



Web Programming and Database Development Final Project

Art by Blind

Submitted to fulfill requirements in
CWEB2135, CWEB2121, and CWEB2111 Spring 2018
Instructors: Paula Merns and Amalan Pulendran

Project Members

Rob Walczak, walrobcdunwoody.edu, Project Lead
Dakota Shell, shedakddunwoody.edu, Developer
Isaac St. Martin, stmisaa@unwoody.edu, Developer
Andrew Blesener, LSmith@unwoody.edu, Developer
Alexander Cervantes, cerale@unwoody.edu, Developer

Approved By:

Paula Merns

Amalan Pulendran

May 15th, 2018

Table of Contents

Contents

Version Log	3
1. Project Overview, Introduction and Requirements Specification	3
1.1. Introduction	3
1.2. Project Overview	4
1.3. Project Scope	4
1.4. Project Requirements	4
1.5. Definitions, Acronyms, and Abbreviations used in documentation	4
2. Analysis	4
2.1. Description of Problem	4
2.2. Technologies Used	5
2.2.1. MySQL	5
2.2.2. HTML/ CSS/ Bootstrap	5
2.2.3. PHP	5
2.2.4. JQuery	5
2.2.5. GoDaddy (Web Hosting)	5
2.3. System Architecture	5
2.4. Use Case Diagrams and Descriptions	6
3. Design	7
3.1. Architecture design	7
3.2. Interface design	10
3.3. **Source Code	10
3.4. Data Model and Storage	10
3.5. Security Considerations	10
4. Implementation	11
4.1. Design Model/Folder structure	11
4.2. Description of major component files	11
4.3. Major shared code files/modules	11
4.4. Major shared database elements/database connectivity/injection protection	12
4.5. Major shared visual elements/styles	15
Testing and On-going Maintenance	16
4.6. Overview of Site Test Plan	16
4.7. Methods used for testing applications	16

4.7.1. Unit Tests (recommended)	16
4.7.2. Interactive Tests (required).....	16
4.7.3. User Tests (optional)	17
4.7.4. External Validations (optional)	17
4.8. Methods used for maintaining application ()	17
4.8.1. Overview of Maintenance Plan	17
4.8.2. Continuity and hand-off information.....	17
4.8.3. Continuity and hand-off information (Cont.)	17

Version Log

- 0.0 Formed Project
- 0.1 Created diagrams and mocks
- 1.0 Built Site
- 2.0 Fixed CSS elements
- 3.0 Designed PHP system locally
- 4.0 Implemented the site on live and fixed PHP
- 5.0 Finalized and Submitted Project

1. Project Overview, Introduction and Requirements Specification

Art by Blind is a for-profit located in Khartoum, Sudan that helps aid the visually disabled. The school allows students to creating stunning arts and crafts while providing a therapeutic way to express themselves. This organization is looking for a way to display and sell these artworks to help alleviate the hardships that these students face.

Our Project goal is to develop an ecommerce site for a client, to sell pieces of art made by blind Sudanese artisans who are learning the “craft as trade” business. We would like to give these artisans a chance to support themselves with their artistic abilities.

We need to create an Asp.Net website application with a MSSQL Database, Login/Sign up, cart and checkout capabilities. Along with a product page for the art pieces, and information pages for about the artists, our vision, and how to contact us.

Checkout and Logins will need encryption to make the site secure. We will attempt to complete this but third party applications are an option.

Shipping software may need to be third party, as we are unaware of the complexity of that aspect.

The Admin page needs a way to upload and edit our products.

1.1. Introduction

Our team consists of Alexander Cervantes, Dakota Shell, Rob Walczak, Issac St. Martin, and Andrew Blesener. Our team name is ARAID. The classes involved for this project are Database Architecture, Web Development, and Capstone Topics.

1.2. Project Overview

Our project will connect with Art by Blind and create a functional e-commerce site and appropriate database for people to upload content and sell items via the website.

1.3. Project Scope

We intend to have a completely functional store website that allows for storage of items in a database. This database will store any products Ream looks to sell, allowing them to profit specifically through the website. Customers will be able to come, learn about ABB and purchase items directly through this website. They will need to register an account for verification and order confirmation.

Will Include:

- Functional and Responsive e-commerce site.
- Saved, safe and secure online transaction marketplace
- Database that can help manage and maintain available products.
- System Administration
- User Side Login

Will Not Include:

- Live Streaming
- Live Chat
- Forums

1.4. Project Requirements

As a User, I would like to create an account.

As a User, I would like to purchase art.

As a System Administrator, I would like to see who has purchased art.

As a Database Administrator, I would like to edit the art within the database

1.5. Definitions, Acronyms, and Abbreviations used in documentation

2. Analysis

2.1. Description of Problem

The problem that currently faces Art by Blind is that they do not have a web-presence that allows users to buy products from a web page. By lacking this, they are not properly creating the web presence to explain this program to people all around the world.

We are focusing on creating a storefront for them that will solve this problem. By creating a website that will be manageable by Reem (and possibly other CWEB students), we are opening a path that could greatly the visibility as well as increase the amount of funding and that goes into this program.

2.2. Technologies Used

2.2.1. MySQL

MySQL will be used for our database system. We plan to use and create tables for the website that including Product, Admin, and User tables. We will also make sure there is a copy of the create statement so that all tables we create can easily be recreated in the future.

2.2.2. HTML/ CSS/ Bootstrap

We are using HTML5 as our website base, with CSS and Bootstrap for styling. The site will be white with black borders with light blue highlights on specific buttons.

2.2.3. PHP

PHP will aid us in back-end functionality allowing our site to talk to the database and fetch information such as products, user information, and necessary admin functionality.

2.2.4. JQuery

JQuery will be used for additional functionality with our website such as an image carousel on the main page, drop-down menus, and other user interface items.

2.2.5. GoDaddy (Web Hosting)

GoDaddy will be the hosting chosen by Reem, allowing for the management through cPanel for creating and displaying the functional, live website.

2.3. System Architecture

The system databased is focused around MySQL, by using MySQL and importing a database made locally, we will be able to use PHP for an admin login so that Reem will be able to create admins, add/edit user information, and manage products. We will be including the database creation script so that in case the database fails; there will be a way to recreate the tables on this site. HTML, CSS, and JQuery were used in the structure of the front-end, allowing us to create a sleek design with intuitive, yet functional user interface.

2.4. Use Case Diagrams and Descriptions

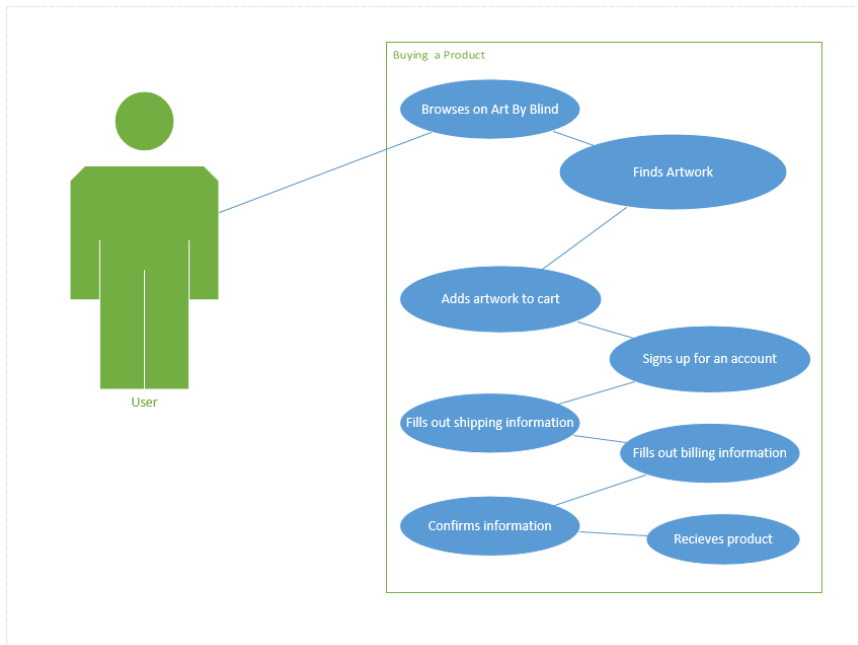


Figure 1 User buying a Product UML

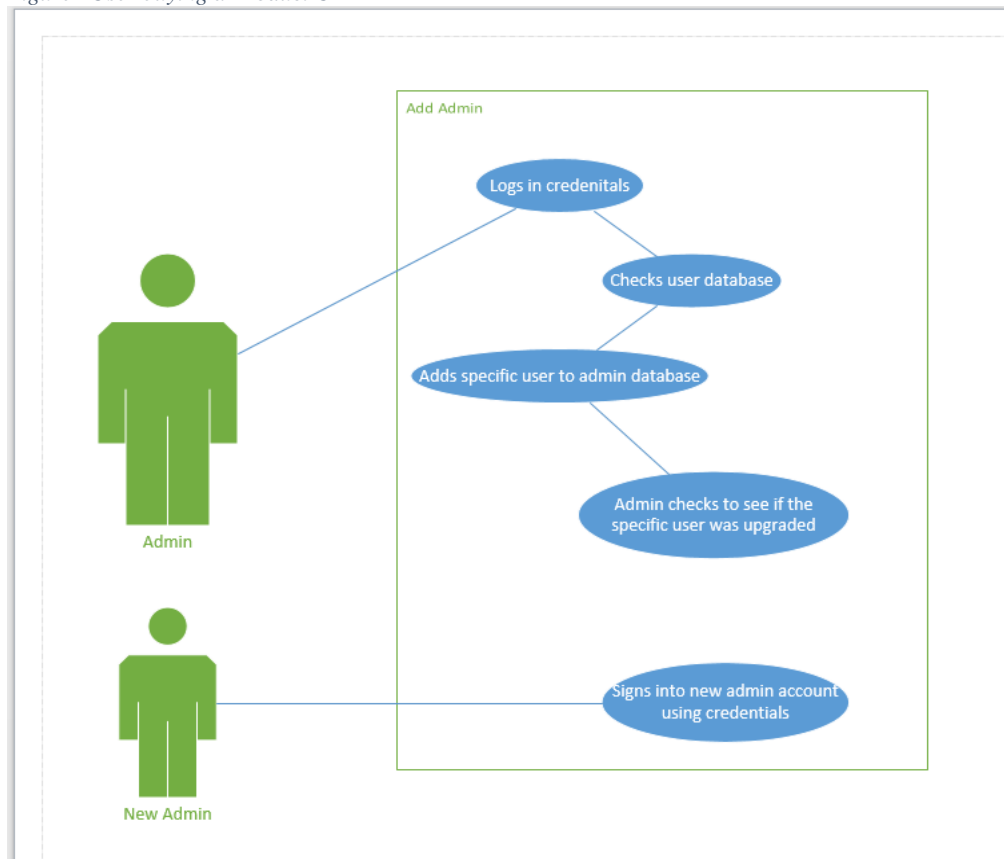


Figure 2 UML 2

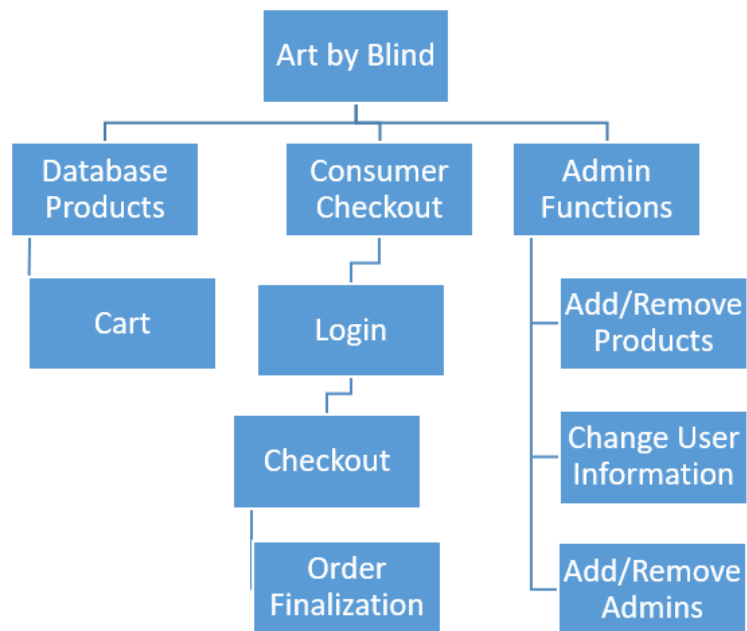


Figure 3 DDL

As a User:

- I want to be able to create an account
- I want to be able to log on
- I want to be able to purchase products
- I want to be able to verify shipping information
- I want to be able to check personal order history

As an Admin:

- I want to be able to maintain the website's database of products
- I want to be able to restrict users from malicious activities
- I want to be able to monitor purchases through the site
- I want to be able to post content to the site
- I want to be able to service customer accounts when necessary

3. Design

3.1. Architecture design

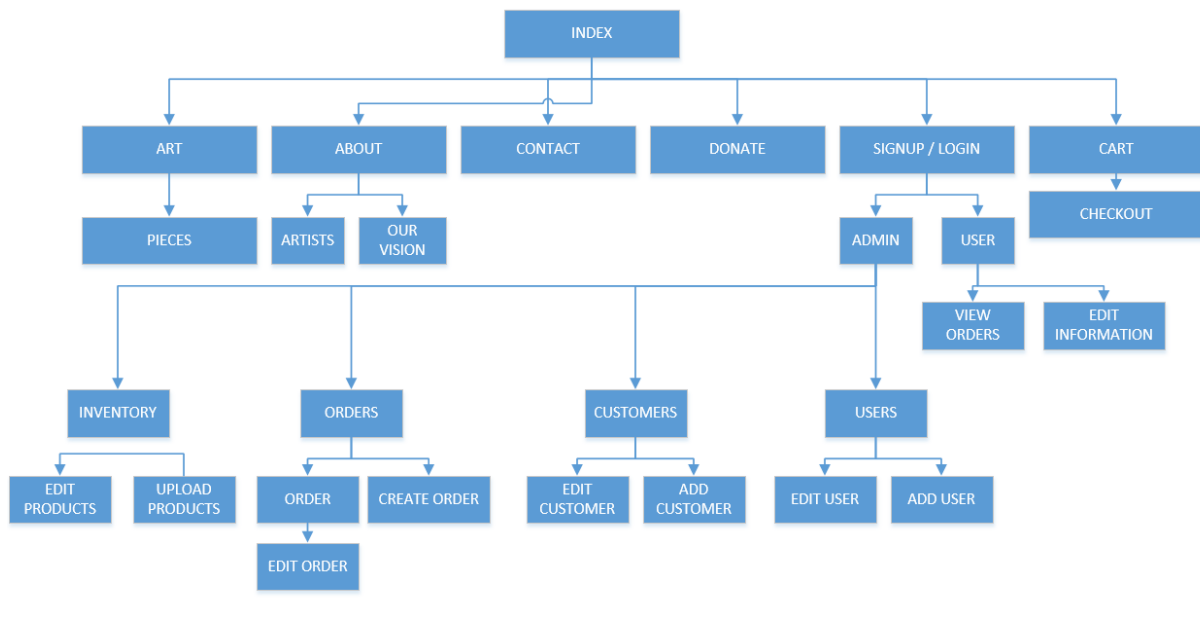


Figure 4 Web Site Hierarchical Structure

ART by BLIND						ART	ABOUT	CONTACT	ADMIN	CART	LOGIN	REGISTER
remove	Art Title	Variant	Price	Quantity	Line Subtotal							
<input type="checkbox"/> Remove	necklace presentation	Gold	25.00	<input type="text" value="5"/>	\$125							
					Sub Total: \$125							
Update Cart			Continue Shopping			Check Out						

Figure 5 Order

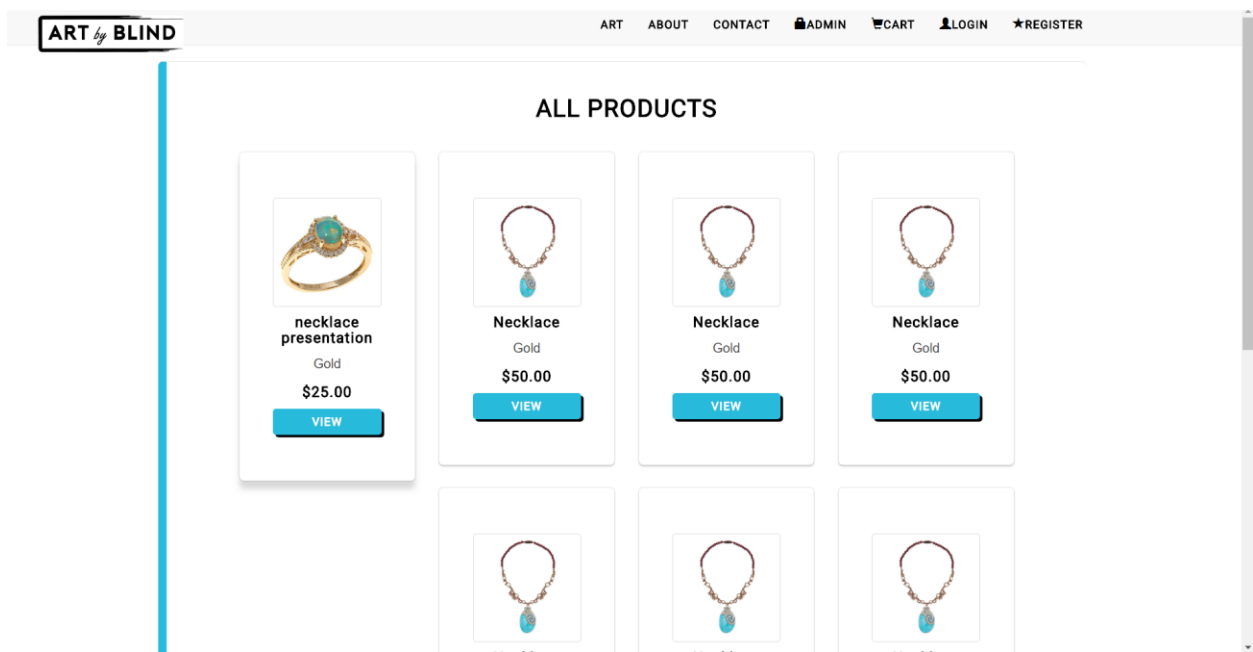


Figure 6 Inventory

ART by BLIND

ARTABOUTCONTACTADMINCARTwalrobcLog Out

PRODUCTS

ORDERS

CUSTOMERS

USERS

EDIT PRODUCTS

Title

necklace presentation

Variant

Gold

Description

Description

Price

\$25.00

Quantity

5

EDIT

DELETE

Image

UPLOAD IMAGE





Figure 7 Add Items


3.2. Interface design

LOG IN

Log In

ART  BLIND

ART ABOUT CONTACT ADMIN CART LOGIN REGISTER



Necklace

Gold

\$50.00

ADD TO CART

With Javel

Figure 8 Pages

3.3. **Source Code

<https://github.com/andrewblesener/ArtByBlind>

All parties have access and can create additional information as files.

3.4. Data Model and Storage

3.5. Security Considerations

Refer to 4.4. DB connection, password encryption, configuration files and what information is being implemented.

PHP INI

[currentEnv]

data = local

```

[local]
db_host      = "localhost"
db_user      = "root"
db_pass      = ""
db_database  = "artbyblind"

[hosted]
db_host      = "localhost"
db_user      = "Reemelradi"
db_pass      = "Dunwoody1!"
db_database  = "artbyblind"

```

4. Implementation

4.1. Design Model/Folder structure

This should be a graphic depiction of the overall structure of your code. What directories at the top level contain what code bits? Where are the configurations files? Is there a common styles library or one at each level or both?

4.2. Description of major component files

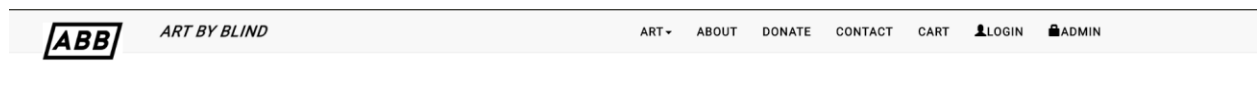


Figure 9 Navigation

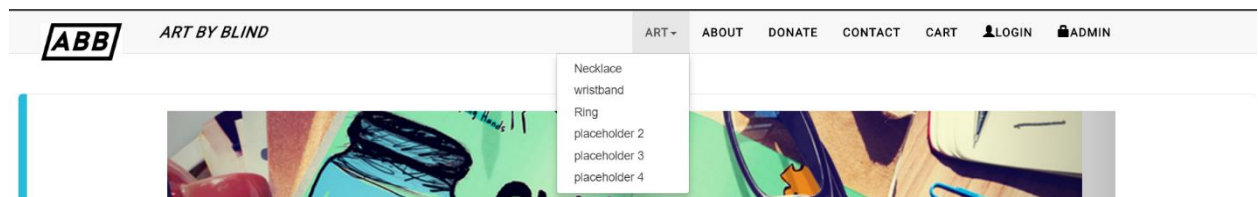


Figure 10 Drop Down Bar

4.3. Major shared code files/modules

TEMPLATE:

```

<?php
include('connect.php');
?>

<!doctype html>
<html>
<head>
<?php include('head.php');?>
</head>
<body>

```

```

<?php include('nav.php');?>
<?php include('footer.php'); ?>
</body>
</html>

```

CONNECT:

```

<?php
    /* Database config */
    $db_host = 'localhost';
    $db_user = 'root';
    $db_pass = '';
    $db_database = 'artbyblind';
    /* End config */
    $db = new PDO('mysql:host=' . $db_host . ';
    dbname=' . $db_database,
    $db_user, $db_pass);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
?>

```

4.4. Major shared database elements/database connectivity/injection protection

```

<?php
/* Database config */
$db_host = 'localhost';
$db_user = 'bleandd';
$db_pass = 'password01';
$db_database = 'artbyblind_users';
/* End config */
$db = new PDO('mysql:host=' . $db_host . ';
dbname=' . $db_database,
$db_user, $db_pass);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
?>

[currentEnv]
data = databases

[local]
db_host = "localhost";
db_user = "root";
db_pass = "";
db_database = "art_by_blind";

[databases]
db_host = "localhost";
db_user = "blindadmin";

```

```
db_pass = "g_@X+z3d-A4%";
db_database = "art_by_blind";
```

```
-- phpMyAdmin SQL Dump
-- version 4.0.10.18
-- https://www.phpmyadmin.net
--
-- Host: localhost:3306
-- Generation Time: May 10, 2018 at 08:51 AM
-- Server version: 5.6.39-cll-lve
-- PHP Version: 5.6.30
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
```

```
--
-- Database: `art_by_blind`
--
```

```
-----
```

```
--
-- Table structure for table `cart`
--
```

```
CREATE TABLE IF NOT EXISTS `cart` (
  `ip_add` varchar(255) NOT NULL,
  `prod_id` int(10) NOT NULL,
  `qty` int(10) NOT NULL,
  PRIMARY KEY (`prod_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `cart`
--
```

```
INSERT INTO `cart` (`ip_add`, `prod_id`, `qty`) VALUES
('174.219.130.118', 53, 1),
('64.132.94.34', 56, 1),
('64.132.94.34', 54, 1),
('64.132.94.34', 55, 33);
```

```
-----
```

```
--
-- Table structure for table `orders`
--
```

```
CREATE TABLE IF NOT EXISTS `orders` (
  `orderId` int(5) NOT NULL AUTO_INCREMENT,
```

```

`CustID` int(5) NOT NULL,
`ship_address` varchar(255) NOT NULL,
`total_sale` decimal(10,0) NOT NULL,
`date` varchar(255) NOT NULL,
`ship_status` varchar(255) NOT NULL,
PRIMARY KEY (`orderID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Table structure for table `order_line`
--

CREATE TABLE IF NOT EXISTS `order_line` (
  `order_lineID` int(5) NOT NULL AUTO_INCREMENT,
  `orderID` int(5) NOT NULL,
  `productID` int(5) NOT NULL,
  PRIMARY KEY (`order_lineID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Table structure for table `person`
--

CREATE TABLE IF NOT EXISTS `person` (
  `perID` int(11) NOT NULL AUTO_INCREMENT,
  `first_name` varchar(255) NOT NULL,
  `last_name` varchar(255) NOT NULL,
  `access_level` varchar(255) NOT NULL,
  `user_name` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  `phone_number` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  PRIMARY KEY (`perID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

--
-- Dumping data for table `person`
--

INSERT INTO `person` (`perID`, `first_name`, `last_name`, `access_level`, `user_name`, `password`,
`address`, `phone_number`, `email`) VALUES
(10, 'test', 'test', 'user', 'test',
'$2y$10$XDslVV6mRl3Iukc4mQl5HOYMcUMS02BKL8j4GCCRKDk4MPh9aiIoa', '2 et', '7894561230',
'test@test.com'),
(11, 'mytest', 'y', 'admin', 'mytest',
'$2y$10$8bYshzhTRhfT9M3mwVfhG.iNqO2oa3HHIvLbe3Gf7zGnKmn1D6Oe.', '7894561523',
'1546546', 'm@bob.com');

-----

--

```

```
-- Table structure for table `products`
--

CREATE TABLE IF NOT EXISTS `products` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(50) NOT NULL,
  `variant` varchar(50) NOT NULL,
  `description` varchar(255) NOT NULL,
  `price` decimal(10,2) NOT NULL,
  `quantity` int(11) NOT NULL,
  `image` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=65 ;

--
-- Dumping data for table `products`
--

INSERT INTO `products` (`id`, `title`, `variant`, `description`, `price`, `quantity`, `image`) VALUES
(55, 'Necklace 55', 'Gold', 'A product description is the copy that describes the features and benefits of a product to a customer. The goal of the product description is to provide the customer with enough information to compel them to want to buy the product immediately.', '50.00', 3, 'images/necklace.jpg'),
(56, 'hexagons', 'green and black', '', '50.00', 1, 'images/featured-product-1.jpg'),
(57, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(58, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(59, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(60, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(61, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(62, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(63, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg'),
(64, 'Necklace', 'Gold', 'With Jewel', '50.00', 3, 'images/necklace.jpg');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

4.5. Major shared visual elements/styles

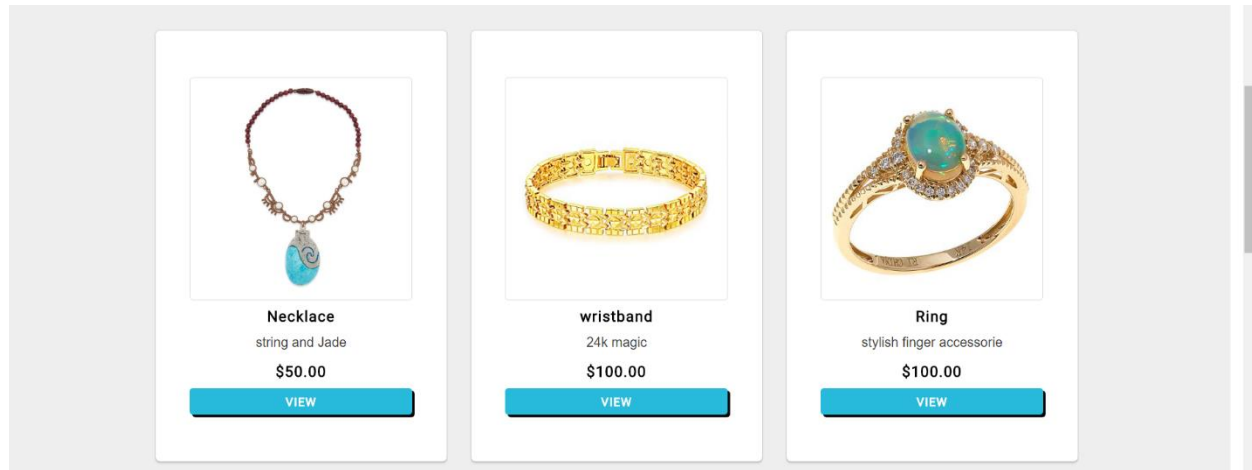


ART BY BLIND

ART - ABOUT DONATE CONTACT CART LOGIN ADMIN



FEATURED PRODUCTS



Testing and On-going Maintenance

4.6. Overview of Site Test Plan

Our testing plan is to make sure that the database is properly set-up so that Reem can add and remove products. We also need to make sure the cart is working properly and functioning as intended.

We plan to use local host testing to make sure the application works as intended with no flaws or bugs. By testing in local before we upload into GoDaddy, we will have a testing environment that will allow us to test as intended.

4.7. Methods used for testing applications

This section describes both your plan for testing and your method of carrying out your test plan. This is required for the Advanced Topics grade. Best practice is to develop a basic test plan in advance of beginning the development/implementation part of the SDLC.

4.7.1. Unit Tests (recommended)

Testing consists of making sure no one flagged as non-admin can access admin pages. We are also testing the database, making sure user information is validated when entered. We are testing that the cart is functional and the user gets a validation email when an order is submitted.

4.7.2. Interactive Tests (required)

We need to test the functionality of the database and adding products to it; we have to test functionality of adding and removing products from the website. We have to test adding and removing users from the database. We have to test to see user account information is correctly taken into the database. Finally, we need to what happens when an order has been finalized and if the correct steps are completed.

4.7.3. User Tests (optional)

Reem will be our user test, as well as cohorts from our class. This will give us a general idea of how the website functions (Ream) and our cohorts will allow us to fix any errors with the code in our base.

4.7.4. External Validations (optional)

We are using W3 Schools to validate our website for any errors within coding. Since the website is hosted, we can get feedback from cohorts. Ream will be our main validation to make sure the website is following her specifications.

4.8. Methods used for maintaining application ()

Our group will not be maintaining this website unless otherwise noted; we find that other students can benefit from working with our program.

4.8.1. Overview of Maintenance Plan

Our codebase will be hosted via GitHub. We will make this a public repository to allow Reem and other students from upcoming years to work from our structure. We also have branches currently in place to make sure that the master is fully protected; we can protect the master and give it to Reem so that she may use it as she sees.

4.8.2. Continuity and hand-off information

We will be handing this project off to Reem to decide what CNTS/CWEB students to aid her in maintaining and updating the site. This website will be good application for database servers/systems, business architecture, and web publishing. By having a steady stream of students to continue work on this project, Reem will have a functional site and students will get experience with a live website.

4.8.3. Continuity and hand-off information (Cont.)

The site needs the following changes/additions:

1. The Addition of PayPal or a Credit Card Validation System.
2. Editing of Shopping Cart IP Validation
3. Remove hashed password & encrypt on edit.
4. Delete should offer option to stop deleting example (“Are you sure you want to delete user Jim/admin Bobby?”)
5. Recommend modal for delete confirmation.
6. Message back to user for admin not available when not *and* only show option when logged in as admin.
7. Not Resize boxes on front/product display with different image sizes.

8. Update code and images based on new information.(n documentation)
9. Add naming standards to document.
10. Remove animation from contact is
11. Add connection for email information to ini.
12. Add email connection
13. (From above bCard sizing shouldn't change with internal image size.)
14. All colors for buttons in same family, fix for color contrast.
15. Add admin folder and place admin functions into that folder.
16. Validation for register and email should work. If attempting to register again, add error.