Name: YUNSIK CHOI

[CS 290] Final Project

Outline

The tables in the database contain customers information, their orders, items, and member password information.

Simply put, I am trying to mimic an Amazon.com application even though my project will be a lot simpler version of the Amazon.com application (Little.com).

There are two kinds of users of the website. One is an administrator and the other is a regular user.

Administrator and users need to log in first in order to access to the website. Administrator can manage customers (users). He/she can even add a customer and view passwords if he/she wants. However, administrator is not allowed to make an order. Only customers, regular users can do shopping only. They are not allowed to view any other information such as other users information including their username and password.

* Please note passwords were encoded when being stored in the database. However, when viewing the passwords, they will be decoded to make them readable.

A customer registers by entering his/her information such as first/last name, address, state ID, username, and password. The states table is displayed to find a state ID when entering information. This information will be stored in the users table.

A registered customer logs in first, which his/her username will be stored and used in shopping items.

The items table is displayed to find which items are available when shopping. A customer place items on his/her shopping cart first. Items placed on the cart will be stored in the cart table.

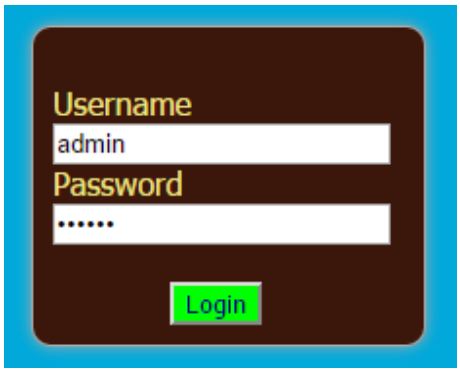
A customer can take items out of the cart if he/she does not want them. This transaction is also updated in the shopping cart table.

Once a customer is satisfied with his/her items on the cart, he/she can make an order. Once the order has been made, ordered items will be displayed along with customer's information such as his/her address.

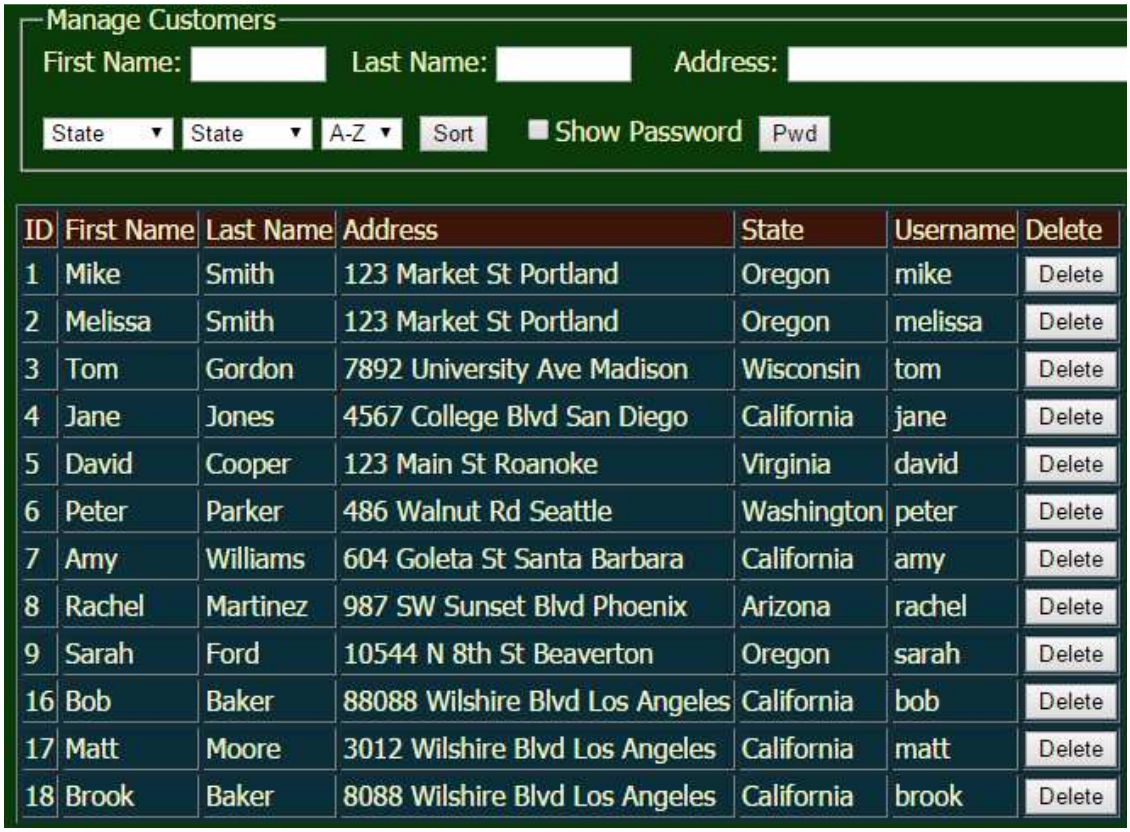
HOW TO USE IT

From the login page
http://web.engr.oregonstate.edu/~choiy/cs290/finalproj/index.php

You can log in as the administrator of the website.
Username is "admin", and password is "admin1". Password is case sensitive.



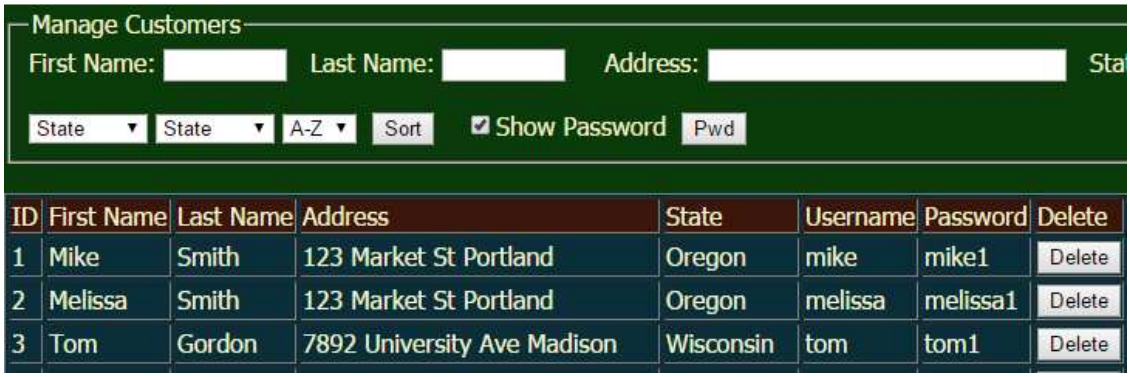
One logged in as the administrator, you can view customers information as shown below.



ID	First Name	Last Name	Address	State	Username	Delete
1	Mike	Smith	123 Market St Portland	Oregon	mike	Delete
2	Melissa	Smith	123 Market St Portland	Oregon	melissa	Delete
3	Tom	Gordon	7892 University Ave Madison	Wisconsin	tom	Delete
4	Jane	Jones	4567 College Blvd San Diego	California	jane	Delete
5	David	Cooper	123 Main St Roanoke	Virginia	david	Delete
6	Peter	Parker	486 Walnut Rd Seattle	Washington	peter	Delete
7	Amy	Williams	604 Goleta St Santa Barbara	California	amy	Delete
8	Rachel	Martinez	987 SW Sunset Blvd Phoenix	Arizona	rachel	Delete
9	Sarah	Ford	10544 N 8th St Beaverton	Oregon	sarah	Delete
16	Bob	Baker	88088 Wilshire Blvd Los Angeles	California	bob	Delete
17	Matt	Moore	3012 Wilshire Blvd Los Angeles	California	matt	Delete
18	Brook	Baker	8088 Wilshire Blvd Los Angeles	California	brook	Delete

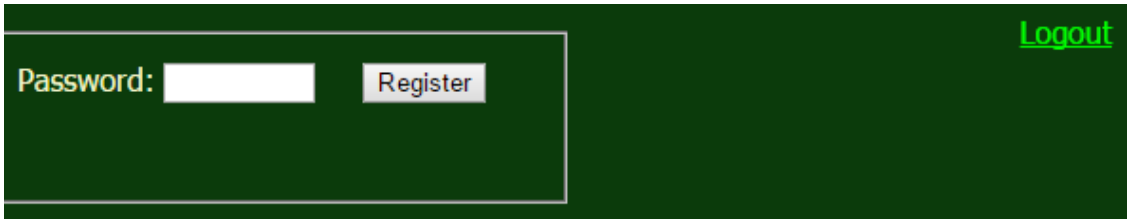
You can add or delete customers.

If you check "Show Password" and click "Pwd" button, each customer's password will be displayed along with other information.



Using the password information, you can log in as an existing customer after you log out as the administrator.

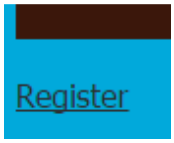
Log out link is located at the right top corner of the page.



As shown above, you can sort customers by "State" and then "Last Name" by ascending order.
Select "State", "Last Nm", and "A-Z" and then click "Sort".
"Default" option sorts the customers by IDs.

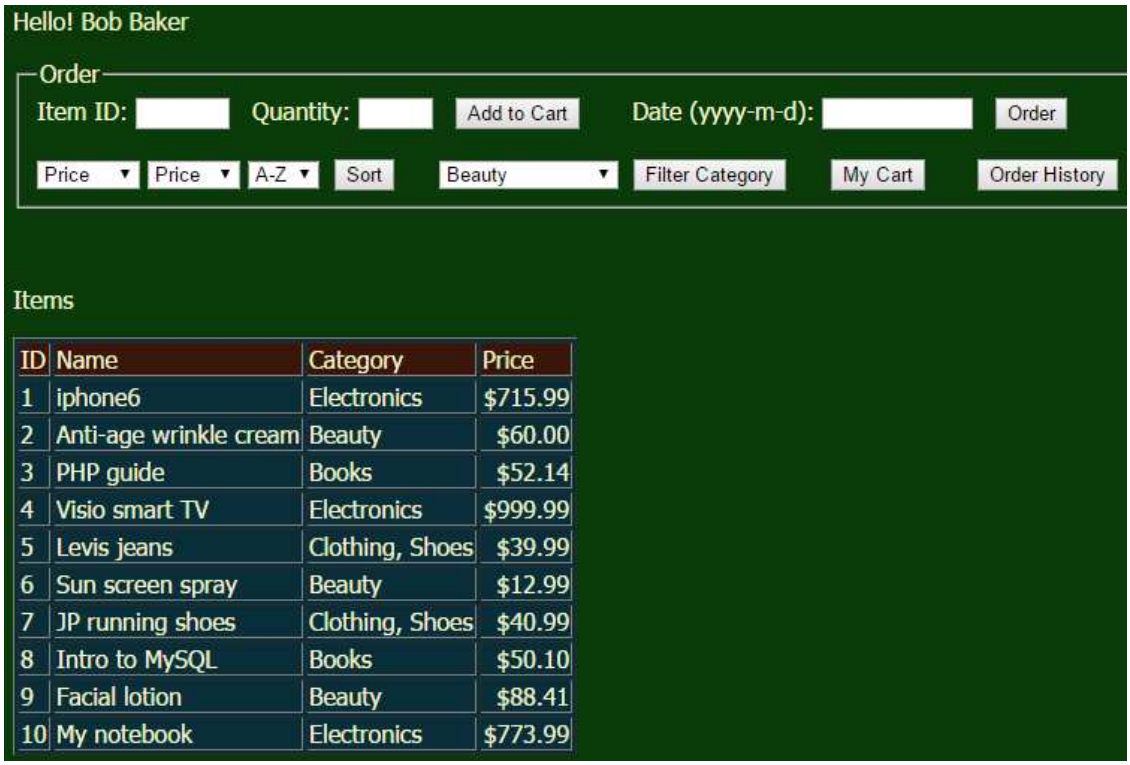
Please note that the administrator is not allowed to do shopping.

In order to register a new customer, from the login page click on "Register" link located at the left top corner of the page. This does not require log in.

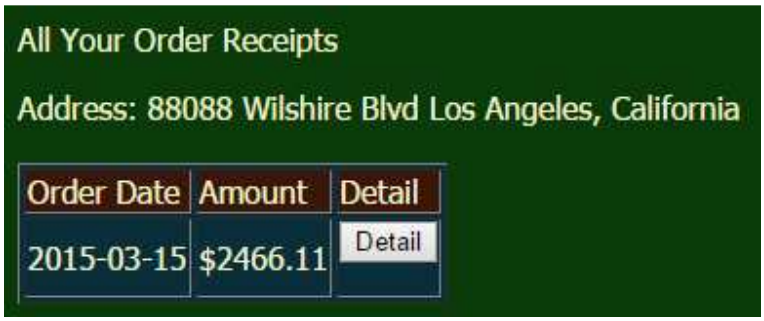


After filling out information, click "Register". This allows you to be registered and then you will be lead to the order page.

In order to make orders, you need to log in as a customer or register as a new customer.



As shown above, Bob Baker has logged in. He can now add items in his cart and then make an order. He can also view his order history.



Clicking "Detail" button allows you to view detailed order history information.

Your Order History Detail (Date: 2015-03-15 Total: $2466.11)

Name	Category	Price	Qnty	Sub Total
Facial lotion	Beauty	$88.41	5	$442.05
Intro to MySQL	Books	$50.10	2	$100.20
iphone6	Electronics	$715.99	2	$1431.98
JP running shoes	Clothing, Shoes	$40.99	12	$491.88

In order to view your cart anytime by clicking "My Cart".
Your items in your cart will be displayed.

You can sort items as the same way as sorting customers.
Select fields and then click "Sort".
"Default" option sorts the items by IDs.

You can also filter items by selecting "category", which gets populated dynamically.
Select the category you want and then click "Filter Category".

Once you are ready to make an order, make sure to enter order date (i.e. 2015-3-15) and then click "Order".

Order results will be displayed as shown below.

Ordered Items

Name	Category	Price	Quantity	Subtotal
Anti-age wrinkle cream	Beauty	$60.00	4	$240.00
Levis jeans	Clothing, Shoes	$39.99	3	$119.97

Order Receipt

First Name	Last Name	Address	State	Order Date	Amount
Amy	Williams	604 Goleta St Santa Barbara	California	2015-03-12	$359.97

Tables in the database

1) Table: users
   stores users information
   Fields:
   id (int): unique id of each customer
   fname (varchar): customer first name
   lname (varchar): customer last name
   address (varchar): customer address (i.e. 123 Main St. Portland)
   state (int): state ID (i.e. 37 for Oregon)
   username (varchar): customer username

   Constraint: FOREIGN KEY (state) REFERENCES states(id)
   FOREIGN KEY (username) REFERENCES members(username)

2) Table: states
   stores states name and population
   Fields:
   id (int): unique id of each state
   name (varchar): name of each state
   pop (int): population of each state

3) Table: things
   stores things, items information
   Fields:
   id (int): id of each item
   name (varchar): item name
   category (varchar): item category
   price (double): item price

4) Table: make_orders
   stores information of orders made by customers
   Fields:
   id (int): order id
   cid (int): customer id; foreign key; references id in the customers table
   odate (date): order date

   Constraint: FOREIGN KEY (cid) REFERENCES users(id)

5) Table: shopping_cart
   represents a shopping cart for all the customers
   Fields:
   cid (int): customer id
   iid (int): item id
   qnty (int): how many particular items placed in the cart
   isordered (boolean): one means this item(s) have been ordered; zero means not ordered yet

   Constraint: FOREIGN KEY (cid) REFERENCES users(id)
   FOREIGN KEY (iid) REFERENCES things(id)

6) Table: orders_things
   stores which items and how many of them were ordered
   Fields:
   oid (int): order id
   iid (int): item id
   qnty (int): how many items were ordered

   Constraint: FOREIGN KEY (oid) REFERENCES make_orders(id)
   FOREIGN KEY (iid) REFERENCES things(id)

7) Table: members
   stores memeber username and password information
   Fields:
   id (int): unique id
   username (varchar): member username
   password (varchar): member password

   Constraint: FOREIGN KEY (username) REFERENCES users(username)

Table Relationships

1) One to One

   One state ID is associated with one state name.
   Each state name belongs to only one state ID.
   State ID "37" is only related with state name "Oregon".

2) One to Many

   In a one-to-many relationship, each row in the related to table can be related to many rows in the relating table.
   Each customer may have zero, one or multiple orders.
   But an order can belong to only one customer.
   One customer can make many orders.
   A customer makes 2 orders (order #1 and #2).

3) Many to One

   Many orders can be made by one customer.
   The relationship between the users table and the make_orders table is a one-to-many relationship.
   Turning around, the relationship between the make_orders table and the users table is a many-to-one relationship.

4) Many to Many

   Each order can contain multiple items (things).
   Each item can also be in multiple orders.
   The orders_things table has only one purpose, and that is to create a many-to-many relationship between the orders and the items.

-- queries

```
--
-- Table structure for table `make_orders`
--

CREATE TABLE IF NOT EXISTS `make_orders` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `cid` int(10) NOT NULL,
  `odate` date NOT NULL,
  PRIMARY KEY (`id`),
  KEY `cid` (`cid`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=19 ;

-- --------------------------------------------------------

--
-- Table structure for table `members`
--

CREATE TABLE IF NOT EXISTS `members` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;


-- --------------------------------------------------------
```

```
--
-- Table structure for table `orders_things`
--

CREATE TABLE IF NOT EXISTS `orders_things` (
  `oid` int(10) NOT NULL,
  `iid` int(10) NOT NULL,
  `qnty` int(10) DEFAULT NULL,
  KEY `oid` (`oid`),
  KEY `iid` (`iid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `shopping_cart`
--

CREATE TABLE IF NOT EXISTS `shopping_cart` (
  `cid` int(10) NOT NULL,
  `iid` int(10) NOT NULL,
  `qnty` int(10) DEFAULT '1',
  `isordered` tinyint(1) DEFAULT '0',
  KEY `cid` (`cid`),
  KEY `iid` (`iid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------
```

```
--
-- Table structure for table `states`
--

CREATE TABLE IF NOT EXISTS `states` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(30) NOT NULL,
  `pop` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=52 ;

-- --------------------------------------------------------

--
-- Table structure for table `things`
--

CREATE TABLE IF NOT EXISTS `things` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(80) NOT NULL,
  `category` varchar(50) DEFAULT NULL,
  `price` decimal(5,2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

-- --------------------------------------------------------
```

```
--
-- Table structure for table `users`
--

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `fname` varchar(30) NOT NULL,
  `lname` varchar(50) NOT NULL,
  `address` varchar(100) NOT NULL,
  `state` int(2) unsigned NOT NULL,
  `username` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `state` (`state`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=19 ;
```