the website functionality and the database implementation, we want to be clear in our objectives as a company. *PhotoArt* has one simple goal; to bring people's dreams into their homes. We are, and will always be, all about the photos.

We hope that this document was informative, easy to follow and showcased all the technical work that went lovingly into this project. Thank you for your time.

The PhotoArt Team
Showcase the Beauty of Our World