# CSC 309 Programming on the Web

### Winter 2014

## Assignment 2

## Candy Store

## Due Date: March 20 at 5:00 PM

### Description

Develop a web front for a fictional candy store. Your site should include the following features:

- Create new account.
- User login.
- Browse candy inventory.
- Add candy to shopping cart.
- Edit shopping cart content's (change item quantity, delete items)
- Checkout. This function should collect payment information (credit card number and expiry date) and display a printable receipt (a simple example that shows how to print from JavaScript is available here).
- Email receipt to customer. Documentation for sending email from CodeIgniter is available here. To use this feature you will need access to a public SMTP server. One possiblity is to use the SMTP server provided by Gmail.

### Requirements

You application most comply with the following requirements:

- Implement your application using all of the following technologies: PHP, CodeIgniter, HTML, CSS, JavaScript, JQuery, and DOM. Use CSS style sheets in favor of deprecated HTML tags and attributes.
- For simplicity, assume that only 1 customer will use the site at any given time, i.e., there is no need to implement concurrency control for this assignment.
- Assume that there is a unlimited ammount of items available for any product sold on the site, i.e., you will never run out of a given item.
- Your site should include an administrator page, which should be accesible by login into the site using the "admin" username and a password. The administrator page should enable the administator to perform the following tasks:
    - Add, delete and edit products
    - Display all finalized orders
    - Delete all customer and order information
- Your application should be easy to use.
- The main entry page for your site should be called *index.php*.
- Make sure your site functions correctly in the CDF environment. TAs will not debug so test to ensure it works the first time!
- Store all your data on a MySql database that conforms to the schema shown in Figure 1. An instance of this database has been pre-created for you to use. You will get instructions for how to connect to the database in a separate email. Don't make changes to the database's schema (e.g., add tables, change attribute names). The TA will use their own version of the database to grade your assignment and if you modify the schema, your application won't work. An sql script for recreating the database is available here (you will need this file only if you plan to run MySql on your home machine).
- Each item offered by the store should include a picture.
- The shopping cart should show the total cost for the items selected.
- For unfinalized orders, store shopping cart information on the server using a PHP session. Store information in the provided database only once orders are finalized.
- All data provided by the user should be validated. Enforce the following validation rules:
    - Credit card number must have 16 digits
    - Expiration date should be a valid date (MM/YY)

- The card should not have expired
- All fields in the customer profile should not be null
- Customer email most conform to a valid email format
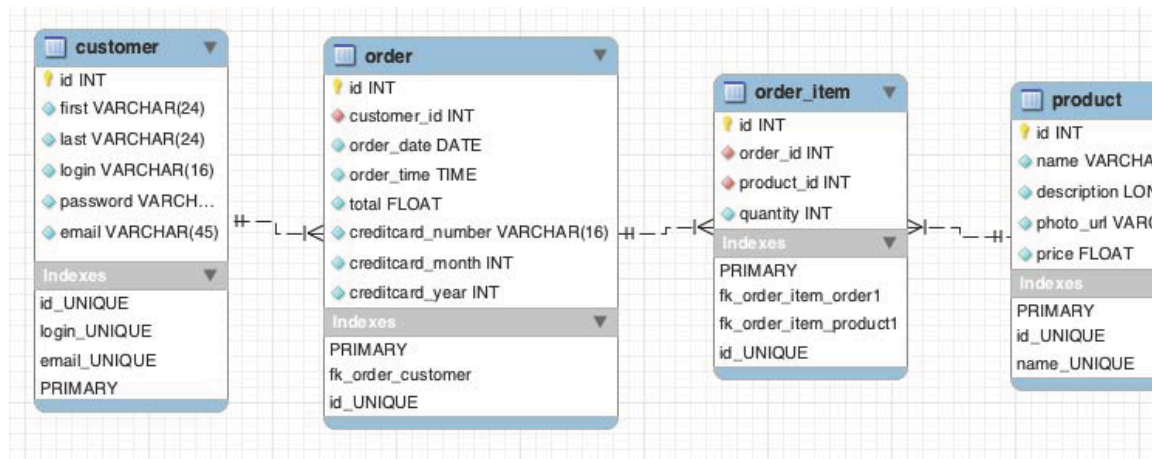- Password most be at least 6 characters long



**Figure 2:** Database Schema

To help you get started, I have created a sample CodeIgniter application called candystore that provides simple functionality for manipulating the products table. To use the application follow these steps:

1. Uncompress and untar the file
2. Install it under apache/htdocs
3. Change the values of the database, username, and password settings in candystore/application/config/database.php to the appropriate values for your database.
4. Start Apache
5. Navigate to http://server_name:your_port_number/candystore

To implement this assignment you will need to setup and run Apache. Instructions for running Apache and Eclipse, as well as for debugging PHP are available on the student guide here.

## Deliverables

Submit your assignment using the BlackBoard system. Submit only one copy per group.

Include a README file with documentation for your Web site (brief explanations of how it all works). Make sure the file includes the names and student numbers of group members.

Submit your assignment as a single file by using a tool like *tar* or *rar* to archive your complete CodeIgniter directory structure.

Note: Do not submit a copy of your database and do not include your SMTP server credentials in your submission. The TAs will use their own.

Finally, all deliverables should be neatly formatted, readable,and be properly documented .

Good luck!