GROUP i

presents:

iBay

CMPS 460

Spring 2009

The BIG group project

Trey Alexander (txa4895)
Dallas Griffith (dlg5367)
Mark McKelvy (jmm0468)
Sayooj Valsan (sxv6633)

Table of Contents

CMPS 460
Spring 2009
Auction Website
Database Project

(Version 2 – Changes in bold and italics)

Project Description

You have been asked to create a database to manage a new website for auction listings. The system will provide the following features:

User Controls:

When a person visits the website, they should be able to either login with an existing username and password or register as a new user. Upon registration, a person should provide a username, password, real name, shipping address, phone number, birth date, email address, and credit card information (card type, card number, and expiration date)
Additionally, the registration date and last activity of each user should be recorded.

From some form of control panel, users should be able to change their password, shipping address, phone number, email address, and payment data. They may also add a brief description and pick an image to represent themselves. Image uploading is not required, but it should be simulated entering a filename in the database and displaying the file on the local machine at that location.

Viewing a profile:

Users should be able to view a profile of any registered user. ***The profile will include the username, picture, description, and a listing of other users' feedback about them.***

Listing an item:

Registered users should be able to create an item listing, providing a title, description, end date & time, shipping cost, shipping method, ***starting price***, and category for the item. Once again, an image for the listing should be able to be chosen from several already on the website. Allowing for image uploading is not necessary.

Browsing auction listings:

Registered users should be able to browse auction listings by categories (Art, Books, Clothing, Collectibles, Electronics, Entertainment, Jewelry, Sporting Goods, Toys). Each item will be assigned to only one category. There should also be methods to sort a listing page by title, seller, time remaining, and current ***bid***.

Bidding on an item:

Registered users should be able to place a bid on any open listings. A history of all current and past bids and the users that placed them should be kept for all items listed.

Leaving feedback:

When a buyer wins an auction, they should have the option to leave feedback for the item's seller. The feedback should consist of a rating from 1 (bad) to 10 (good) and a short description. The seller should similarly be able to leave feedback for the buyer. The buyer and seller should be able to leave feedback once at any time after the listing becomes closed.

Notifications:

If a user logs in and an auction they participated in had closed since their last visit, the website should display a notification box on the home page. If they were the winning bidder or the seller, this notification should be shown every time the user loads the home page until they leave feedback for that auction. The ***notification*** box itself should include the form for a user to leave feedback.

Also, the website should notify any users that are outbid in an auction in a similar manner. The notification should persist until either the auction ends or until the user chooses to remove the message.

All notifications should be displayed at the same time on the home page, with the most recent appearing closest to the top.

Admin Controls:

There should be a method by which an administrator control panel can be accessed for testing purposes. From there, a user should be able to view, edit, and delete any entity within the database.

Time:

The website's time does not have to advance on its own. A method to advance the time by minutes, hours, and days should be made available for testing purposes at the top of every page of the website. ***This will be accomplished by using 3 integer dialog boxes. Ex: 2 1 1 means 1 minute, 1 hour, 2 days ahead of the current date/time. For simplicity, time begins at day 0, hour 0, minute 0.***

System Requirements

Your system will be responsible for creating and maintaining the database. This includes the ability to add, change, and delete all data in the database (taking into account referential integrity). At a minimum, your system will include the following capabilities:

- Display screens for all data in the database.
- Add, change, and delete a registered user.
- Add, change, and delete an auction listing.
- Add, change, and delete a bid.
- Add, change, and delete feedback.
- Display a registered user's profile.
- Display a sortable list of auction items.
- Display an auction listing.
- Display a bid history for a particular auction
- Display a bid history for a particular user.
- Display feedback for a registered user.
- ***Display the bid history for a registered user***

Constraints

Sellers may not bid on their own auctions.

Users may not outbid themselves if they are already the high bidder.

Users may not place bids lower than the current high bid.

Project Requirements

Your system will use the MySQL database and will be written in PHP and all interaction with the system will be via a web browser. All programs will be created and maintained in your cs4601 class account.

Demonstrations will be scheduled for the last week of classes. A signup sheet will be posted on my office door sometime in April. All group members are required to be present during the demonstration.

Each member of the group must be prepared to demonstrate his/her portion of the system. You are responsible for providing test data adequate to demonstrate all features of the system. Any feature of the system that cannot be proven to function correctly will be counted as inoperative. Under NO circumstances will I (or you) "logon" to MySQL during the demonstration. There must be test data in the database that will allow all features of the system to be demonstrated.

The moral is – be finished, have the necessary test data loaded in the database, be ready to demonstrate, and have your documentation with you.

Documentation

The following documentation is required and due at the demonstration:

- A document describing the architecture of the application. This includes the database design (ER diagram), the data dictionary, and the function of all PHP scripts and a description of the interaction between them.
- A user's guide with sufficient detail to enable a new user to understand and use the system.
- A printout of the final version of the PHP scripts. The code must be well structured and appropriately commented. Each script must include the author's name, the date, and a certification that the code is the work of the author.

Grading

Obviously, you will be graded on the extent to which your system conforms to the requirements. You will also be graded on consistency, error checking, spelling, grammar, and system design. Remember, all members of your group will receive the same grade on the project.

There will be no extensions given. All material is due at the beginning of the presentation.

Please note that any material of an offensive nature (by my standards) found in your project will result in a grade of 0 on the project for all group members.

Insert ER diagram printout

Insert datadictionary.ods printout

PhP Function Description and Interaction

category.php

Handles category views, showing listings in a specific category and sorting.

No functions defined

---

category_pictures.php

Shows the user the available pictures for each category

No functions defined

---

common.php

Common include script which includes other utility scripts. Also has other common functions such as echoing the html headers and footers, html wrapping functions, and some formatting functions.

Call at the beginning of a script to output html header information including navbar
function echo_header ($dbinfo)

Call at the end of a script for closing html footer information
function echo_footer ($dbinfo)

Determine the current executing script
function current_script ()

Quick echo of a special div area
function echo_div ($name)

Quick close div
function end_div ()

Wrap text in a div
function div ($text, $style_name = "")

Wrap text in a span
function span ($text, $style_name)

Make a quick link
function href ($url, $text)

Make a quick link
function hreft ($url, $text, $target)

Quick html image code
function img ($path)

Quick html image code for local images
function local_img ($path)

Make the user go to a different page
function redirect ($url)

Get from the GET variable
function get ($key)

Get from the POST variable
function post ($key)

Script to make setting session variables easier.
function session_set ($key, $val)

Script to make getting session variables easier.
function session_get ($key)

Heading level 1
function h1 ($text)

Heading level 2
function h2 ($text)

Heading level 3
function h3 ($text)

An echo with a
 at the end
function cout ($str)

Format time to readable format
function format_time ($day, $hour, $minute, $prefix_zeros_to_day = false)

Pad time for sorting on the home page
function pad_time ($day, $hour, $minute, $priority = 0)

If time1 is older than time2, returns true
function is_older ($d1, $h1, $m1, $d2, $h2, $m2)

Strip quotes from a string (for form input)
function fix_quotes ($str)

Escape quotes in a string
function escape ($val)

Embed html link for javascript alert
function alert ($msg, $linktext)

dbinfo.php

Database information class, with useful utility functions for getting info out of the database

Call this function to start the session and initialize the variables
    function init ()

Is debug enabled
    function debug ()

Make a connection to the database returns true if successful or false if not
    function connect ()

Close the connection to the database usually call this at the end of a script
    function close ()

Make a query to the database and return the result
    function query ($q)

Get the link associated with the database not usually needed but added here for convenience
    function get_link ()

Call all update functions
    function update_all ()

Update user information in session variables (realname, is an admin)
    function update_user_info ()

Attempt to login a user. Will return true if successful and false if not. Additionally, will setup session vars for username and admin or not
    function login ($username, $password)

Used by admin and also when registering - can login as a user without password
    function login_as ($username)

Logout and save activity
    function logout ($is_delete = false)

Check if visitor is logged in
    function logged_in ()

Check if visitor is admin
    function is_admin ()

Get logged in username
function username ()

Get logged in in name of user
function realname ()

Update the time from the database into the session vars
function update_time ()

Will increment the site time by amount
function increment_site_time ($amount)

Get the current day, this is called after update_time();
function day ()

Get the current hour, this is called after update_time();
function hour ()

Get the current minute, this is called after update_time();
function minute ()

Save activity for the current user.
function save_activity ($activity_description)

Must be admin, save activity for another user - user must exist
function save_activity_for ($username, $activity_description)

Save a registration activity for the current user
function save_registration ($username)

Update all other bidder's notification status that they've been outbid, and also current_price for auction
function update_auction_before_bid ($title, $seller, $category, $end_day, $end_hour, $end_minute, $bid_amount)

Updates all item listing buyer information for closed auctions
function update_closed_item_listings ()

See if a user exists in the system
function user_exists ($username)

Get a realname for a username
function get_realname ($username)

Get the last logout time of a user
function get_lastlogout ($username)

Get the registration date of a user
function get_registration_date ($username)

Test if an auction already exists
function auction_exists ($title, $seller, $category, $end_day, $end_hour, $end_minute)

Test if a bid already exists

Get the winner of an auction
function get_winner ($title, $seller, $category, $end_day, $end_hour, $end_minute)

Find the current highest bidder for an auction
function get_highest_bidder ($title, $seller, $category, $end_day, $end_hour, $end_minute)

Get the number of bids for an auction
function get_num_bids ($title, $seller, $category, $end_day, $end_hour, $end_minute)

Gets the user description
function get_userdesc ($username)

Gets the picture associated with a user profile
function get_picture ($username)

---

form_common.php

Action: script to execute
Method: POST or GET

function form_begin ($action, $method)

Same as above but is names and has javascript onsubmit

function form_begin_n ($action, $method, $name, $on_submit)

End form

function form_end ()

Typical text input for a form

function text_input ($name, $value)

Typical text input for a form with size

function text_input_s ($name, $value, $size, $maxsize)

Typical text input for a form with size and readonly

function text_input_sr ($name, $value, $size, $maxsize)

Typical password input for a form

function password_input ($name, $value)

Typical password input for a form with size

function password_input_s ($name, $value, $size, $maxsize)

Typical submit input for a form

function submit_input ($value)

Hidden input for a form

function hidden_input ($name, $value)

Input for a form that has more freedom

function input ($name, $value, $size, $id, $type, $maxsize = "", $other_opts = "")

Form select which takes html string of options

function select ($name, $options)

Dynamic select used for javascript

function select_dyn ($name, $options, $javascript_func)

Html option used for a select dropdown

function option ($value, $display, $is_selected = "")

---

index.php

No functions defined

---

itemlisting.php

Functions for validation/etc..

Resets inputs
    function reset_inputs (thisform)

Shows fields that require validation as yellow
    function validate_required (field)

Sets fieldvar
    function field_set (fieldvar, field)

Appends a newline on text
    function append_with_newline (orig, append_text)

Ensures valid input
    function validate_form (thisform)

Gives image from image directory
    function local_img (file)

Formats image source
    function image_swap ()

A catch-all form for editing & creating auctions
    function item_listing_form ($mode = "new", $t = "", $s = "", $c = "", $d = -1, $h = -1, $m = -1)

Simple checking of the form data to meet basic requirements
    function verify_data ()

---

list_common.php

Functions for common HTML tags

Unordered Lists HTML helper tags

HTML ul begin tag
    function ul_begin ()

HTML ul end tag
    function ul_end ()

HTML ul wrap tag helper
    function ul_wrap ($list_elems)

HTML ul wrap tag helper
    function ul ($list_elems)

Unordered Lists HTML helper tags

HTML ol begin tag
    function ol_begin ()

HTML ol end tag
    function ol_end ()

HTML ol wrap tag helper
    function ol_wrap ($list_elems)

HTML ol wrap tag helper
    function ol ($list_elems)

List HTML helper tags

HTML li begin tag
    function li_begin ()

HTML li end tag
    function li_end ()

HTML li wrap tag helper
    function li_wrap ($list_elem)

HTML li wrap helper tag
    function li ($list_elem)

---

login.php

User login form page

No functions defined

---

logout.php

User logout processing page

No functions defined

---

profile.php

Profile page for editing & viewing user (and your own) profile.

Functions for image loading
function local_img (file)
function image_swap ()

Data validation
function verify_data ()

---

profile_pictures.php

Shows the user the available pictures for a profile

No functions defined

---

registration.php

Handles registration related things such as new users and editing your current registration, as well as view registered users (admin).

Resets inputs
function reset_inputs (thisform)

Shows fields that require validation as yellow
function validate_required (field)

Sets fieldvar
	function field_set (fieldvar, field)

Appends newline on text
	function append_with_newline (orig, append_text)

Validates email field
	function validate_email (field)

String replacement
	function trim (string)

Field validation
	function validate_int (field, length)

Form validation
	function validate_form (thisform)

Main registration form - kind of catch-all for editing and creating registrations
	function registration_form ($user = "")

Checking of the form data to meet basic requirements
	function verify_data ()

---

table_common.php

Table related common functions/class (for styles)

Initialize to use certain style
	function init ($main_class_)

Set to use certain style
	function set_main_class ($main_class_)

Get which style it is using
	function get_main_class ()

Begin the html table
	function table_begin ()

End the html table
	function table_end ()

Begin the html table head
	function table_head_begin ($custom_class = "")

End the html table head
function table_head_end ()

Begin the html table body
function table_body_begin ($custom_class = "")

End the html table body
function table_body_end ()

HTML table row
function tr ($s, $custom_class = "")

HTML table row begin
function tr_begin ($custom_class = "")

HTML table row end
function tr_end ()

HTML table data
function td ($s, $custom_class = "")

HTML table data which spans several columns
function td_span ($s, $custom_class = "", $span = 0, $align = "")

---

viewtable.php

Helpful page for administrator to see the data in the db.
function display_message ()

insert user manaul

category.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Handles category views, showing listings in a specific category and sorting.




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);
$view = get ("view");
$sortby = get ("sortby");

// Redirect if user not logged in
if (!$dbinfo->logged_in ())
        redirect ("index.php");

if ($dbinfo->logged_in ())
{
        // Default view shows user a dropdown
        if (empty ($view))
        {
                echo h3 ("All Categories");
                echo_div ("scriptstatus");
                echo href (" itemlisting.php?mode=new", "Create a New Item Listing");
                end_div ();

                cout ("Select a category:");
                $options = "";
                $options = $options.option ("Art", "Art");
                $options = $options.option ("Books", "Books");
                $options = $options.option ("Clothing", "Clothing");
```

```
                $options = $options.option ("Collectibles", "Collectibles");
                $options = $options.option ("Electronics", "Electronics");
                $options = $options.option ("Entertainment", "Entertainment");
                $options = $options.option ("Jewelry", "Jewelry");
                $options = $options.option ("Sporting Goods", "Sporting Goods");
                $options = $options.option ("Toys", "Toys");
                echo form_begin ("$current_script?", "get");
                echo select ("view", $options);
                echo submit_input ("Select");

        }
        else // show category specific listings
        {

                echo h3 ("$view");
                echo_div ("scriptstatus");
                echo href ("itemlisting.php?mode=new", "Create a New Item Listing");
                echo "<br/>Sort by: ";
                echo href ("category.php?view=$view&sortby=time_remaining", "Time Remaining");
                echo " | ";
                echo href ("category.php?view=$view&sortby=title", "Title");
                echo " | ";
                echo href ("category.php?view=$view&sortby=seller", "Seller");
                echo " | ";
                echo href ("category.php?view=$view&sortby=current_bid", "Current Bid (Lowest at
Top)");
                end_div ();

                $day = $dbinfo->day ();
                $hour = $dbinfo->hour ();
                $minute = $dbinfo->minute ();
                $query = "select title, seller, end_day, end_hour, end_minute, current_price from
item_listing
where  category = '$view'
AND   ((end_day > $day)
        OR (end_day = $day AND end_hour > $hour)
        OR (end_day = $day AND end_hour = $hour AND end_minute >= $minute))";

                if ($sortby == "title")
                {
                        $query = $query." order by title";
                }
                else if ($sortby == "seller")
                {
                        $query = $query." order by seller";
                }

                else if ($sortby == "current_bid")
                {
```

```
                $query = $query." order by current_price";
        }
        else // ($sortby == "time_remaining")
        {
                $query = $query." order by end_day,
end_hour, end_minute";
        }

        // Run the query
        $results_id = mysql_query($query);
        if($results_id)
        {
                if (mysql_num_rows($results_id) == 0)
                {
                        echo "No data.";
                }
                else
                {
                        $table = new table_common_t ();
                        $table->init ("category_listing");

                        echo $table->table_begin ();
                        echo $table->table_head_begin ();
                        echo $table->tr ($table->td_span ("Auctions available for bid
in \"$view\"", "", 6));
                        echo $table->tr ($table->td ("Title").
                                        $table->td ("Seller").
                                        $table->td ("Closing Time").
                                        $table->td ("Current Price"));
                        echo $table->tr_end ();
                        echo $table->table_head_end ();

                        echo $table->table_body_begin ();
                        while (list($title, $seller, $end_day, $end_hour,$end_min,
                                $curr_price) = mysql_fetch_row($results_id))
                        {
                                echo $table->tr ($table->td (href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$view&end_day=$end_day&end_hour=$end_hour
&end_minute=$end_min", $title)).
                                                $table->td (href ("profile.php?
mode=view&username=$seller", $dbinfo->get_realname ($seller))).
                                                $table->td (format_time ($end_day,
                                                        $end_hour, $end_minute)).
                                        $table->td ("\$$curr_price"));

                        }
                        echo $table->table_body_end ();

                        echo $table->table_end ();
```

```
                        echo "<br><br>";

                    }
                }

                else if ($dbinfo->debug ())
                {
                    // Display the query and the MySQL error message
                    print "<br><br>QUERY FAILED !!! <br><br>QUERY = $query <br>
            <br>ERROR = ";
                    die (mysql_error());
                }
        }


}

echo_footer ($dbinfo);

?>
```

category_picture.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Shows the user the available pictures for each category




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);
$current_script = current_script ();
$mode = get ("mode");

if (empty ($mode)) // dropdown to select category
{
        cout ("Listing of avialable pictures.");
        cout ("Pick a category:");

        $options = "";
        $options = $options.option ("Art", "Art");
        $options = $options.option ("Books", "Books");
        $options = $options.option ("Clothing", "Clothing");
        $options = $options.option ("Collectibles", "Collectibles");
        $options = $options.option ("Electronics", "Electronics");
        $options = $options.option ("Entertainment", "Entertainment");
        $options = $options.option ("Jewelry", "Jewelry");
        $options = $options.option ("Sporting Goods", "Sporting Goods");
        $options = $options.option ("Toys", "Toys");
        echo form_begin ("$current_script?mode=view", "post");
        echo select ("category", $options);
        echo submit_input ("Select");
```

```
}
else // assume 3 pictures per category
{
	$file = str_replace (" ", "_", strtolower (post ("category")));
	for ($i = 1; $i <= 3; $i++)
	{
		cout (local_img ("$file$i.jpg"));
		cout ("$file$i.jpg");
		cout ("");
	}
}

echo_footer ($dbinfo);

?>
```

common.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Common include script which includes other utility scripts. Also has other
// common functions such as echoing the html headers and footers, html
// wrapping functions, and some formatting functions.




include_once ("dbinfo.php");
include_once ("form_common.php");
include_once ("table_common.php");
include_once ("list_common.php");

// Call at the beginning of a script to output html header information including
// navbar
function echo_header ($dbinfo)
{
        $dbinfo->init ();
        if (!$dbinfo->connect ())
                die ("Couldn't connect to database.");
        $dbinfo->update_all ();
        $site_time = post ("site_time");
        if (!empty ($site_time))
        {
                $post_site_time = post ("site_time");
                $post_curr_day = strtok ($post_site_time, " ");
                $post_curr_hour = strtok (" ");
                $post_curr_minute = strtok (" ");
                $user = $dbinfo->username ();
                $formatted_time = format_time ($post_curr_day, $post_curr_hour, $post_curr_minute);
                $dbinfo->query ("insert into user_activity values (
'$user', $post_curr_day, $post_curr_hour, $post_curr_minute, 'Updated website time to
```

```
$formatted_time')");
                $dbinfo->update_all ();
        }

        echo <<<HEREDOC
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>CMPS 460 DB Project</title>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>

HEREDOC;
        echo_div ("container");

        // HEADER AREA
        echo_div ("header");
        echo "iBay";
        end_div ();

        // STATUS AREA
        echo_div ("status");
        $day = $dbinfo->day () + 0;
        $hour = $dbinfo->hour ();
        $minute = $dbinfo->minute ();
        if ($dbinfo->logged_in ())
        {
                $realname = $dbinfo->realname ();
                $username = $dbinfo->username ();
                $logoutlink = href ("logout.php", "Logout");
                cout ("Welcome, $realname. [$username] $logoutlink");
        }
        else
        {
                $loginlink = href ("login.php", "Login");
                $reglink= href ("registration.php", "Register");
                cout ("Welcome! $loginlink or $reglink");
        }
        echo ("Day: $day, Time: $hour:$minute");
        if ($dbinfo->is_admin () && current_script () == "index.php")
        {
                $curr_day = $dbinfo->day ();
                $curr_hour = $dbinfo->hour ();
                $options = "";
                for ($i = $curr_day; $i <= $curr_day + 7; $i++)
```

```
                {
                        for ($j = ($i == $curr_day)? $curr_hour + 1 : 0; $j < 24; $j++)
                        {
                                $options = $options.option ("$i $j 0", format_time ($i, $j, 0));
                        }
                }
                echo form_begin ("$current_script", "post");
                echo select ("site_time", $options);
                echo submit_input ("Do it").alert ("This box allows you to advance the website time.",
"?");
                echo form_end ();
//              echo "<br/>[Box to increase the time]";
        }
        end_div ();

        // NAVIGATION AREA
        echo_div ("navbar");
        if ($dbinfo->logged_in ())
        {
                if ($dbinfo->is_admin ())
                        $nav = "Admin Panel<br/>";
                else
                        $nav = "User Panel<br/>";

/*              if (current_script () != "index.php") */
                $nav = $nav.li_wrap (href ("index.php", "Home"));
                $nav = $nav.li_wrap (href ("registration.php?mode=view", "Account"));
                $nav = $nav.li_wrap (href ("profile.php", "Profile"));
                $nav = $nav.li_wrap (href ("profile.php?mode=browse", "All Profiles"));
                if ($dbinfo->is_admin ())
                        $nav = $nav.li_wrap (href ("registration.php?mode=browse", "All Accounts"));

                $nav = $nav.li_wrap (href ("category.php", "All Categories"));
                $nav = $nav.li_wrap (href ("category.php?view=Art", " - Art"));
                $nav = $nav.li_wrap (href ("category.php?view=Books", " - Books"));
                $nav = $nav.li_wrap (href ("category.php?view=Clothing", " - Clothing"));
                $nav = $nav.li_wrap (href ("category.php?view=Collectibles", " - Collectibles"));
                $nav = $nav.li_wrap (href ("category.php?view=Electronics", " - Electronics"));
                $nav = $nav.li_wrap (href ("category.php?view=Entertainment", " - Entertainment"));
                $nav = $nav.li_wrap (href ("category.php?view=Jewelry", " - Jewelry"));
                $nav = $nav.li_wrap (href ("category.php?view=Sporting Goods", " - Sporting
Goods"));
                $nav = $nav.li_wrap (href ("category.php?view=Toys", " - Toys"));
                $nav = $nav.li_wrap (href ("itemlisting.php?mode=new", "Create Auction"));

                if ($dbinfo->is_admin ())
                {
                        $nav = $nav.li_wrap ("Database:");
                        $nav = $nav.li_wrap (href ("viewtable.php?name=user", "User Table"));
```

```
                        $nav = $nav.li_wrap (href ("viewtable.php?name=user_activity", "Activity
Table"));
                        $nav = $nav.li_wrap (href ("viewtable.php?name=item_listing", "Auction
Table"));
                        $nav = $nav.li_wrap (href ("viewtable.php?name=bids_on", "Bid Table"));
                        $nav = $nav.li_wrap (href ("viewtable.php?name=reload", "Reload DB"));
                }
                echo ul_wrap ($nav);
        }
        else
        {
                $nav = "Visitor<br/>";
                if (current_script () != "index.php")
                        $nav = $nav.li_wrap (href ("index.php", "Home"));
                if (current_script () != "login.php")
                        $nav = $nav.li_wrap (href ("login.php", "Login"));
                if (current_script () != "registration.php")
                        $nav = $nav.li_wrap (href ("registration.php", "Register"));
                echo ul_wrap ($nav);
        }
        end_div ();
        echo_div ("content");
        echo "<!-- END HEADER -->\n";
}

// Call at the end of a script for closing html footer information
function echo_footer ($dbinfo)
{
        echo "<!-- FOOTER -->\n";
        end_div ();
        end_div ();
        echo <<<HEREDOC
</body>
</html>

HEREDOC;
        $dbinfo->close ();
}

// determine the current executing script
function current_script ()
{
        return basename ($_SERVER["SCRIPT_NAME"]);
}

// Quick echo of a special div area
function echo_div ($name)
{
        if (empty ($name))
```

```
                echo "<div>\n";
        else
                echo "<div id=\"$name\">\n";
}

// Quick close div
function end_div ()
{
        echo "</div>\n";
}

// Wrap text in a div
function div ($text, $style_name = "")
{
        if (!empty ($style_name))
                $style_name = "id=\"$style_name\"";
        return "<div $style_name>\n$text\n</div>\n";
}

// Wrap text in a span
function span ($text, $style_name)
{
        if (!empty ($style_name))
                $style_name = "id=\"$style_name\"";
        return "<span $style_name>\n$text\n</span>\n";
}

// make a quick link
function href ($url, $text)
{
        return "<a href=\"$url\">$text</a>";
}

// make a quick link
function hreft ($url, $text, $target)
{
        return "<a href=\"$url\" target=\"$target\">$text</a>";
}

// Quick html image code
function img ($path)
{
        return "<img src=\"$path\">";
}

// Quick html image code for local images
function local_img ($path)
{
        return "<img id=dynimg src=\"images/$path\">";
```

```
}

// Make the user go to a different page
function redirect ($url)
{
//	header ("Location: $url");
	echo <<<HEREDOC
<script type="text/javascript">
<!--
window.location = "$url"
//-->
</script>
HEREDOC;

}

// Get from the GET variable
function get ($key)
{
	if (isset ($_GET[$key]))
		$val = $_GET[$key];
	else
		$val = "";
	return $val;
}

// Get from the POST variable
function post ($key)
{
	if (isset ($_POST[$key]))
		$val = $_POST[$key];
	else
		$val = "";
	return $val;
}

// Script to make setting session variables easier.
function session_set ($key, $val)
{
	$key = "tmp_$key";
	$_SESSION[$key] = $val;
}

// Script to make getting session variables easier.
function session_get ($key)
{
	$key = "tmp_$key";
	return $_SESSION[$key];
}
```

```
// Heading level 1
function h1 ($text)
{
	return "<h1>$text</h1>\n";
}

// Heading level 2
function h2 ($text)
{
	return "<h2>$text</h2>\n";
}

// Heading level 3
function h3 ($text)
{
	return "<h3>$text</h3>\n";
}

// an echo with a <br/> at the end
function cout ($str)
{
	echo "$str"."<br/>\n";
}

// Format time to readable format
function format_time ($day, $hour, $minute, $prefix_zeros_to_day = false)
{
	$day += 0;
	$hour += 0;
	$minute += 0;

	if ($minute > 59)
	{
		$minute -= 60;
		$hour++;
	}

	if ($hour > 23)
	{
		$hour -= 24;
		$day++;
	}

	if ($minute < 10)
	{
		$minute = "0".$minute;
	}
```

```
	if ($hour < 10)
	{
		$hour = "0".$hour;
	}

	if ($prefix_zeros_to_day)
	{
		if ($day < 10)
		{
			$day = "0000".$day;
		}
		else if ($day < 100)
		{
			$day = "000".$day;
		}
		else if ($day < 1000)
		{
			$day = "00".$day;
		}
		else if ($day < 10000)
		{
			$day = "0".$day;
		}
	}
	return "Day: $day, Time: $hour:$minute";
}
// Pad time for sorting on the home page
function pad_time ($day, $hour, $minute, $priority = 0)
{
	if ($minute > 59)
	{
		$minute -= 60;
		$hour++;
	}

	if ($hour > 23)
	{
		$hour -= 24;
		$day++;
	}

	if ($minute < 10)
	{
		$minute = "0".$minute;
	}

	if ($hour < 10)
	{
		$hour = "0".$hour;
```

```
        }

        if ($day < 10)
        {
                $day = "0000".$day;
        }
        else if ($day < 100)
        {
                $day = "000".$day;
        }
        else if ($day < 1000)
        {
                $day = "00".$day;
        }
        else if ($day < 10000)
        {
                $day = "0".$day;
        }

        return "$day$hour$minute$priority ";
}

// If time1 is older than time2, returns true
function is_older ($d1, $h1, $m1, $d2, $h2, $m2)
{
        if (($d1 < $d2) ||
           ($d1 == $d2 && $h1 < $h2) ||
           ($d1 == $d2 && $h1 == $h2 && $m1 < $m2))
                return true;
        return false;
}

// Strip quotes from a string (for form input)
function fix_quotes ($str)
{
        return str_replace ("\'", "", $str);
}

// Escape quotes in a string
function escape ($val)
{
        return mysql_real_escape_string ($val);
}

// Embed html link for javascript alert
function alert ($msg, $linktext)
{
        return "<a href=\"javascript:alert ('$msg')\">$linktext</a>";
}
```

?>
dbinfo.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Database information class, with useful utility functions for getting info
// out of the database




class dbinfo_t
{
	var $host; // database host
	var $user; // database user
	var $pass; // database password
	var $dbname; // database name
	var $dblink; // connection link
	var $debug; // debug mode enabled

	// Call this function to start the session
	// and initialize the variables
	function init ()
	{
		session_start ();
		$this->host = "calvados.ucs.louisiana.edu";
		$this->user = "cs4601i";
		$this->pass = "foursixty";
		$this->dbname = "cs4601_i";
		$this->admin = 0;
		$this->debug = true;
	}

	// Is debug enabled
```

```
function debug ()
{
	return $this->debug;
}

// Make a connection to the database
// returns true if successful or false if not
function connect ()
{
	$success = 1;
	$this->dblink = mysql_connect ($this->host,
				$this->user,
				$this->pass)
		or ($success = 0);
	if (!$success)
		return $success;

	mysql_select_db ($this->dbname)
		or ($success = 0);
	return $success;
}

// Close the connection to the database
// usually call this at the end of a script
function close ()
{
	mysql_close ($this->dblink);
}

// Make a query to the database and return the result
function query ($q)
{
	if ($this->debug)
		$result = mysql_query ($q) or die ('BAD QUERY: '.mysql_error ());
	else
		$result = mysql_query ($q);
	return $result;
}

// Get the link associated with the database
// not usually needed but added here for convenience
function get_link ()
{
	return $this->dblink;
}

// Call all update functions
function update_all ()
{
```

```
                $this->update_time ();
                $this->update_user_info ();
                $this->update_closed_item_listings ();
        }

        // Update user information in session variables (realname, is an admin)
        function update_user_info ()
        {
                $username = $this->username ();
                $result = $this->query ("select realname, is_admin from user
where username = '$username'");
                $row = mysql_fetch_assoc ($result);
                $_SESSION['Admin'] = $row['is_admin'];
                $_SESSION['Realname'] = $row['realname'];
                mysql_free_result ($result);
        }

        // Attempt to login a user
        // Will return true if successful and false if not.
        // Additionally, will setup session vars for username
        // and admin or not
        function login ($username, $password)
        {
                $result = $this->query ("select username, realname, is_admin from user
where username = '$username' and password = '$password'");
                if (mysql_num_rows ($result) == 0)
                {
                        unset ($_SESSION['Username']);
                        unset ($_SESSION['Realname']);
                        unset ($_SESSION['Admin']);
                        mysql_free_result ($result);
                        return false;
                }
                else
                {
                        $row = mysql_fetch_assoc ($result);
                        $_SESSION['Username'] = $row['username'];
                        $_SESSION['Admin'] = $row['is_admin'];
                        $_SESSION['Realname'] = $row['realname'];
                        mysql_free_result ($result);
                        $this->save_activity ("Logged In");
                        return true;
                }
        }

        // Used by admin and also when registering - can login as a user without password
        function login_as ($username)
        {
                $result = $this->query ("select username, realname, is_admin from user
```

```
where username = '$username'");
		if (mysql_num_rows ($result) == 0)
		{
			unset ($_SESSION['Username']);
			unset ($_SESSION['Realname']);
			unset ($_SESSION['Admin']);
			mysql_free_result ($result);
			return false;
		}
		else
		{
			$row = mysql_fetch_assoc ($result);
			$_SESSION['Username'] = $row['username'];
			$_SESSION['Admin'] = $row['is_admin'];
			$_SESSION['Realname'] = $row['realname'];
			mysql_free_result ($result);
			$this->save_activity ("Logged In");
			return true;
		}
	}

	// Logout and save activity
	function logout ($is_delete = false)
	{
		if (!$is_delete)
			$this->save_activity ("Logged Out");
		unset ($_SESSION['Username']);
		unset ($_SESSION['Realname']);
		unset ($_SESSION['Admin']);
		session_destroy ();
		if (!$is_delete)
			header ("Location: index.php");
	}


	// Check if visitor is logged in
	function logged_in ()
	{
		return isset ($_SESSION) && isset ($_SESSION['Username']);
	}

	// Check if visitor is admin
	function is_admin ()
	{
		return $this->logged_in () && $_SESSION['Admin'] == 1;
	}

	// Get logged in username
	function username ()
```

```
	{
		return $_SESSION['Username'];
	}

	// Get logged in in name of user
	function realname ()
	{
		return $_SESSION['Realname'];
	}

	// Update the time from the database into the session vars
	function update_time ()
	{
		$result = $this->query ("select day,hour,minute from user_activity
order by day desc, hour desc, minute desc limit 1");
		$row = mysql_fetch_assoc ($result);
		$day = $row['day'];
		$hour = $row['hour'];
		$minute = $row['minute'] + 1;

		if ($minute > 59)
		{
			$minute -= 60;
			$hour++;
		}

		if ($hour > 23)
		{
			$hour -= 24;
			$day++;
		}

		if ($minute < 10)
		{
			$minute = "0".$minute;
		}

		if ($hour < 10)
		{
			$hour = "0".$hour;
		}

		if ($day < 10)
		{
			$day = "0000".$day;
		}
		else if ($day < 100)
		{
			$day = "000".$day;
```

```
		}
		else if ($day < 1000)
		{
			$day = "00".$day;
		}
		else if ($day < 10000)
		{
			$day = "0".$day;
		}

		$_SESSION['Day'] = $day;
		$_SESSION['Hour'] = $hour;
		$_SESSION['Minute'] = $minute;
		mysql_free_result ($result);
	}

	// Will increment the site time by amount
	function increment_site_time ($amount)
	{
		if ($amount < 1)
			return false;
		$amount -= 1;
		$this->update_time ();
		$user = $this->username ();
		$day = $this->day ();
		$hour = $this->hour ();
		$minute = $this->minute () + $amount;
		if ($minute > 59)
		{
			$minute -= 60;
			$hour++;
		}
		if ($hour > 23)
		{
			$hour -= 24;
			$day++;
		}
		$this->query ("insert into user_activity values ('$user', $day, $hour, $minute, 'Current
Time')");
		return true;
	}

	// Get the current day, this is called after update_time();
	function day ()
	{
		return $_SESSION['Day'];
	}

	// Get the current hour, this is called after update_time();
```

```
	function hour ()
	{
		return $_SESSION['Hour'];
	}

	// Get the current minute, this is called after update_time();
	function minute ()
	{
		return $_SESSION['Minute'];
	}

	// Save activity for the current user.
	function save_activity ($activity_description)
	{
		$this->update_time ();
		$user = $this->username ();
		$day = $this->day ();
		$hour = $this->hour ();
		$minute = $this->minute ();
		$activity_description = mysql_real_escape_string ($activity_description);

		$this->query ("insert into user_activity values ('$user', $day, $hour, $minute,
'$activity_description')");
		return true;
	}

	// Must be admin, save activity for another user - user must exist
	function save_activity_for ($username, $activity_description)
	{
		if (!$this->is_admin ())
			return false;

		$this->update_time ();
		$user = $username;
		$day = $this->day ();
		$hour = $this->hour ();
		$minute = $this->minute ();
		$activity_description = mysql_real_escape_string ($activity_description);

		// determine if $username is a valid user
		$result = $this->query ("select count(*) as count from user where username =
'$username'");
		if (mysql_num_rows ($result) == 0)
			return false;
		mysql_free_result ($result);

		$this->query ("insert into user_activity values ('$user', $day, $hour, $minute,
'$activity_description')");
	}
```

```
	// Save a registration activity for the current user
	function save_registration ($username)
	{
		$this->update_time ();
		$user = $username;
		$day = $this->day ();
		$hour = $this->hour ();
		$minute = $this->minute ();
		$this->query ("insert into user_activity values ('$user', $day, $hour, $minute,
'Registered')");
		return true;
	}

	// Update all other bidder's notification status that they've been outbid, and also current_price
for auction
	function update_auction_before_bid ($title, $seller, $category, $end_day, $end_hour,
$end_minute, $bid_amount)
	{
		$this->query ("update bids_on set
display_notification = 'y'
where  display_notification = 'n'
AND   item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute");
		$this->query ("update item_listing set
current_price = $bid_amount
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");
	}

	// Updates all item listing buyer information for closed auctions
	function update_closed_item_listings ()
	{
		$day = $this->day ();
		$hr = $this->hour ();
		$min = $this->minute ();
		$result1 = $this->query ("select title, seller, category, end_day, end_hour, end_minute
from item_listing
where  (buyer = '' OR buyer = 'None')
AND   (end_day < $day
	OR (end_day = $day
```

```
                AND end_hour < $hr)
        OR (end_day = $day
                AND end_hour = $hr
                AND end_minute <= $min)
        )");
                if (mysql_num_rows ($result1) > 0)
                {
                        while ($row = mysql_fetch_assoc ($result1))
                        {
                                $title = $row['title'];
                                $seller = $row['seller'];
                                $category = $row['category'];
                                $endday = $row['end_day'];
                                $endhr = $row['end_hour'];
                                $endmin = $row['end_minute'];
                                $result2 = $this->query ("select username, bid_amount from bids_on
where  item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $endday
AND   item_end_hour = $endhr
AND   item_end_minute = $endmin
order by bid_amount desc limit 1");
                                // no buyer
                                if (mysql_num_rows ($result2) == 0)
                                {
                                        $this->query ("update item_listing set
buyer = 'None',
current_price = starting_price
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $endday
AND   end_hour = $endhr
AND   end_minute = $endmin");
                                }
                                // buyer
                                else
                                {
                                        $row2 = mysql_fetch_assoc ($result2);
                                        $buyer = $row2['username'];
                                        $price = $row2['bid_amount'];
                                        $this->query ("update item_listing set
buyer = '$buyer',
current_price = $price
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $endday
```

```
AND   end_hour = $endhr
AND   end_minute = $endmin");
                                }
                                mysql_free_result ($result2);
                        }
                        mysql_free_result ($result1);
                }
        }

        // See if a user exists in the system
        function user_exists ($username)
        {
                $result = $this->query ("select username from user where username = '$username'");
                if (mysql_num_rows ($result) == 0)
                {
                        mysql_free_result ($result);
                        return false;
                }
                else
                {
                        mysql_free_result ($result);
                        return true;
                }
        }

        // Get a realname for a username
        function get_realname ($username)
        {
                $result = $this->query ("select realname from user where username = '$username'");
                if (mysql_num_rows ($result) == 0)
                        return "ERROR";
                else
                {
                        $row = mysql_fetch_assoc ($result);
                        return $row['realname'];
                }
        }

        // Get the last logout time of a user
        function get_lastlogout ($username)
        {
                $result = $this->query ("select day, hour, minute from user_activity
where  username = '$username'
AND   activity = 'Logged Out'
order by day desc, hour desc, minute desc
limit 1");
                if (mysql_num_rows ($result) == 0)
                        return array (-1, -1, -1);
                else
```

```
		{
			list ($day, $hr, $min) = mysql_fetch_row ($result);
			return array ($day, $hr, $min);
		}
	}

	// Get the registration date of a user
	function get_registration_date ($username)
	{
		$result = $this->query ("select day, hour, minute from user_activity
where  username = '$username'
AND   activity = 'Registered'
order by day desc, hour desc, minute desc
limit 1");
		if (mysql_num_rows ($result) == 0)
			return array (-1, -1, -1);
		else
		{
			list ($day, $hr, $min) = mysql_fetch_row ($result);
			return array ($day, $hr, $min);
		}
	}

	// Test if an auction already exists
	function auction_exists ($title, $seller, $category, $end_day, $end_hour, $end_minute)
	{
		$result = $this->query ("select seller from item_listing
where title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");
		if (mysql_num_rows ($result) == 0)
		{
			mysql_free_result ($result);
			return false;
		}
		else
		{
			mysql_free_result ($result);
			return true;
		}
	}

	// Test if a bid already exists
	function bid_exists ($user, $title, $seller, $category, $end_day, $end_hour, $end_minute,
				$bid_day, $bid_hour, $bid_minute)
	{
```

```
		$result = $this->query ("select username from bids_on
where  username = '$user'
AND   item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute
AND   bid_day = $bid_day
AND   bid_hour = $bid_hour
AND   bid_minute = $bid_minute");
		if (mysql_num_rows ($result) == 0)
		{
			mysql_free_result ($result);
			return false;
		}
		else
		{
			mysql_free_result ($result);
			return true;
		}
	}

	// Get the winner of an auction
	function get_winner ($title, $seller, $category, $end_day, $end_hour, $end_minute)
	{
		$result = $this->query ("select buyer, realname, current_price from item_listing, user
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute
AND   (buyer != '' OR buyer != 'None')
AND   buyer = username");
		if (mysql_num_rows ($result) == 0)
			return array (-1, -1, -1);
		else
		{
			list ($username, $realname, $bid_amount) = mysql_fetch_row ($result);
			return array ($username, $realname, $bid_amount);
		}
	}

	// Find the current highest bidder for an auction
	function get_highest_bidder ($title, $seller, $category, $end_day, $end_hour, $end_minute)
	{
		$result = $this->query ("select username, bid_amount from bids_on
where  item_title = '$title'
```

```
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute
order by bid_amount desc");
                if (mysql_num_rows ($result) == 0)
                        return array ("None", 0);
                else
                {
                        list ($username, $bid_amount) = mysql_fetch_row ($result);
                        return array ($username, $bid_amount);
                }
        }

        // Get the number of bids for an auction
        function get_num_bids ($title, $seller, $category, $end_day, $end_hour, $end_minute)
        {
                $result = $this->query ("select count(*) from bids_on
where  item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute");
                list ($bid_count) = mysql_fetch_row ($result);
                return $bid_count;
        }

        //Begin - Changes made by Sayooj Valsan > # User Profile
        // Gets the user description
        function get_userdesc ($username)
        {
                $result = $this->query ("select description from user where username =  '$username'");
                $row = mysql_fetch_assoc ($result);
                return $row['description'];
        }
        //End - Changes made by Sayooj Valsan > # User Profile
        //Begin - Changes made by Sayooj Valsan <04-10-2009>
        function get_picture ($username)
        {
                $result = $this->query ("select picture from user where username = '$username'");
                $row = mysql_fetch_assoc ($result);
                return $row['picture'];
        }
        //End - Changes made by Sayooj Valsan <04-10-2009>

}
```

?>

form_common.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Form related common functions




// Action: script to execute
// Method: POST or GET
function form_begin ($action, $method)
{
	return "<form action=\"$action\" method=\"$method\">\n";
}

// Same as above but is names and has javascript onsubmit
function form_begin_n ($action, $method, $name, $on_submit)
{
	return "<form action=\"$action\" method=\"$method\" accept-charset=\"UTF-8\"
name=\"$name\" onsubmit=\"$on_submit\">\n";
}

// End form
function form_end ()
{
	return "</form>\n";
}

// Typical text input for a form
function text_input ($name, $value)
{
	return input ($name, $value, 20, "", "text");
```

```
}

// Typical text input for a form with size
function text_input_s ($name, $value, $size, $maxsize)
{
	return input ($name, $value, $size, "", "text", $maxsize);
}

// Typical text input for a form with size and readonly
function text_input_sr ($name, $value, $size, $maxsize)
{
	return input ($name, $value, $size, "", "text", $maxsize, "readonly");
}

// Typical password input for a form
function password_input ($name, $value)
{
	return input ($name, $value, 20, "", "password");
}

// Typical password input for a form with size
function password_input_s ($name, $value, $size, $maxsize)
{
	return input ($name, $value, $size, "", "password", $maxsize);
}

// Typical submit input for a form
function submit_input ($value)
{
	return input ("formsubmit", $value, "", "", "submit");
}

// Hidden input for a form
function hidden_input ($name, $value)
{
	return "<input type=\"hidden\" name=\"$name\" value=\"$value\">\n";
}

// Input for a form that has more freedom
function input ($name, $value, $size, $id, $type, $maxsize = "", $other_opts = "")
{
	if (!empty ($maxsize))
		$maxsize = "maxlength=\"$maxsize\"";
	if (empty ($size))
		return "<input name=\"$name\" $maxsize id=\"$id\" type=\"$type\" value=\"$value\"
$other_opts>\n";
	else
		return "<input name=\"$name\" $maxsize size=\"$size\" id=\"$id\" type=\"$type\"
value=\"$value\" $other_opts>\n";
```

```
}

// Form select which takes html string of options
function select ($name, $options)
{
	return "<select name=\"$name\">\n$options \n</select>";
}

// Dynamic select used for javascript
function select_dyn ($name, $options, $javascript_func)
{
	return "<select onChange=\"$javascript_func\" id=\"dyn$name\" name=\"$name\">\n$options \
n</select>";
}

// Html option used for a select dropdown
function option ($value, $display, $is_selected = "")
{
	if (!empty ($is_selected) && $value == $is_selected)
		return "<option selected=\"yes\" value=\"$value\">$display</option>\n";
	else
		return "<option value=\"$value\">$display</option>\n";
}
?>
```

index.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Main page user sees whether logged in to site or not. If logged in, shows
// activity and notifications and if admin, also shows time advance box




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);
$mode = get ("mode");
$current_script = current_script ();
$username = $dbinfo->username ();
$curr_day = $dbinfo->day ();
$curr_hour = $dbinfo->hour ();
$curr_minute = $dbinfo->minute ();

$x = 0;
$output;

// Remove an outbid notification
if ($dbinfo->logged_in () && $mode == "remove")
{
        $title = get ("t");
        $seller = get ("s");
        $category = get ("c");
        $end_day = get ("d");
        $end_hour = get ("h");
```

```
		$end_min = get ("m");
		$bid_day = get ("bd");
		$bid_hour = get ("bh");
		$bid_min = get ("bm");
		$dbinfo->query ("update bids_on set display_notification = 'c'
where  username = '$username'
AND   item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_min
AND   bid_day = $bid_day
AND   bid_hour = $bid_hour
AND   bid_minute = $bid_min");
}

// Save feedback
if ($dbinfo->logged_in () && $mode == "feedback")
{
		// Need to save feedback for user.
		$title = post ("title");
		$seller = post ("seller");
		$category = post ("category");
		$end_day = post ("end_day");
		$end_hour = post ("end_hour");
		$end_min = post ("end_min");
		$description = fix_quotes (post ("feedback"));
		$rating = post ("rating");
		$for = post ("for");
		$real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
		if ($for == "buyer")
		{
			$dbinfo->query ("update item_listing set sellerfeedbackforbuyer_description =
'$description'
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_min");
			$dbinfo->save_activity ("You left feedback for the winner of your
auction \"$real_title\"");
		}
		if ($for == "seller")
		{
			$dbinfo->query ("update item_listing set buyerfeedbackforseller_description =
```

```
'$description',
buyerfeedbackforseller_rating = $rating
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_min");
                $dbinfo->save_activity ("You left feedback for the seller of the auction \"$real_title\"");
        }
}

// The big mambo jambo
if ($dbinfo->logged_in ())
{
        // Get all feedback notifications which you are seller for
        $result = $dbinfo->query ("select * from item_listing
where  seller = '$username'
AND   (end_day < $curr_day
        OR (end_day = $curr_day
                AND end_hour < $curr_hour)
        OR (end_day = $curr_day
                AND end_hour = $curr_hour
                AND end_minute < $curr_minute))
AND   (buyer <> 'None')
order by end_day desc, end_hour desc, end_minute desc");
        $t = new table_common_t ();
        $t->init ("feedback");
        while (list ($title, $seller, $category, $end_day, $end_hour, $end_min,
                $description, $shipping_cost, $shipping_method, $starting_price,
                $current_price, $picture, $buyer, $buyer_fdbk, $buyer_fdbk_rating,
                $seller_fdbk)
            = mysql_fetch_row ($result))
        {
                $real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
                $winner_realname = href ("profile.php?mode=view&username=$buyer", $dbinfo-
>get_realname ($buyer));
                if (empty ($buyer_fdbk))
                {
                        $bfdbk = "$winner_realname has not left any feedback for you.<br/>";
                }
                else
                {
                        $bfdbk = "$winner_realname has given you a rating of \"$buyer_fdbk_rating\"
and has left the following feedback: \"$buyer_fdbk\"";
                }
```

```
                if (empty ($seller_fdbk))
                        $sfdbk = $t->table_begin ().$t->table_head_begin ().
                                $t->tr ($t->td ("Leave feedback for $winner_realname?")).
                                $t->table_head_end ().$t->table_body_begin ().
                                form_begin ("$current_script?mode=feedback", "post").
                                $t->tr ($t->td (text_input_s ("feedback", "", 50, 250).
                                                hidden_input ("title", $title).
                                                hidden_input ("seller", $seller).
                                                hidden_input ("category", $category).
                                                hidden_input ("end_day", $end_day).
                                                hidden_input ("end_hour", $end_hour).
                                                hidden_input ("end_min", $end_min).
                                                hidden_input ("for", "buyer")).
                                        $t->td_span (submit_input ("Save"), "", 1, "center")).
                                form_end ().
                                $t->table_body_end ().$t->table_end ();
                else
                        $sfdbk = "<br/>You left the following feedback: \"$seller_fdbk\".";

                $output[$x] = pad_time ($end_day, $end_hour, $end_min, 5).div (span (format_time
($end_day, $end_hour, $end_min), "time").
                                "Regarding your auction \"$real_title\" which has ended, $winner_realname
won with a high bid of \$$current_price.<br/>$bfdbk$sfdbk", "feedback");
                $x++;
        }
        mysql_free_result ($result);


        // Get all feedback notifications which you are buyer for
        $result = $dbinfo->query ("select * from item_listing
where  buyer = '$username'
order by end_day desc, end_hour desc, end_minute desc");
        while (list ($title, $seller, $category, $end_day, $end_hour, $end_min,
                        $description, $shipping_cost, $shipping_method, $starting_price,
                        $current_price, $picture, $buyer, $buyer_fdbk, $buyer_fdbk_rating,
                        $seller_fdbk)
                = mysql_fetch_row ($result))
        {
                $real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
                $seller_realname = href ("profile.php?mode=view&username=$seller", $dbinfo-
>get_realname ($seller));

                if (empty ($seller_fdbk))
                        $sfdbk = "$seller_realname has not left any feedback for you.<br/>";
                else
                        $sfdbk = "$seller_realname has left the following feedback: \"$seller_fdbk\"";
```

```
		if (empty ($buyer_fdbk))
		{
			$options = "";
			for ($i = 10; $i >= 0; $i--)
			{
				$options = $options.option ($i, $i, "");
			}
			$bfdbk = $t->table_begin ().$t->table_head_begin ().
				$t->tr ($t->td ("Leave feedback for $seller_realname?")).
				$t->table_head_end ().$t->table_body_begin ().
				form_begin ("$current_script?mode=feedback", "post").
				$t->tr ($t->td (text_input_s ("feedback", "", 50, 250).
						hidden_input ("title", $title).
						hidden_input ("seller", $seller).
						hidden_input ("category", $category).
						hidden_input ("end_day", $end_day).
						hidden_input ("end_hour", $end_hour).
						hidden_input ("end_min", $end_min).
						hidden_input ("for", "seller")).
					$t->td (select ("rating", $options)).
					$t->td_span (submit_input ("Save"), "", 1, "center")).
				form_end ().
				$t->table_body_end ().$t->table_end ();
		}
		else
		{
			$bfdbk = "<br/>You rated $seller_realname '$buyer_fdbk_rating' and left the
following feedback: \"$buyer_fdbk\".";
		}

		$output[$x] = pad_time ($end_day, $end_hour, $end_min, 5).div (span (format_time
($end_day, $end_hour, $end_min), "time").
			"Regarding the auction \"$real_title\" you have won with a high bid of \$
$current_price.<br/>$sfdbk$bfdbk", "feedback");
		$x++;
	}
	mysql_free_result ($result);

	// Get all closed auction notifications
	list ($lastvisit_day, $lastvisit_hour, $lastvisit_min) = $dbinfo->get_lastlogout ($username);
	if ($lastvisit_day != -1)
	{
		// Get closed auctions they have bid on, since last visit
		$result = $dbinfo->query ("select * from bids_on
where  username = '$username'
AND   (item_end_day < $curr_day
	OR (item_end_day = $curr_day
		AND item_end_hour < $curr_hour)
	OR (item_end_day = $curr_day
```

```
                AND item_end_hour = $curr_hour
                AND item_end_minute < $curr_minute))
AND   (item_end_day > $lastvisit_day
        OR (item_end_day = $lastvisit_day
                AND item_end_hour > $lastvisit_hour)
        OR (item_end_day = $lastvisit_day
                AND item_end_hour = $lastvisit_hour
                AND item_end_minute >= $lastvisit_min))
order by item_end_day desc, item_end_hour desc, item_end_minute desc");
                while (list ($user, $title, $seller, $category, $end_day, $end_hour, $end_min,
                        $bid_day, $bid_hour, $bid_min, $bid_amt)
                    = mysql_fetch_row ($result))
                {
                        $real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
                        $seller_realname = href ("profile.php?mode=view&username=$seller", $dbinfo-
>get_realname ($seller));
                        list ($winner, $winner_realname, $win_price) =
                                $dbinfo->get_winner ($title, $seller, $category, $end_day, $end_hour,
$end_min);
                        if ($winner != -1 && $winner != $username)
                        {
                                $winner_realname = href ("profile.php?
mode=view&username=$winner", $winner_realname);
                                $output[$x] = pad_time ($end_day, $end_hour, $end_min, 4).div (span
(format_time ($end_day, $end_hour, $end_min), "time").
                                        "The auction \"$real_title\" by $seller_realname that you have
partipated in has ended since your last visit. ".
                                        "$winner_realname won the auction with a bid of \$$win_price.",
"closed");
                                $x++;
                        }
                        else
                        {
                                $winner_realname = href ("profile.php?
mode=view&username=$winner", $winner_realname);
                                $output[$x] = pad_time ($end_day, $end_hour, $end_min, 4).div (span
(format_time ($end_day, $end_hour, $end_min), "time").
                                        "The auction \"$real_title\" by $seller_realname that you have
partipated in has ended since your last visit. ".
                                        "You won the auction with a bid of \$$win_price.", "closed");
                                $x++;
                        }
                }
                mysql_free_result ($result);

                // Get closed auctions they are the seller for
                $result = $dbinfo->query ("select * from item_listing
```

```
where  seller = '$username'
AND   (end_day < $curr_day
        OR (end_day = $curr_day
                AND end_hour < $curr_hour)
        OR (end_day = $curr_day
                AND end_hour = $curr_hour
                AND end_minute < $curr_minute))
AND   (end_day > $lastvisit_day
        OR (end_day = $lastvisit_day
                AND end_hour > $lastvisit_hour)
        OR (end_day = $lastvisit_day
                AND end_hour = $lastvisit_hour
                AND end_minute >= $lastvisit_min))
order by end_day desc, end_hour desc, end_minute desc");
                $t = new table_common_t ();
                $t->init ("feedback");
                while (list ($title, $seller, $category, $end_day, $end_hour, $end_min,
                        $description, $shipping_cost, $shipping_method, $starting_price,
                        $current_price, $picture, $buyer, $buyer_fdbk, $buyer_fdbk_rating,
                        $seller_fdbk)
                    = mysql_fetch_row ($result))
                {
                    $real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
                    if ($buyer == "None")
                    {
                        $output[$x] = pad_time ($end_day, $end_hour, $end_min, 4).div (span
(format_time ($end_day, $end_hour, $end_min), "time").
                            "Your auction \"$real_title\" has ended since your last visit with
no buyer.", "closed");
                        $x++;
                    }
                    else
                    {
                        $winner_realname = href ("profile.php?mode=view&username=$buyer",
$dbinfo->get_realname ($buyer));
                        $output[$x] = pad_time ($end_day, $end_hour, $end_min, 4).div (span
(format_time ($end_day, $end_hour, $end_min), "time").
                            "Your auction \"$real_title\" has ended since your last visit, with
$winner_realname winning with a high bid of \$$current_price.", "closed");
                        $x++;
                    }

                }
                mysql_free_result ($result);
        }

        // Get all "outbid" notifications
```

```
        $result = $dbinfo->query ("select * from bids_on
where  username = '$username'
AND   (item_end_day > $curr_day
        OR      (item_end_day = $curr_day AND item_end_hour > $curr_hour)
        OR      (item_end_day = $curr_day AND item_end_hour = $curr_hour AND item_end_minute
> $curr_minute)
        )
AND   display_notification = 'y'
order by bid_day desc, bid_hour desc, bid_minute desc");
        while (list ($user, $title, $seller, $category, $end_day, $end_hour, $end_min,
                    $bid_day, $bid_hour, $bid_min, $bid_amt)
            = mysql_fetch_row ($result))
        {
                $real_title = href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title);
                // Find the person that out bid "you"
                $result2 = $dbinfo->query ("select username, bid_amount, bid_day, bid_hour,
bid_minute from bids_on
where  item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_min
AND   bid_amount > $bid_amt
order by bid_amount");
                list ($outbid_user, $outbid_amt, $outbid_day, $outbid_hour, $outbid_min)
                        = mysql_fetch_row ($result2);
                mysql_free_result ($result2);
                // Find the current highest bidder for the item.
                $result2 = $dbinfo->query ("select username, bid_amount, bid_day, bid_hour,
bid_minute from bids_on
where  item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_min
order by bid_amount desc");
                list ($highest_user, $highest_amt, $highest_day, $highest_hour, $highest_min)
                        = mysql_fetch_row ($result2);
                mysql_free_result ($result2);
                $outbid_realname = href ("profile.php?mode=view&username=$outbid_user", $dbinfo-
>get_realname ($outbid_user));
                $highest_realname = href ("profile.php?mode=view&username=$highest_user",
$dbinfo->get_realname ($highest_user));
                $seller_realname = href ("profile.php?mode=view&username=$seller", $dbinfo-
>get_realname ($seller));
```

```
                $bid_diff = $outbid_amt - $bid_amt;
                $highest_bid_diff = $highest_amt - $bid_amt;
                $output[$x] = pad_time ($outbid_day, $outbid_hour, $outbid_min, 6).div (div (href
("$current_script?
mode=remove&t=$title&s=$seller&c=$category&d=$end_day&h=$end_hour&m=$end_min&bd=$bi
d_day&bh=$bid_hour&bm=$bid_min", "X"), "outbid_hide").
                        span (format_time ($outbid_day, $outbid_hour, $outbid_min), "time").
                        "$outbid_realname outbid you by \$$bid_diff on $seller_realname's auction
of \"$real_title\" with a bid of \$$outbid_amt.<br/>".
                        span (format_time ($highest_day, $highest_hour, $highest_min),
"time2")."$highest_realname became the highest bidder (\$$highest_bid_diff over your bid) with a bid
of \$$highest_amt.<br/>".
                        span (format_time ($end_day, $end_hour, $end_min), "time2")." The auction
will end.", "outbid");
                $x++;
        }
        mysql_free_result ($result);

        // Get user activity
        $result = $dbinfo->query ("select day, hour, minute, activity from user_activity
where username = '$username'
AND activity != 'Current Time'
order by day desc, hour desc, minute desc");
        while (list ($d, $h, $m, $a) = mysql_fetch_row ($result))
        {
                $output[$x] = pad_time ($d, $h, $m, 3).div (span (format_time ($d, $h, $m),
"time")."$a", "useractivity");
                $x++;
        }
        mysql_free_result ($result);


        // Actual output of all notifications, sorted by time
        echo h3 ("Welcome to your home page.");
        cout ("Activity and notifications:");
        if ($x > 1)
                rsort ($output);
        else
                cout ("None.");
        for ($i = 0; $i < count ($output); $i++)
        {
                echo substr ($output[$i], strpos ($output[$i], " "), strlen ($output[$i]));
        }
}
else // user is not logged in
{
        $loginlink = href ("login.php", "Login");
        $reglink= href ("registration.php", "Register");
        echo <<<HEREDOC
```

```
Welcome "iBay", presented by CMPS460 Group I!<br/>You will need to $loginlink or $reglink to
proceed further.
HEREDOC;
        echo_div("home-img");
        echo local_img("gavel.jpg");
        end_div();
}

echo_footer ($dbinfo);
?>
```

itemlisting.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Handles item / auction listing related things such as display, bidding,
// editing, deleting




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);

// Javascript functions for validation/etc..
echo <<<HEREDOC
<script type="text/javascript">
function reset_inputs (thisform)
{
        // reset inputs
        // http://webcheatsheet.com/javascript/form_validation.php
        for each (elem in thisform.elements)
        {
                if (elem.type == "text" || elem.type == "select-one")
                {
                        elem.style.background = '';
                }
        }
}

function validate_required (field)
```

```
{
	// http://www.w3schools.com/jS/js_form_validation.asp
	// http://webcheatsheet.com/javascript/form_validation.php
	with (field)
	{
		if (value == null || value == "" || value == "-")
		{
			field.style.background = 'Yellow';
			return false;
		}
		else
		{
			return true;
		}
	}
}

function field_set (fieldvar, field)
{
	if (fieldvar == null)
		return field;
	else
		return fieldvar;
}

function append_with_newline (orig, append_text)
{
	if (orig != null)
		return orig + '\\n' + append_text;
	return append_text;
}

function validate_form (thisform)
{
	var field_of_focus;
	var alert_message;

	reset_inputs (thisform);

	with (thisform)
	{
		if (validate_required (title) == false)
		{
			field_of_focus = field_set (field_of_focus, title);
			alert_message = append_with_newline (alert_message, "Auction title required.");
		}
		if (validate_required (description) == false)
		{
			field_of_focus = field_set (field_of_focus, description);
```

```
                        alert_message = append_with_newline (alert_message, "Auction description
required.");
                }

                if (alert_message != null)
                {
                        alert (alert_message);
                        field_of_focus.focus();
                        return false;
                }
        }
}

function local_img (file)
{
        return "images/" + file;
}

function image_swap ()
{
        var image = document.getElementById ("dynimg");
        var cat = document.getElementById ("dyncategory");
        var pic = document.getElementById ("dynpicture");
        image.src = local_img (cat.value.toLowerCase ().replace (" ", "_") + pic.value);
}

/* onload = function() */
/* { */
/*      image_swap (); */
/* } */

</script>
HEREDOC; // end javascript functions


$mode = get ("mode");
$current_script = current_script ();
$username = $dbinfo->username ();
$curr_day = $dbinfo->day ();
$curr_hour = $dbinfo->hour ();
$curr_minute = $dbinfo->minute ();
$title = get ("title");
$seller = get ("seller");
$category = get ("category");
$end_day = get ("end_day");
$end_hour = get ("end_hour");
$end_minute = get ("end_minute");

// Redirect if not logged in
```

```
if (!$dbinfo->logged_in ())
	redirect ("index.php");

// Kind of a catch-all form for editing & creating auctions
function item_listing_form ($mode = "new", $t = "", $s = "", $c = "", $d = -1, $h = -1, $m = -1)
{
	global $dbinfo;
	global $current_script;
	global $curr_day, $curr_hour, $curr_minute;

	// If editing auction, fetch the current information from the database
	if ($mode == "edit" && $m != -1 && $dbinfo->auction_exists ($t, $s, $c, $d, $h, $m))
	{
		$result = $dbinfo->query ("select * from item_listing
where title = '$t'
AND   seller = '$s'
AND   category = '$c'
AND   end_day = $d
AND   end_hour = $h
AND   end_minute = $m");
		if ($result)
		{
			list ($title, $seller, $category, $end_day, $end_hour, $end_min,
			    $description, $shipping_cost, $shipping_method, $starting_price,
			    $current_price, $picture, $buyer, $buyer_fdbk, $buyer_fdbk_rating,
			    $seller_fdbk)
				= mysql_fetch_row ($result);
		}
		else if ($dbinfo->debug ())
		{
			mysql_free_result ($result);
			die (mysql_error ());
		}
		else
		{
			mysql_free_result ($result);
			echo "Couldn't get auction information.";
			return;
		}
	}
	else if ($mode != "new")
	{
		echo "You've reached this page in error.";
		return;
	}

	// Start writing table to page
	echo h3 ("Auction Management");
	$table = new table_common_t ();
```

```
	$table->init ("itemlisting");

	echo $table->table_begin ();
	echo $table->table_head_begin ();
	if ($mode == "edit")
		echo $table->tr ($table->td_span ("Edit your auction", "", 2));
	else
		echo $table->tr ($table->td_span ("Create a new auction", "", 2));

	echo $table->tr_end ();
	echo $table->table_head_end ();
	echo $table->table_body_begin ();

	// Different modes depending on if editing or making new
	if ($mode == "edit")
	{
		echo form_begin_n ("$current_script?mode=save", "post", "item_listing_form", "return
validate_form (this)");
		echo $table->tr ($table->td ("Title<br/>".href ("$current_script?
mode=delete&title=$t&seller=$s&category=$c&end_day=$d&end_hour=$h&end_minute=$m",
"Delete Auction")).
				$table->td (text_input_sr ("title", $t, 20, 50).
					alert ("A required field, this is the title of the auction", "?")));
		if ($dbinfo->is_admin () && $username != $s)
			echo $table->tr ($table->td ("Seller").
				$table->td (text_input_sr ("seller", $s, 20, 50).
					alert ("The person who initiated the auction.", "?")));
		else
			echo hidden_input ("seller", $s);
		echo $table->tr ($table->td ("Category").
				$table->td (input ("category", $c, 20, "dyncategory", "text", 50,
"readonly").
					alert ("An auction can only belong to one category.", "?")));
		echo $table->tr ($table->td ("Description").
				$table->td (text_input_s ("description", $description, 50, 250).
					alert ("Under no circumstances leave this blank.
Be....descriptive", "?")));
		echo $table->tr ($table->td ("Closing Time").
				$table->td (text_input_sr ("closing_time", format_time ($d, $h, $m), 20,
50).
					alert ("Up to 10 days in advance. When do you want the fun to
end?", "?").
					hidden_input ("end_day", $d).
					hidden_input ("end_hour", $h).
					hidden_input ("end_minute", $m)));
		$options = "";
		$options = $options.option ("FedEx", "FedEx", $shipping_method);
		$options = $options.option ("UPS", "UPS", $shipping_method);
		$options = $options.option ("Air", "Air", $shipping_method);
```

```
                echo $table->tr ($table->td ("Shipping Method").
                                $table->td (select ("shipping_method", $options).
                                        alert ("Sorry, we no longer ship via horse and carriage.", "?")));
                $options = "";
                for ($i = 1; $i <= 100; $i++)
                        $options = $options.option ($i, "\$$i.00", $shipping_cost);
                echo $table->tr ($table->td ("Shipping Cost").
                                $table->td (select ("shipping_cost", $options).
                                        alert ("Your best estimate please.", "?")));
                echo $table->tr ($table->td ("Starting Price").
                                $table->td ("\$".text_input_sr ("starting_price", $starting_price, 10, 10).
                                        alert ("Be competitive.", "?")));
                $options = "";
                for ($i = 1; $i <= 3; $i++)
                {
                        $cat = str_replace (str_replace (" ", "_", strtolower ($category)), "", $picture);
                        $options = $options.option ("$i.jpg", "Picture $i", $cat);
                }
                echo $table->tr ($table->td ("Picture").
                                $table->td (select_dyn ("picture", $options, "image_swap()").
                                        href ("category_pictures.php", "?", "_blank")."<br/>".
                                        local_img ($picture)));
                echo $table->tr ($table->td_span (submit_input ("Save"), "", 2, "center"));
        }
        else // making new listing
        {
                echo form_begin_n ("$current_script?mode=savenew", "post", "item_listing_form",
"return validate_form (this)");
                echo $table->tr ($table->td ("Title").
                                $table->td (text_input_s ("title", $t, 20, 50).
                                        alert ("A required field, this is the title of the auction", "?")));
                $options = "";
                $options = $options.option ("Art", "Art");
                $options = $options.option ("Books", "Books");
                $options = $options.option ("Clothing", "Clothing");
                $options = $options.option ("Collectibles", "Collectibles");
                $options = $options.option ("Electronics", "Electronics");
                $options = $options.option ("Entertainment", "Entertainment");
                $options = $options.option ("Jewelry", "Jewelry");
                $options = $options.option ("Sporting Goods", "Sporting Goods");
                $options = $options.option ("Toys", "Toys");
                echo $table->tr ($table->td ("Category").
                                $table->td (select_dyn ("category", $options, "image_swap()").
                                        alert ("An auction can only belong to one category.", "?")));
                echo $table->tr ($table->td ("Description").
                                $table->td (text_input_s ("description", "", 50, 250).
                                        alert ("Under no circumstances leave this blank.
Be....descriptive", "?")));
                $options = "";
```

```
		for ($i = $curr_day; $i <= $curr_day + 7; $i++)
		{
			for ($j = ($i == $curr_day)? $curr_hour + 1 : 0; $j < 24; $j++)
			{
				$options = $options.option ("$i $j 0", format_time ($i, $j, 0));
			}
		}
		echo $table->tr ($table->td ("Closing Time").
				$table->td (select ("closing_time", $options).
					alert ("Up to 10 days in advance. When do you want the fun to
end?", "?")));
		$options = "";
		$options = $options.option ("FedEx", "FedEx");
		$options = $options.option ("UPS", "UPS");
		$options = $options.option ("Air", "Air");
		echo $table->tr ($table->td ("Shipping Method").
				$table->td (select ("shipping_method", $options).
					alert ("Sorry, we no longer ship via horse and carriage.", "?")));
		$options = "";
		for ($i = 1; $i <= 100; $i++)
			$options = $options.option ($i, "\$$i.00");
		echo $table->tr ($table->td ("Shipping Cost").
				$table->td (select ("shipping_cost", $options).
					alert ("Your best estimate please.", "?")));
		$options = "";
		for ($i = 1; $i <= 350; $i++)
		{
			$options = $options.option ("$i.00", "\$$i.00");
			$options = $options.option ("$i.50", "\$$i.50");
		}
		echo $table->tr ($table->td ("Starting Price").
				$table->td (select ("starting_price", $options).
					alert ("Be competitive.", "?")));

		$options = "";
		for ($i = 1; $i <= 3; $i++)
		{
			$options = $options.option ("$i.jpg", "Picture $i");
		}
		echo $table->tr ($table->td ("Picture").
				$table->td (select_dyn ("picture", $options, "image_swap()").
					href ("category_pictures.php", "?", "_blank")."<br/>".
					local_img ("art1.jpg")));
		echo $table->tr ($table->td_span (submit_input ("Create"), "", 2, "center"));
	}
	echo form_end ();
	echo $table->table_body_end ();
	echo $table->table_end ();
}
```

```
// Very simple checking of the form data to meet basic requirements
function verify_data ()
{
	global $dbinfo, $errors, $post_title, $post_seller,
		$post_category,
		$post_end_day, $post_end_hour, $post_end_minute,
		$post_closing_time, $post_description,
		$post_shipping_method,
		$post_shipping_cost, $post_starting_price,
		$post_picture;
	$errors = "";
	$post_title = fix_quotes (post ("title"));
	$post_seller = post ("seller");
	$post_category = post ("category");
	$post_closing_time = post ("closing_time");
	$post_end_day = strtok ($post_closing_time, " ");
	$post_end_hour = strtok (" ");
	$post_end_minute = strtok (" ");
	$post_description = fix_quotes (post ("description"));
	$post_shipping_method = post ("shipping_method");
	$post_shipping_cost = post ("shipping_cost");
	$post_starting_price = post ("starting_price");
	$post_picture = post ("picture");

	if (empty ($post_title))
		$errors = $errors.li ("Title can't be empty.");
	if (strlen ($post_title) > 50)
		$errors = $errors.li ("Title can't more than 50 characters.");
	if (empty ($post_description))
		$errors = $errors.li ("Description can't be empty.");
	if (strlen ($post_description) > 250)
		$errors = $errors.li ("Description can't be more than 250 characters.");
	return $errors;
}

// Handle bid
if ($mode == "bid")
{
	$bid_title = (post ("title"));
	$bid_seller = post ("seller");
	$bid_category = post ("category");
	$bid_end_day = post ("end_day");
	$bid_end_hour = post ("end_hour");
	$bid_end_minute = post ("end_minute");
	$bid_amount = post ("bid_amount");

	list ($highest_bidder, $amt) = $dbinfo->get_highest_bidder ($bid_title, $bid_seller,
$bid_category,
```

```
                                                        $bid_end_day, $bid_end_hour,
$bid_end_minute);
        if ($highest_bidder != $username)
        {
                $dbinfo->update_auction_before_bid (
                        $bid_title, $bid_seller, $bid_category,
                        $bid_end_day, $bid_end_hour, $bid_end_minute,
                        $bid_amount);
                $dbinfo->query ("insert into bids_on values ('$username', '$bid_title', '$bid_seller',
'$bid_category', $bid_end_day, $bid_end_hour, $bid_end_minute, $curr_day, $curr_hour,
$curr_minute, $bid_amount, 'n')");
                $dbinfo->save_activity ("Bid \$$bid_amount on the auction: \"".href ("itemlisting.php?
mode=view&title=$bid_title&seller=$bid_seller&category=$bid_category&end_day=$bid_end_day&e
nd_hour=$bid_end_hour&end_minute=$bid_end_minute", $bid_title)."\".");
        }
        else
        {
                echo "Errors were detected. Please correct before continuing:<br/>";
                echo ul (li ("Don't double bid!"));
                $dbinfo->save_activity ("You tried to game the system by double bidding \$
$bid_amount on the auction: \"".href ("itemlisting.php?
mode=view&title=$bid_title&seller=$bid_seller&category=$bid_category&end_day=$bid_end_day&e
nd_hour=$bid_end_hour&end_minute=$bid_end_minute", $bid_title)."\".");
        }

        $mode = "view";
}

// View an auction listing
if ($mode == "view")
{
        $sfdbk = get ("sfdbk");
        $bfdbk = get ("bfdbk");
        if ($dbinfo->is_admin () && !empty ($sfdbk))
        {
                $dbinfo->query ("update item_listing set
sellerfeedbackforbuyer_description = "
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");
        }
        if ($dbinfo->is_admin () && !empty ($bfdbk))
        {
                $dbinfo->query ("update item_listing set
buyerfeedbackforseller_description = ",
buyerfeedbackforseller_rating = -1
```

```
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");
        }

        $result = $dbinfo->query ("select * from item_listing
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");

        if (mysql_num_rows ($result) == 0)
        {
                cout ("Invalid item listing.");
        }
        else
        {
                list ($title, $seller, $category, $end_day, $end_hour, $end_min,
                    $description, $shipping_cost, $shipping_method, $starting_price,
                    $current_price, $picture, $buyer, $buyer_fdbk, $buyer_fdbk_rating,
                    $seller_fdbk)
                        = mysql_fetch_row ($result);
                mysql_free_result ($result);

                $t = new table_common_t ();
                $t->init ("itemlisting");
                $seller_realname = href ("profile.php?mode=view&username=$seller", $dbinfo-
>get_realname ($seller));
                list ($d, $h, $m) = $dbinfo->get_registration_date ($seller);
                $seller_registration = format_time ($d, $h, $m);
                $closed = is_older ($end_day, $end_hour, $end_minute, $curr_day, $curr_hour,
$curr_minute);
                if ($closed)
                {
                        $time_message = $t->tr ($t->td ("Ended on:").
                                                $t->td (format_time ($end_day, $end_hour, $end_min)));
                        $curr_price_message = $t->tr ($t->td ("End Price:").
                                                $t->td ("\$$current_price"));
                        // feedback
                        if ($buyer_fdbk_rating != -1)
                        {
                                if ($dbinfo->is_admin ())
                                {
                                        $url = "$current_script?";
```

```
                                        foreach ($_GET as $var => $val)
                                                $url = $url."$var=$val&";
                                        $url = $url."bfdbk=delete";
                                        $fdbk = $t->tr ($t->td ("Feedback from buyer:<br/>".
                                                                href ($url, "Delete")).
                                                        $t->td ($buyer_fdbk." Rating:
$buyer_fdbk_rating"));
                                }
                                else
                                {
                                        $fdbk = $t->tr ($t->td ("Feedback from buyer:").
                                                        $t->td ($buyer_fdbk." Rating:
$buyer_fdbk_rating"));
                                }
                        }
                        else
                        {
                                $fdbk = $t->tr ($t->td ("Feedback from buyer:").
                                                $t->td ("None."));
                        }
                        // feedback
                        if (!empty ($seller_fdbk))
                        {
                                if ($dbinfo->is_admin ())
                                {
                                        $url = "$current_script?";
                                        foreach ($_GET as $var => $val)
                                                $url = $url."$var=$val&";
                                        $url = $url."sfdbk=delete";
                                        $fdbk = $fdbk.$t->tr ($t->td ("Feedback from seller:<br/>".
                                                                href ($url, "Delete")).
                                                        $t->td ($seller_fdbk));
                                }
                                else
                                {
                                        $fdbk = $fdbk.$t->tr ($t->td ("Feedback from seller:").
                                                        $t->td ($seller_fdbk));
                                }
                        }
                        else
                        {
                                $fdbk = $fdbk.$t->tr ($t->td ("Feedback from seller:").
                                                $t->td ("None."));
                        }
                }
                else
                {
                        $time_message = $t->tr ($t->td ("Ends on:").
                                                $t->td (format_time ($end_day, $end_hour, $end_min)));
```

```
                        list ($highest_bidder, $highest_bid) =
                                $dbinfo->get_highest_bidder ($title, $seller, $category, $end_day,
$end_hour, $end_min);
                        if ($highest_bidder == "None")
                                $highest_bid = $current_price;

                        // bid form
                        if ($username != $seller &&
                           $username != $highest_bidder)
                        {
                                $options = "";
                                for ($i = $highest_bid + 0.50; $i <= $highest_bid + 50; $i += 0.50)
                                {
                                        $i = sprintf ("%.2f", $i);
                                        $options = $options.option ($i, "\$$i");
                                }
                                $url = "$current_script?";
                                foreach ($_GET as $var => $val)
                                        $url = $url."$var=$val&";
                                $url = $url."mode=bid";
                                $curr_price_message = $t->tr ($t->td ("Current Price:").
                                                        $t->td ("\$$current_price".
                                                                form_begin ("$url", "post").
                                                                select ("bid_amount", $options).
                                                                hidden_input ("title", $title).
                                                                hidden_input ("seller", $seller).
                                                                hidden_input ("category", $category).
                                                                hidden_input ("end_day", $end_day).
                                                                hidden_input ("end_hour", $end_hour).
                                                                hidden_input ("end_minute",
$end_min).
                                                                submit_input ("Place Bid").
                                                                form_end ())));
                        }
                        else
                        {
                                $curr_price_message = $t->tr ($t->td ("Current Price:").
                                                        $t->td ("\$$current_price"));
                        }
                }
                $num_bids = $dbinfo->get_num_bids ($title, $seller, $category, $end_day, $end_hour,
$end_min);

                if ($username == $seller || $dbinfo->is_admin ())
                {
                        echo_div ("scriptstatus");
                        echo href ("$current_script?
mode=edit&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_ho
ur&end_minute=$end_minute", "Edit Listing");
```

```
                        echo " | ";
                        echo href ("$current_script?
mode=delete&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_
hour&end_minute=$end_minute", "Delete Listing");
                        end_div ();
                }

                // output auction info
                echo $t->table_begin ().$t->table_head_begin ().
                        $t->tr ($t->td_span ("Auction Listing: \"".$title."\"", "", 2)).
                        $t->table_head_end ().$t->table_body_begin ().
                        $t->tr ($t->td_span (local_img ($picture), "", 2, "center")).
                        $t->tr ($t->td ("Start price:").
                                $t->td ("\$$starting_price")).
                        $t->tr ($t->td ("Shipping:").
                                $t->td ("\$$shipping_cost ($shipping_method)")).
                        $curr_price_message.
                        $t->tr ($t->td ("Number of bids:").
                                $t->td ("$num_bids")).
                        $t->tr ($t->td ("Description:").
                                $t->td ($description)).
                        $t->tr ($t->td ("Sold by:").
                                $t->td ("$seller_realname<br/>(Registered since $seller_registration)")).
                        $t->tr ($t->td ("Sold in:").
                                $t->td ($category)).
                        $time_message.
                        $fdbk.
                        $t->table_body_end ().$t->table_end ();

                // Get bid history for item.
                if (false && $dbinfo->debug ())
                {
                        cout ("Debug enabled.");
                        cout ("select username, bid_day, bid_hour, bid_minute, bid_amount from
bids_on
where  item_title = '$title'
AND    item_seller = '$seller'
AND    item_category = '$category'
AND    item_end_day = $end_day
AND    item_end_hour = $end_hour
AND    item_end_minute = $end_minute
order by bid_amount desc");
                }
                $result = $dbinfo->query ("select username, bid_day, bid_hour, bid_minute, bid_amount
from bids_on
where  item_title = '$title'
AND    item_seller = '$seller'
AND    item_category = '$category'
AND    item_end_day = $end_day
```

```
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute
order by bid_amount desc");
                cout ("");
                cout ("Bid history:");
                if (mysql_num_rows ($result) == 0)
                {
                        echo div ("None.", "bidhistory");
                }
                else
                {
                        while (list ($bidder, $bid_day, $bid_hour, $bid_minute, $bid_amount)
                             = mysql_fetch_row ($result))
                        {
                                $bidder_realname = href ("profile.php?mode=view&username=$bidder",
$dbinfo->get_realname ($bidder));
                                echo div (span (format_time ($bid_day, $bid_hour, $bid_minute),
"time").
                                        "$bidder_realname bid \$$bid_amount.", "bidhistory");
                        }
                }
        }
}
else if ($mode == "new") // make a new auction listing
{
        item_listing_form ($mode);
}
else if ($mode == "edit") // edit an auction listing
{
        item_listing_form ($mode, $title, $seller, $category, $end_day, $end_hour, $end_minute);
}
else if ($mode == "savenew") // save a new auction listing
{
                verify_data ();
                if ($dbinfo->auction_exists ($post_title, $username, $post_category,
                                        $post_end_day, $post_end_hour, $post_end_minute))
                {
                        $errors = $errors.li ("Auction already exists.");
                }

                if (!empty ($errors))
                {
                        echo "Errors were detected. Please correct before continuing:<br/>";
                        echo ul ($errors);
                        echo href ("$current_script?mode=new", "Go to create a new listing.");
                }
                else
                {
                        $post_picture = str_replace (" ", "_", strtolower ($post_category.$post_picture));
```

```
                        // insert item into the database
                        $dbinfo->query ("insert into item_listing values (
'$post_title',
'$username',
'$post_category',
$post_end_day,
$post_end_hour,
$post_end_minute,
'$post_description',
$post_shipping_cost,
'$post_shipping_method',
$post_starting_price,
$post_starting_price,
'$post_picture',
'',
'',
-1,
')");

                        if (!$dbinfo->auction_exists ($post_title, $username, $post_category,
                                                $post_end_day, $post_end_hour, $post_end_minute))
                        {
                                cout ("Auction listing failed. ");
                                cout (href (current_script (), "Try again."));
                        }
                        else
                        {
                                $dbinfo->save_activity ("Listed a new auction: \"".href ("itemlisting.php?
mode=view&title=$post_title&seller=$username&category=$post_category&end_day=$post_end_day
&end_hour=$post_end_hour&end_minute=$post_end_minute", $post_title)."\".");
                                cout ("Auction listing successful.");
                                cout ("Would you like to ".href ("itemlisting.php?
mode=view&title=$post_title&seller=$username&category=$post_category&end_day=$post_end_day
&end_hour=$post_end_hour&end_minute=$post_end_minute", "view")." it?");
                        }
                }
}
else if ($mode == "save") // save an editing (current) auction listing
{
                verify_data ();
                if (!empty ($errors))
                {
                        echo "Errors were detected. Please correct before continuing:<br/>";
                        echo ul ($errors);
                }
                else
                {
                        $post_end_day = post ("end_day");
```

```
                        $post_end_hour = post ("end_hour");
                        $post_end_minute = post ("end_minute");
                        $post_picture = str_replace (" ", "_", strtolower ($post_category.$post_picture));
                        // update item in the database

                        $dbinfo->query ("update item_listing set
description = '$post_description',
shipping_cost = $post_shipping_cost,
shipping_method = '$post_shipping_method',
picture = '$post_picture'
where  title = '$post_title'
AND   seller = '$post_seller'
AND   category = '$post_category'
AND   end_day = $post_end_day
AND   end_hour = $post_end_hour
AND   end_minute = $post_end_minute");
                        $dbinfo->save_activity ("Edited auction: \"".href ("itemlisting.php?
mode=view&title=$post_title&seller=$post_seller&category=$post_category&end_day=$post_end_da
y&end_hour=$post_end_hour&end_minute=$post_end_minute", $post_title)."\".");
                        if ($dbinfo->is_admin () && $post_seller != $username)
                        {
                                $realname = href ("profile.php?mode=view&username=$username",
$dbinfo->realname ());
                                $dbinfo->save_activity_for ($post_seller, "$realname edited your auction:
\"".href ("itemlisting.php?
mode=view&title=$post_title&seller=$post_seller&category=$post_category&end_day=$post_end_da
y&end_hour=$post_end_hour&end_minute=$post_end_minute", $post_title)."\".");
                        }
                        cout ("Auction update successful.");
                        cout ("Would you like to ".href ("itemlisting.php?
mode=view&title=$post_title&seller=$post_seller&category=$post_category&end_day=$post_end_da
y&end_hour=$post_end_hour&end_minute=$post_end_minute", "view")." it?");
                }
}
else if ($mode == "delete") // delete an exiting auction listing
{
        $title = get ("title");
        $seller = get ("seller");
        $category = get ("category");
        $end_day = get ("end_day");
        $end_hour = get ("end_hour");
        $end_minute = get ("end_minute");

        if ($dbinfo->is_admin () || ($seller == $username))
        {
                if ($dbinfo->get_num_bids ($title, $seller, $category,
                                        $end_day, $end_hour, $end_minute) == 0)
                {
                        $dbinfo->query ("delete from item_listing
```

```
where  title = '$title'
AND   seller = '$seller'
AND   category = '$category'
AND   end_day = $end_day
AND   end_hour = $end_hour
AND   end_minute = $end_minute");
                        $dbinfo->save_activity ("Deleted the auction \"$title\".");
                        if ($dbinfo->is_admin () && ($seller != $username))
                        {
                                $realname = href ("profile.php?mode=view&username=$username",
$dbinfo->realname ());
                                $dbinfo->save_activity_for ($seller, "$realname deleted the
auction \"$title\".");
                        }
                        cout ("Back to your ".href ("index.php", "home").".");
                }
                else
                {
                        echo "Errors were detected:<br/>";
                        echo ul (li ("There must be no bids for the item listing."));
                        cout ("Back to the ".href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_minute", "item").".");
                }
        }
        else
        {
                echo "Errors were detected:<br/>";
                echo ul (li ("Must be an administrator or be the seller for this auction."));
                cout ("Back to the ".href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_minute", "item").".");
        }
}

echo_footer ($dbinfo);
?>
```

list_common.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// HTML list related common functions




// Unordered Lists
function ul_begin ()
{
	return "<ul>\n";
}

function ul_end ()
{
	return "</ul>\n";
}

function ul_wrap ($list_elems)
{
	return ul_begin ().$list_elems.ul_end ();
}

function ul ($list_elems)
{
	return ul_wrap ($list_elems);
}

// Ordered Lists
function ol_begin ()
```

```
{
	return "<ol>\n";
}

function ol_end ()
{
	return "</ol>\n";
}

function ol_wrap ($list_elems)
{
	return ol_begin ().$list_elems.ol_end ();
}

function ol ($list_elems)
{
	return ol_wrap ($list_elems);
}

// List elements
function li_begin ()
{
	return "<li>\n";
}

function li_end ()
{
	return "</li>\n";
}

function li_wrap ($list_elem)
{
	return li_begin ().$list_elem.li_end ();
}

function li ($list_elem)
{
	return li_wrap ($list_elem);
}
?>
```

login.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// User login form page




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);

if ($dbinfo->logged_in ())
        redirect ("index.php");

// display login form
$info = post ("username");
$mode = get ("login");
if (empty ($mode))
{
        $msg = get ("message");
        if (!empty ($msg))
                cout ($msg);
        echo h3 ("Login");
        $table = new table_common_t ();
        $table->init ("tbl_std");
        echo $table->table_begin ();
        echo $table->table_head_begin ();
        echo $table->tr ($table->td_span ("Login with your iBay ID", "", 2));
        echo $table->table_head_end ();
```

```
	echo $table->table_body_begin ();
	echo form_begin ("login.php?login=y", "post");
	echo $table->tr ($table->td ("Username").
			$table->td (text_input_s ("username", "", 20, 50)));
	echo $table->tr ($table->td ("Password").
			$table->td (password_input_s ("password", "", 20, 50)));
	echo $table->tr ($table->td_span (submit_input ("Login"), "", 2, "center"));
	echo form_end ();
	echo $table->table_body_end ();
	echo $table->table_end ();
}
// process login
else
{
	$user = fix_quotes (post ("username"));
	$pass = fix_quotes (post ("password"));
	if ($dbinfo->login ($user, $pass))
	{
		$dbinfo->close ();
		redirect ("index.php");
	}
	else
	{
		$dbinfo->close ();
		redirect ("login.php?message=Login%20Failed!");
	}
}

echo_footer ($dbinfo);
?>
```

logout.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Logout processing page




include_once ("common.php");

$dbinfo = new dbinfo_t ();
$dbinfo->init ();
if (!$dbinfo->connect ())
        die ("Couldn't connect to database.");
$dbinfo->logout (); // saves activity and destroys session
?>
```

profile.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Shows the user the available pictures for a profile




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);

cout ("The following are the available profile pictures:");

cout (local_img ("default_profile.jpg"));
cout ("default_profile.jpg");
cout ("");
for ($i = 1; $i <= 7; $i++) // assume 7 pictures are available
{
        cout (local_img ("profile$i.jpg"));
        cout ("profile$i.jpg");
        cout ("");
}

echo_footer ($dbinfo);
?>
```

registration.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Handles registration related things such as new users and editing your
// current registration, as well view registered users (admin).




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);

// Javascript functions for validation/etc..
echo <<<HEREDOC
<script type="text/javascript">
function reset_inputs (thisform)
{
	// reset inputs
	// http://webcheatsheet.com/javascript/form_validation.php
	for each (elem in thisform.elements)
	{
		if (elem.type == "text" || elem.type == "select-one")
		{
			elem.style.background = '';
		}
	}
}

function validate_required (field)
{
```

```
	// http://www.w3schools.com/jS/js_form_validation.asp
	// http://webcheatsheet.com/javascript/form_validation.php
	with (field)
	{
		if (value == null || value == "" || value == "-")
		{
			field.style.background = 'Yellow';
			return false;
		}
		else
		{
			return true;
		}
	}
}

function field_set (fieldvar, field)
{
	if (fieldvar == null)
		return field;
	else
		return fieldvar;
}

function append_with_newline (orig, append_text)
{
	if (orig != null)
		return orig + '\\n' + append_text;
	return append_text;
}

function validate_email (field)
{
	// http://www.w3schools.com/jS/js_form_validation.asp
	// http://webcheatsheet.com/javascript/form_validation.php

	var email_filter = /^[^@]+@[^@.]+\.[^@]*\w\w$/;
	var illegal_chars= /[\(\)\<\>\,\;\:\\\"\[\]]/;
	var value = trim (field.value);

	if (value == null
		|| value == ""
		|| !email_filter.test (value)
		|| value.match (illegal_chars))
	{
		field.style.background = 'Yellow';
		return false;
	}
	return true;
```

```
}

function trim (string)
{
	// regex to replace one or more spaces at beginning of string
	// or one or more spaces at end of string, with empty string
	return string.replace (/^\s+|\s+$/, '');
}

function validate_int (field, length)
{
	with (field)
	{
		if (value.length < length || value * 1 == 0)
		{
			field.style.background = 'Yellow';
			return false;
		}
		else
		{
			return true;
		}
	}
}

function validate_form (thisform)
{
	var field_of_focus;
	var alert_message;

	reset_inputs (thisform);

	with (thisform)
	{
		if (validate_required (username) == false)
		{
			field_of_focus = field_set (field_of_focus, username);
			alert_message = append_with_newline (alert_message, "Username required.");
		}
		if (validate_required (password) == false)
		{
			field_of_focus = field_set (field_of_focus, password);
			alert_message = append_with_newline (alert_message, "Password required.");
		}
		if (validate_required (password2) == false)
		{
			field_of_focus = field_set (field_of_focus, password2);
			alert_message = append_with_newline (alert_message, "Password confirmation
required.");
```

```
		}
		if (password.value != password2.value)
		{
			field_of_focus = field_set (field_of_focus, password);
			alert_message = append_with_newline (alert_message, "Passwords must
match.");
		}
		if (validate_required (realname) == false)
		{
			field_of_focus = field_set (field_of_focus, realname);
			alert_message = append_with_newline (alert_message, "Real name required.");
		}
		if (validate_required (birth_date) == false)
		{
			field_of_focus = field_set (field_of_focus, birth_date);
			alert_message = append_with_newline (alert_message, "Birth date required.");
		}
		else if ((birth_date.value.length >= 6) == false)
		{
			field_of_focus = field_set (field_of_focus, birth_date);
			alert_message = append_with_newline (alert_message, "Birth date too short.");
		}
		if (validate_required (shipping_street) == false)
		{
			field_of_focus = field_set (field_of_focus, shipping_street);
			alert_message = append_with_newline (alert_message, "Street required.");
		}
		if (validate_required (shipping_city) == false)
		{
			field_of_focus = field_set (field_of_focus, shipping_city);
			alert_message = append_with_newline (alert_message, "City required.");
		}
		if (validate_required (shipping_zip) == false)
		{
			field_of_focus = field_set (field_of_focus, shipping_zip);
			alert_message = append_with_newline (alert_message, "Postal/zip code
required.");
		}
		else if (validate_int (shipping_zip, 5) == false) // this test works only for US codes
		{
			field_of_focus = field_set (field_of_focus, shipping_zip);
			alert_message = append_with_newline (alert_message, "Postal/zip code is not
valid.");
		}
		if (validate_required (phone) == false)
		{
			field_of_focus = field_set (field_of_focus, phone);
			alert_message = append_with_newline (alert_message, "Phone number
required.");
```

```
                }
                else if ((phone.value.length >= 10) == false)
                {
                        field_of_focus = field_set (field_of_focus, phone);
                        alert_message = append_with_newline (alert_message, "Phone number
required.");
                }
                if (validate_required (email) == false)
                {
                        field_of_focus = field_set (field_of_focus, email);
                        alert_message = append_with_newline (alert_message, "E-mail required.");
                }
                else if (validate_email (email) == false)
                {
                        field_of_focus = field_set (field_of_focus, email);
                        alert_message = append_with_newline (alert_message, "E-mail address not
valid.");
                }
                if (validate_required (card_number) == false)
                {
                        field_of_focus = field_set (field_of_focus, card_number);
                        alert_message = append_with_newline (alert_message, "Credit card number
required.");
                }
                else if (validate_int (card_number, 12) == false)
                {
                        field_of_focus = field_set (field_of_focus, card_number);
                        alert_message = append_with_newline (alert_message, "Credit card number is
too short.");
                }

                if (alert_message != null)
                {
                        alert (alert_message);
                        field_of_focus.focus();
                        return false;
                }
        }
}

/* onload = function() */
/* { */
/* } */

</script>
HEREDOC; // end javascript functions


$mode = get ("mode");
```

```
$current_script = current_script ();

$get_username = get ("username");
$get_msg = get ("msg");

if (!empty ($get_msg))
	echo "$msg";

// Main registration form - kind of catch-all for editing and creating registrations
function registration_form ($user = "")
{
	global $dbinfo;
	global $current_script;

	// If editing registration, fetch the current information from the database
	if (!empty ($user) && $dbinfo->user_exists ($user))
	{
		$result = $dbinfo->query ("select * from user where username = '$user'");
		if ($result)
		{
			list($username, $passwd, $is_admin, $name, $dob, $street,
			    $city, $state, $zip, $phone, $email, $cc, $ccn, $ccx,
			    $picture, $desc) = mysql_fetch_row ($result);
			mysql_free_result ($result);
		}
		else if ($dbinfo->debug ())
		{
			mysql_free_result ($result);
			die (mysql_error ());
		}
		else
		{
			mysql_free_result ($result);
			echo "Couldn't get user information.";
			return;
		}
	}
	else if (!empty ($username))
	{
		echo "You've reached this page in error.";
		return;
	}

	// Start writing table to page
	echo h3 ("Accounts and Registration");
	$table = new table_common_t ();
	$table->init ("tbl_std");

	echo $table->table_begin ();
```

```
	echo $table->table_head_begin ();
	if (!$dbinfo->logged_in ())
		echo $table->tr ($table->td_span ("Register for an iBay account", "", 2));
	else if ($dbinfo->logged_in () && $user != $dbinfo->username ())
		echo $table->tr ($table->td_span ("Update (not) your iBay account", "", 2));
	else
		echo $table->tr ($table->td_span ("Update your iBay account", "", 2));
	echo $table->table_head_end ();
	echo $table->table_body_begin ();

	// Different modes depending on if user is logged in or not
	if ($dbinfo->logged_in ())
		echo form_begin_n ("$current_script?mode=save", "post", "registration_form", "return
validate_form (this)");
	else
		echo form_begin_n ("$current_script?mode=savenew", "post", "registration_form",
"return validate_form (this)");

	// If editing, then make username field read only
	if (empty ($user))
		echo $table->tr ($table->td ("Desired Username").
				$table->td (text_input_s ("username", $username, 30, 50).
					alert ("50 characters or less. Have fun.", "?")));
	else
		echo $table->tr ($table->td ("Username<br/>".href ("$current_script?
mode=delete&username=$username", "Delete Account")).
				$table->td (text_input_sr ("username", $username, 30, 50)));

	// Passwords must match
	echo $table->tr ($table->td ("Desired Password").
			$table->td (password_input_s ("password", $passwd, 30, 50).
				alert ("50 characters or less, must match. Make it super secret.", "?")));
	echo $table->tr ($table->td ("Repeat Password").
			$table->td (password_input_s ("password2", $passwd, 30, 50).
				alert ("You thought there would be something helpful here. Just type
the same thing as in the above box.", "?")));

	// Only allow editing of this if the current user is already an admin
	if ($dbinfo->is_admin ())
	{
		$options = "";
		$options = $options.option ("0", "No", $is_admin);
		$options = $options.option ("1", "Yes", $is_admin);
		echo $table->tr ($table->td ("Admin").
				$table->td (select ("is_admin", $options).
					alert ("Give administrator priv.", "?")));
	}
	echo $table->tr ($table->td ("Real Name").
			$table->td (text_input_s ("realname", $name, 30, 100).
```

```
                        alert ("100 characters or less. e.g. John Smith", "?")));
echo $table->tr ($table->td ("Birth Date").
                $table->td (text_input_s ("birth_date", $dob, 10, 10).
                        alert ("Up to 10 characters, at least 6. e.g. 1980-10-17", "?")));
echo $table->tr ($table->td ("Shipping Street").
                $table->td (text_input_s ("shipping_street", $street, 30, 100).
                        alert ("100 characters or less. e.g. 56 Dodo Ln", "?")));
echo $table->tr ($table->td ("Shipping City").
                $table->td (text_input_s ("shipping_city", $city, 30, 50).
                        alert ("50 characters or less. e.g. Eunice", "?")));
// Gosh this is a lot - some aren't even states! Ha!
$options = "";
$options = $options.option ("Alabama", "Alabama", $state);
$options = $options.option ("Alaska", "Alaska", $state);
$options = $options.option ("American Samoa", "American Samoa", $state);
$options = $options.option ("Arizona", "Arizona", $state);
$options = $options.option ("Arkansas", "Arkansas", $state);
$options = $options.option ("California", "California", $state);
$options = $options.option ("Colorado", "Colorado", $state);
$options = $options.option ("Connecticut", "Connecticut", $state);
$options = $options.option ("Delaware", "Delaware", $state);
$options = $options.option ("District of Columbia", "District of Columbia", $state);
$options = $options.option ("Florida", "Florida", $state);
$options = $options.option ("Georgia", "Georgia", $state);
$options = $options.option ("Guam", "Guam", $state);
$options = $options.option ("Hawaii", "Hawaii", $state);
$options = $options.option ("Idaho", "Idaho", $state);
$options = $options.option ("Illinois", "Illinois", $state);
$options = $options.option ("Indiana", "Indiana", $state);
$options = $options.option ("Iowa", "Iowa", $state);
$options = $options.option ("Kansas", "Kansas", $state);
$options = $options.option ("Kentucky", "Kentucky", $state);
$options = $options.option ("Louisiana", "Louisiana", $state);
$options = $options.option ("Maine", "Maine", $state);
$options = $options.option ("Maryland", "Maryland", $state);
$options = $options.option ("Massachusetts", "Massachusetts", $state);
$options = $options.option ("Michigan", "Michigan", $state);
$options = $options.option ("Minnesota", "Minnesota", $state);
$options = $options.option ("Mississippi", "Mississippi", $state);
$options = $options.option ("Missouri", "Missouri", $state);
$options = $options.option ("Montana", "Montana", $state);
$options = $options.option ("Nebraska", "Nebraska", $state);
$options = $options.option ("Nevada", "Nevada", $state);
$options = $options.option ("New Hampshire", "New Hampshire", $state);
$options = $options.option ("New Jersey", "New Jersey", $state);
$options = $options.option ("New Mexico", "New Mexico", $state);
$options = $options.option ("New York", "New York", $state);
$options = $options.option ("North Carolina", "North Carolina", $state);
$options = $options.option ("North Dakota", "North Dakota", $state);
```

```
$options = $options.option ("Northern Marianas Islands", "Northern Marianas Islands", $state);
$options = $options.option ("Ohio", "Ohio", $state);
$options = $options.option ("Oklahoma", "Oklahoma", $state);
$options = $options.option ("Oregon", "Oregon", $state);
$options = $options.option ("Pennsylvania", "Pennsylvania", $state);
$options = $options.option ("Puerto Rico", "Puerto Rico", $state);
$options = $options.option ("Rhode Island", "Rhode Island", $state);
$options = $options.option ("South Carolina", "South Carolina", $state);
$options = $options.option ("South Dakota", "South Dakota", $state);
$options = $options.option ("Tennessee", "Tennessee", $state);
$options = $options.option ("Texas", "Texas", $state);
$options = $options.option ("Utah", "Utah", $state);
$options = $options.option ("Vermont", "Vermont", $state);
$options = $options.option ("Virginia", "Virginia", $state);
$options = $options.option ("Virgin Islands", "Virgin Islands", $state);
$options = $options.option ("Washington", "Washington", $state);
$options = $options.option ("West Virginia", "West Virginia", $state);
$options = $options.option ("Wisconsin", "Wisconsin", $state);
$options = $options.option ("Wyoming", "Wyoming", $state);
echo $table->tr ($table->td ("Shipping State").
                $table->td (select ("shipping_state", $options).
                        alert ("Don't lie on this choice.", "?")));
echo $table->tr ($table->td ("Shipping Zip Code").
                $table->td (text_input_s ("shipping_zip", $zip, 5, 10).
                        alert ("5 to 10 digits only. e.g. 90210", "?")));
echo $table->tr ($table->td ("Phone Number").
                $table->td (text_input_s ("phone", $phone, 12, 12).
                        alert ("10 to 12 characters. e.g. 123-456-7890", "?")));
echo $table->tr ($table->td ("Email Address").
                $table->td (text_input_s ("email", $email, 30, 50).
                        alert ("50 characters or less. e.g. bill@hat.com.", "?")));
$options = "";
$options = $options.option ("American Express", "American Express", $cc);
$options = $options.option ("Discover", "Discover", $cc);
$options = $options.option ("Mastercard", "Mastercard", $cc);
$options = $options.option ("Visa", "Visa", $cc);
echo $table->tr ($table->td ("Credit Card Type").
                $table->td (select ("card_type", $options).
                        alert ("You better have one.", "?")));
echo $table->tr ($table->td ("Credit Card Number").
                $table->td (text_input_s ("card_number", $ccn, 16, 16).
                        alert ("12 to 16 digits. e.g. 1111222233334444", "?")));

// For credit card expiration dates
$options = "";
for ($j = 9; $j <= 14; $j++)
{
        if ($j < 10)
                $j = "0".$j;
```

```
		for ($i = 1; $i <= 12; $i++)
		{
			if ($i < 10)
				$i = "0".$i;
			$options = $options.option ("$i/$j", "$i/$j", $ccx);
		}
	}
	echo $table->tr ($table->td ("Credit Card Expiration").
			 $table->td (select ("card_expire", $options).
				alert ("Expiration date of your credit card.", "?")));

	// Save or register depending on if logged in
	if ($dbinfo->logged_in ())
	{
		echo $table->tr ($table->td_span (submit_input ("Save"), "", 2, "center"));
	}
	else
	{
		echo $table->tr ($table->td_span (submit_input ("Register"), "", 2, "center"));
	}
	echo form_end ();
	echo $table->table_body_end ();
	echo $table->table_end ();
}

// Very simple checking of the form data to meet basic requirements
function verify_data ()
{
	global $dbinfo, $errors, $post_username, $passwd1, $passwd2,
		$is_admin, $post_realname, $post_birth_date, $post_street,
		$post_city, $post_state, $post_zip, $post_phone, $post_email,
		$post_card_type, $post_card_number, $post_card_expire,
		$post_picture, $post_description;
	$errors = "";
	$post_username = fix_quotes (post ("username"));
	$passwd1 = fix_quotes (post ("password"));
	$passwd2 = fix_quotes (post ("password2"));
	$is_admin = post ("is_admin");
	$post_realname = fix_quotes (post ("realname"));
	$post_birth_date = fix_quotes (post ("birth_date"));
	$post_street = fix_quotes (post ("shipping_street"));
	$post_city = fix_quotes (post ("shipping_city"));
	$post_state = post ("shipping_state");
	$post_zip = post ("shipping_zip");
	$post_phone = fix_quotes (post ("phone"));
	$post_email = fix_quotes (post ("email"));
	$post_card_type = post ("card_type");
	$post_card_number = post ("card_number");
```

```
	$post_card_expire = post ("card_expire");
	$post_picture = "default_profile.jpg";
	$post_description = "N/A";

	if (empty ($post_username))
		$errors = $errors.li ("Username can't be empty.");
	if (!$dbinfo->logged_in () && $dbinfo->user_exists ($post_username))
		$errors = $errors.li ("Username already exists.");
	if (strlen ($post_username) > 50)
		$errors = $errors.li ("Username must be less than 50 characters.");
	if (empty ($passwd1))
		$errors = $errors.li ("Password can't be empty");
	if ($passwd1 != $passwd2)
		$errors = $errors.li ("Passwords don't match");
	if ($dbinfo->is_admin () && $is_admin == 0 && $dbinfo->username () == $post_username)
	{
		$result = $dbinfo->query ("select username from user where username !=
'$post_username' AND is_admin = 1");
		if (mysql_num_rows ($result) == 0)
			$errors = $errors.li ("The number of users left that are admins can't be zero.");
	}
	if (empty ($post_realname))
		$errors = $errors.li ("Name can't be empty");
	if (empty ($post_birth_date))
		$errors = $errors.li ("Birth date can't be empty");
	else if (strlen ($post_birth_date) < 6)
		$errors = $errors.li ("Birth date too short");
	if (empty ($post_street))
		$errors = $errors.li ("Street can't be empty");
	if (empty ($post_city))
		$errors = $errors.li ("City can't be empty");
	if ($post_zip + 0 <= 0)
		$errors = $errors.li ("Invalid zip code");
	else if (strlen ($post_zip) < 5)
		$errors = $errors.li ("Zip code too short");
	if (empty ($post_phone))
		$errors = $errors.li ("Phone number can't be empty");
	else if (strlen ($post_phone) < 10)
		$errors = $errors.li ("Phone number too short");
	if (strlen ($post_email) < 5)
		$errors = $errors.li ("Invalid e-mail");
	if ($post_card_number + 0 <= 0)
		$errors = $errors.li ("Invalid credit card number");
	else if (strlen ($post_card_number) < 12)
		$errors = $errors.li ("Credit card number too short");
	return $errors;
}
```

```
if ($dbinfo->logged_in ())
{
	$user = $dbinfo->username ();

	// Admin user may be editing/viewing another user's page
	if ($dbinfo->is_admin () && !empty ($get_username))
	{
		$user = $get_username;
	}

	if ($mode == "login_as" && !empty ($get_username))
	{
		$dbinfo->save_activity ("Logged in as '$user'.");
		$dbinfo->login_as ($user);
		echo href ("$current_script?", "Click to refresh.");
	}
	// Admin wants to delete a user
	else if ($mode == "delete" && $dbinfo->is_admin () && !empty ($get_username) &&
$get_username != $dbinfo->username ())
	{
		$errors = "";
		// check to make sure not last user and that there is at least one admin
		$result = $dbinfo->query ("select username from user where username != '$user' AND
is_admin = 1");
		if (mysql_num_rows ($result) == 0)
			$errors = $errors.li ("The number of users left that are admins can't be zero.");
		mysql_free_result ($result);
		// check to make sure not participating in as buyer or seller in any auctions
		$result = $dbinfo->query ("select count(*) from item_listing where seller = '$user' OR
buyer = '$user'");
		list ($count) = mysql_fetch_row ($result);
		if ($count != 0)
			$errors = $errors.li ("The user must have not partipated in any auctions as seller
or buyer.");
		mysql_free_result ($result);
		if (empty ($errors))
		{
			$dbinfo->save_activity ("Deleted user '$user'.");
			$dbinfo->query ("delete from user_activity where username = '$user'");
			$dbinfo->query ("delete from bids_on where username = '$user'");
			$dbinfo->query ("delete from user where username = '$user'");
			echo "Deleted user. ".href ("$current_script?mode=browse", "Click to browse all
users.");
		}
		else
		{
			echo "Errors were detected:<br/>";
			echo ul ($errors);
			echo href ("$current_script?mode=browse", "Click to browse all users.");
```

```
                }
        }
        // User wants to delete their account
        else if ($mode == "delete")
        {
                $errors = "";
                // check to make sure not last user and that there is at least one admin
                $result = $dbinfo->query ("select username from user where username != '$user' AND
is_admin = 1");
                if (mysql_num_rows ($result) == 0)
                        $errors = $errors.li ("The number of users left that are admins can't be zero.");
                mysql_free_result ($result);
                // check to make sure not participating in as buyer or seller in any auctions
                $result = $dbinfo->query ("select count(*) from item_listing where seller = '$user' OR
buyer = '$user'");
                list ($count) = mysql_fetch_row ($result);
                if ($count != 0)
                        $errors = $errors.li ("You should not have partipated in any auctions as seller or
buyer to be deleted.");
                mysql_free_result ($result);
                if (empty ($errors))
                {
                        $dbinfo->query ("delete from user_activity where username = '$user'");
                        $dbinfo->query ("delete from bids_on where username = '$user'");
                        $dbinfo->query ("delete from user where username = '$user'");
                        $dbinfo->logout (true);
                        echo "Deleted account. ".href ("index.php", "Click to go to the home page.");
                }
                else
                {
                        echo "Errors were detected:<br/>";
                        echo ul ($errors);
                        echo href ("$current_script?mode=view", "Go to your account.");
                }
        }

        // A user is saving updated registration information
        else if ($mode == "save")
        {
                verify_data ();
                if (!empty ($errors))
                {
                        echo "Errors were detected. Please correct before continuing:<br/>";
                        echo ul ($errors);
                        echo href ("$current_script", "Go to your account.");
                }
                else
                {
                        // We might be saving information for another user that isn't the current user
```

```
                        // $post_username will be the current user if not admin since it isn't editable.
                        $admin_user = $dbinfo->username ();
                        if (empty ($is_admin))
                                $is_admin = 0;
                        if ($dbinfo->debug ())
                        {
                                cout ("Debug enabled.");
                                cout ("update user set password = '$passwd1', is_admin = $is_admin,
realname = '$post_realname', birth_date = '$post_birth_date', shipping_street = '$post_street',
shipping_city = '$post_city',
shipping_state = '$post_state', shipping_zip = $post_zip, phone = '$post_phone', email = '$post_email',
card_type = '$post_card_type',
card_number = $post_card_number, card_expire = '$post_card_expire' where username =
'$post_username'");
                        }
                        $dbinfo->query ("update user set password = '$passwd1', is_admin = $is_admin,
realname = '$post_realname', birth_date = '$post_birth_date', shipping_street = '$post_street',
shipping_city = '$post_city',
shipping_state = '$post_state', shipping_zip = $post_zip, phone = '$post_phone', email = '$post_email',
card_type = '$post_card_type',
card_number = $post_card_number, card_expire = '$post_card_expire' where username =
'$post_username'");
                        if ($post_username != $admin_user)
                        {
                                // save activity for both other user and current user
                                $dbinfo->save_activity_for ($post_username, "Admin '$admin_user'
updated your registration information.");
                                $dbinfo->save_activity ("You updated the registration information of
'$post_username'.");
                        }
                        else
                        {
                                // save current user activity
                                $dbinfo->save_activity ("You updated your registration information.");
                        }
                        // display status
                        cout ("Update successful.");
                        echo href ("$current_script?mode=view&username=$post_username", "Click to
refresh");
                }
        }
        // browse a list of registered users
        else if ($mode == "browse" && $dbinfo->is_admin ())
        {
                echo h3 ("All Accounts");
                echo_div ("scriptstatus");
                echo href ("$current_script?mode=view", "View Your Account");
                end_div ();
```

```
		$curr_user = $dbinfo->username ();
		$result = $dbinfo->query ("select u.username, is_admin, realname, email, day, hour,
minute
from user u, user_activity ua
where u.username = ua.username AND
u.username != '$curr_user' AND
ua.activity = 'Registered' order by username");
		if ($result)
		{
			$table = new table_common_t ();
			$table->init ("tbl_std");
			echo $table->table_begin ();
			echo $table->table_head_begin ();
			echo $table->tr ($table->td_span ("Registered users of iBay", "", 6));
			echo $table->tr ($table->td ("Username").
					$table->td ("Realname").
					$table->td ("Admin?").
					$table->td ("Email Address").
					$table->td ("Registration Time").
					$table->td ("Manage"));
			echo $table->table_head_end ();
			echo $table->table_body_begin ();
			while (list($username, $is_admin, $name, $email, $day, $hour, $min) =
mysql_fetch_row ($result))
			{
				echo $table->tr ($table->td ($username).
						$table->td ($name).
						$table->td (($is_admin)? "Yes" : "No").
						$table->td ($email).
						$table->td (format_time ($day, $hour, $min)).
						$table->td (href ("$current_script?
mode=view&username=$username", "Account").
							" | ".
							href ("profile.php?
mode=view&username=$username", "Profile").
							" | ".
							href ("$current_script?
mode=delete&username=$username", "Delete").
							"<br/>".
							href ("$current_script?
mode=login_as&username=$username", "Login As")));
			}
			echo $table->table_body_end ();
			echo $table->table_end ();
			mysql_free_result ($result);
		}
		else if ($dbinfo->debug ())
		{
			mysql_free_result ($result);
```

```
                        die (mysql_error ());
                }
                else
                {
                        mysql_free_result ($result);
                        echo "Couldn't get user information.";
                        return;
                }

        }
        else
        {       // browse a registered user's registration information
                if ($dbinfo->is_admin ())
                {
                        echo_div ("scriptstatus");
                        echo href ("$current_script?mode=browse", "Browse all");
                        if ($user != $dbinfo->username ())
                        {
                                echo " | ";
                                echo href ("$current_script?mode=view", "View Your Account");
                        }
                        end_div ();
                }
                registration_form ($user);
        }
}
else // User is not logged in, display new registration page or save registration
{
        if ($mode == "savenew") // save registration for user
        {
                verify_data ();
                if (!empty ($errors))
                {
                        echo "Errors were detected. Please correct before continuing:<br/>";
                        echo ul ($errors);
                        echo href ("$current_script", "Go to registration.");
                }
                else
                {
                        // insert user into the database
                        $is_admin = 0;
                        $dbinfo->query ("insert into user values ('$post_username', '$passwd1',
$is_admin, '$post_realname', '$post_birth_date', '$post_street', '$post_city', '$post_state', $post_zip,
'$post_phone', '$post_email', '$post_card_type', $post_card_number, '$post_card_expire',
'$post_picture', '$post_description')");
                        if (!$dbinfo->user_exists ($post_username))
                        {
                                cout ("Registration failed. ");
                                cout (href (current_script (), "Try again."));
```

```
                        }
                        else
                        {
                                $dbinfo->save_registration ($post_username);
                                $dbinfo->login_as ($post_username);
                                cout ("Registration successful.");
                                cout ("Click to automatically login and continue to your ".href
("index.php", "home")."?");
                        }
                }
        }
        else // display empty form for user to fill out
                registration_form ();
}

echo_footer ($dbinfo);
?>
```

table_common.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Table related common functions/class (for styles)




class table_common_t
{
/*      for ($i = 0; $i < mysql_num_fields ($result); $i++) */
/*              td (mysql_field_name ($result, $i)); */
/*              for ($i = 0; $i < mysql_num_fields ($result); $i++) */
/*              { */
/*                      $field_name = mysql_field_name ($result, $i); */
/*                      td ($row[$field_name]); */
/*              } */

        var $main_class;

        // initialize to use certain style
        function init ($main_class_)
        {
                $this->main_class = $main_class_;
                return $this->main_class;
        }

        // set to use certain style
        function set_main_class ($main_class_)
        {
                return $this->init ($main_class_);
        }
```

```
// get which style it is using
function get_main_class ()
{
	return $this->main_class;
}

// begin the html table
function table_begin ()
{
	return "<table class=\"$this->main_class\">\n";
}

// end the html table
function table_end ()
{
	return "</table>\n";
}

// begin the html table head
function table_head_begin ($custom_class = "")
{
	if (empty ($custom_class))
		return "<thead class=$this->main_class>\n";
	else
		return "<thead class=$custom_class>\n";
}

// end the html table head
function table_head_end ()
{
	return "</thead>\n";
}

// begin the html table body
function table_body_begin ($custom_class = "")
{
	if (empty ($custom_class))
		return "<tbody class=$this->main_class>\n";
	else
		return "<tbody class=$custom_class>\n";
}

// end the html table body
function table_body_end ()
{
	return "</tbody>\n";
}
```

```
	// html table row
	function tr ($s, $custom_class = "")
	{
		if (empty ($custom_class))
			return "\t<tr class=$this->main_class>\n\t$s\n\t</tr>\n";
		else
			return "\t<tr class=$custom_class>\n\t$s\n\t</tr>\n";
	}

	// html table row begin
	function tr_begin ($custom_class = "")
	{
		if (empty ($custom_class))
			return "\t<tr class=$this->main_class>\n";
		else
			return "\t<tr class=$custom_class>\n";
	}

	// html table row end
	function tr_end ()
	{
		return "\t</tr>\n";
	}

	// html table data
	function td ($s, $custom_class = "")
	{
		return $this->td_span ($s, $custom_class);
	}

	// html table data which spans several columns
	function td_span ($s, $custom_class = "", $span = 0, $align = "")
	{
		if (!empty ($align))
			$align = "align=$align";

		if ($span == 0)
		{
			if (empty ($custom_class))
				return "\t<td $align class=$this->main_class>\n\t$s\n\t</td>\n";
			else
				return "\t<td $align class=$custom_class>\n\t$s\n\t</td>\n";
		}
		else
		{
			if (empty ($custom_class))
				return "\t<td $align class=$this->main_class
colspan=$span>\n\t$s\n\t</td>\n";
			else
```

```
                return "\t<td $align class=$custom_class colspan=$span>\n\t$s\n\t</td>\
n";
            }
        }

}

?>
```

viewtable.php

```
<?php
/*
CMPS460 Database Project
Group I
April 20, 2009

Authors:
 - Trey Alexander      (txa4895)
 - Dallas Griffith     (dlg5367)
 - Mark McKelvy        (jmm0468)
 - Sayooj Valsan       (sxv6633)

~~~ CERTIFICATION OF AUTHENTICITY ~~~
The code contained within this script is the combined work of the above mentioned authors.
*/

// Helpful page for admin to see the data in the db.




include_once ("common.php");

$dbinfo = new dbinfo_t ();
echo_header ($dbinfo);

$mode = get ("mode");
$name = get ("name");
$order = get("order");
$desc = get("desc");
$current_script = current_script ();

function display_message ()
{
        $message = get ("msg");
        echo "Messages:<br/>";
        echo ul (li ($message));
}

if ($dbinfo->logged_in () && $dbinfo->is_admin ())
{
        if ($name == "reload") // reload the database - kind of hacked but oh well
        {
                passthru ("/pkgs2/mysql/bin/mysql -h calvados.ucs.louisiana.edu -u cs4601i -pfoursixty
cs4601_i < ../data/db.dump.sql");
```

```
                cout ("Done.");
                echo href ("index.php", "Refresh");
        }
        else if ($name == "user") // view user table
        {
                echo h3 ("User Data");
                echo_div ("scriptstatus");
                echo href ("$current_script?name=user&order=username&desc=0", "Sort Ascending
(Usernames)");
                echo " | ";
                echo href ("$current_script?name=user&order=username&desc=1", "Sort Descending
(Usernames)");
                end_div ();

                // Select the database

                if ($order == "username")
                {
                        if ($desc == "0")
                        {
                                $query="select * from $name order by username";
                        }

                        else if ($desc == "1")
                        {
                                $query="select * from $name order by username desc";
                        }
                        else if ($desc != "0"|"1")
                        {
                                $query="select * from $name order by username";
                        }
                }

                else $query="select * from $name";

                // Run the query
                $results_id = mysql_query($query);
                if($results_id)
                {
                        // Get each row of the result, assign a variable to each
                        // attribute in the row, and echo the data with labels
                        while (list($username, $passwd, $is_admin, $name, $dob, $street,
                                $city, $state, $zip, $phone, $email, $cc, $ccn, $ccx,
                                $picture, $descr)
                            = mysql_fetch_row($results_id))
                        {
                                // Start writing table to page
                                $table = new table_common_t ();
                                $table->init ("user_database");
```

```
                        echo $table->table_begin ();
                        echo $table->table_head_begin ();
                        echo $table->tr ($table->td_span ("Registered User of iBay", "", 6));
                        echo $table->tr_end ();
                        echo $table->table_head_end ();

                        echo $table->table_body_begin ();
                        echo $table->tr ($table->td ("Username").
                                        $table->td ($username."<br/>".href ("profile.php?
mode=view&username=$username", "Profile")));
                        echo $table->tr ($table->td ("Password").
                                        $table->td ($passwd));
                        echo $table->tr ($table->td ("Admin?").
                                        $table->td ($is_admin));
                        echo $table->tr ($table->td ("Name").
                                        $table->td ($name));
                        echo $table->tr ($table->td ("D.O.B.").
                                        $table->td ($dob));
                        echo $table->tr ($table->td ("Street").
                                        $table->td ($street));
                        echo $table->tr ($table->td ("City").
                                        $table->td ($city));
                        echo $table->tr ($table->td ("State").
                                        $table->td ($state));
                        echo $table->tr ($table->td ("Zip").
                                        $table->td ($zip));
                        echo $table->tr ($table->td ("Phone").
                                        $table->td ($phone));
                        echo $table->tr ($table->td ("Email").
                                        $table->td ($email));
                        echo $table->tr ($table->td ("Credit Card").
                                        $table->td ($cc));
                        echo $table->tr ($table->td ("CC Number").
                                        $table->td ($ccn));
                        echo $table->tr ($table->td ("CC Exp.Date").
                                        $table->td ($ccx));
                        echo $table->tr ($table->td ("Picture").
                                        $table->td (local_img ($picture)));
                        echo $table->tr ($table->td ("Description").
                                        $table->td ($descr));
                        echo $table->table_body_end ();

                        echo $table->table_end ();
                        echo "<br><br>";
                }
        }
        else if ($dbinfo->debug ())
        {
```

```
                        // Display the query and the MySQL error message
                        print "<br><br>QUERY FAILED !!! <br><br>QUERY = $query <br>
                <br>ERROR = ";
                        die (mysql_error());
                }
        }

        else if ($name == "user_activity") // view the activity table
        {
                echo h3 ("User Activity Data");
                echo_div ("scriptstatus");
                echo href ("$current_script?name=user_activity&order=username", "Sort by
username");
                echo " | ";
                echo href ("$current_script?name=user_activity&order=time", "Sort by time");
                end_div ();

                // Select the database
                if (empty ($order) || $order == "time")
                        $query="select * from $name order by day desc, hour desc, minute desc";
                else
                        $query="select * from $name order by username";

                // Run the query
                $results_id = mysql_query($query);
                if($results_id)
                {
                        // Get each row of the result, assign a variable to each
                        // Start writing table to page
                        $table = new table_common_t ();
                        $table->init ("user_activity_database");

                        echo $table->table_begin ();
                        echo $table->table_head_begin ();
                        echo $table->tr ($table->td_span ("Activity Log Entry", "", 3));
                        echo $table->tr ($table->td ("Username").
                                         $table->td ("Time").
                                         $table->td ("Activity"));
                        echo $table->table_head_end ();
                        echo $table->table_body_begin ();

                        // attribute in the row, and echo the data with labels
                        while (list($username, $day, $hour, $minute, $activity)
                              = mysql_fetch_row($results_id))
                        {
                                echo $table->tr (
                                        $table->td (href ("profile.php?
mode=view&username=$username", $username)).
                                        $table->td (format_time ($day, $hour, $minute)).
```

```
                                        $table->td ($activity));
                        }
                        echo $table->table_body_end ();
                }
                else if ($dbinfo->debug ())
                {
                        // Display the query and the MySQL error message
                        print "<br><br>QUERY FAILED !!! <br><br>QUERY = $query <br>
        <br>ERROR = ";
                        die (mysql_error());
                }
        }

        else if ($name == "item_listing") // view the item listing table
        {
                echo h3 ("Item Listing Data");
                echo_div ("scriptstatus");
                echo href ("$current_script?name=item_listing&order=seller&desc=0", "Sort by seller
(ascending)");
                echo " | ";
                echo href ("$current_script?name=item_listing&order=seller&desc=1", "Sort by seller
(descending)");
                end_div ();

                // Select the database

                if ($order == "seller")
                {
                        if ($desc == "0")
                        {
                                $query="select * from $name order by seller";
                        }

                        else if ($desc == "1")
                        {
                                $query="select * from $name order by seller desc";
                        }
                        else if ($desc != "0"|"1")
                        {
                                $query="select * from $name order by seller";
                        }
                }

                else $query="select * from $name";

                // Run the query
                $results_id = mysql_query($query);
                if($results_id)
                {
```

```
                        // Get each row of the result, assign a variable to each
                        // attribute in the row, and echo the data with labels
                        while (list($title, $seller, $category, $end_day, $end_hour,
                                $end_min, $descr, $ship_cost, $ship_meth, $str_price,
                                $curr_price, $pict, $buyer, $buy_fdbk_descr,
                                $buy_fdbk_rate, $sell_fdbk_descr)
                            = mysql_fetch_row($results_id))
                        {
                                // Start writing table to page
                                $table = new table_common_t ();
                                $table->init ("item_listing_database");

                                echo $table->table_begin ();
                                echo $table->table_head_begin ();
                                echo $table->tr ($table->td_span ("Auction Item Listing", "", 6));
                                echo $table->tr_end ();
                                echo $table->table_head_end ();

                                echo $table->table_body_begin ();
                                echo $table->tr ($table->td ("Title").
                                                $table->td (href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title)));
                                echo $table->tr ($table->td ("Seller").
                                                $table->td (href ("profile.php?
mode=view&username=$seller", $seller)));
                                echo $table->tr ($table->td ("Category").
                                                $table->td (href ("category.php?view=$category",
$category)));
                                echo $table->tr ($table->td ("End Time").
                                                $table->td (format_time ($end_day, $end_hour,
$end_min)));
                                echo $table->tr ($table->td ("Description").
                                                $table->td ($descr));
                                echo $table->tr ($table->td ("Shipping Cost").
                                                $table->td ("\$$ship_cost"));
                                echo $table->tr ($table->td ("Shipping Method").
                                                $table->td ($ship_meth));
                                echo $table->tr ($table->td ("Starting Price").
                                                $table->td ("\$$str_price"));
                                echo $table->tr ($table->td ("Current Price").
                                                $table->td ("\$$curr_price"));
                                echo $table->tr ($table->td ("Picture").
                                                $table->td (local_img ($pict)));
                                echo $table->tr ($table->td ("Buyer").
                                                $table->td ($buyer));
                                if ($buy_fdbk_rate != -1)
                                {
                                        echo $table->tr ($table->td ("Buyer Feedback Description").
```

```
                                        $table->td ($buy_fdbk_descr));
                                echo $table->tr ($table->td ("Buyer Feedback Rating").
                                        $table->td ($buy_fdbk_rate));
                        }
                        else
                        {
                                echo $table->tr ($table->td ("Buyer Feedback Description").
                                        $table->td ("None."));
                        }

                        if (!empty ($sell_fdbk_descr))
                        {
                                echo $table->tr ($table->td ("Seller Feedback Description").
                                        $table->td ($sell_fdbk_descr));
                        }
                        else
                        {
                                echo $table->tr ($table->td ("Seller Feedback Description").
                                        $table->td ("None."));
                        }
                        echo $table->table_body_end ();

                        echo $table->table_end ();
                        echo "<br><br>";
                }
        }
        else if ($dbinfo->debug ())
        {
                // Display the query and the MySQL error message
                print "<br><br>QUERY FAILED !!! <br><br>QUERY = $query <br>
    <br>ERROR = ";
                die (mysql_error());
        }
}

else if ($name == "bids_on") // view the bids on table
{
        if ($mode == "delete")
        {
                $user = get ("username");
                $title = get ("title");
                $seller = get ("seller");
                $category = get ("category");
                $end_day = get ("end_day");
                $end_hour = get ("end_hour");
                $end_minute = get ("end_minute");
                $bid_day = get ("bid_day");
                $bid_hour = get ("bid_hour");
                $bid_minute = get ("bid_minute");
```

```
                        $dbinfo->query ("delete from bids_on
where  username = '$user'
AND   item_title = '$title'
AND   item_seller = '$seller'
AND   item_category = '$category'
AND   item_end_day = $end_day
AND   item_end_hour = $end_hour
AND   item_end_minute = $end_minute
AND   bid_day = $bid_day
AND   bid_hour = $bid_hour
AND   bid_minute = $bid_minute");
                }

                echo h3 ("Bidding Data");
                echo_div ("scriptstatus");
                echo href ("$current_script?name=bids_on&order=username&desc=0", "Sort by
username (ascending)");
                echo " | ";
                echo href ("$current_script?name=bids_on&order=username&desc=1", "Sort by
username (descending)");
                end_div ();

                // Select the database
                if ($order == "username")
                {
                        if ($desc == "0")
                        {
                                $query="select * from $name order by username";
                        }

                        else if ($desc == "1")
                        {
                                $query="select * from $name order by username desc";
                        }
                        else if ($desc != "0"|"1")
                        {
                                $query="select * from $name order by username";
                        }
                }

                else $query="select * from $name";

                // Run the query
                $results_id = mysql_query($query);
                if($results_id)
                {
                        $table = new table_common_t ();
                        $table->init ("bidding_database");
```

```
                        echo $table->table_begin ();
                        echo $table->table_head_begin ();
                        echo $table->tr ($table->td_span ("Bid Information", "", 8));
                        echo $table->tr ($table->td ("Bidder").
                                         $table->td ("Title / Category").
                                         $table->td ("Seller").
                                         $table->td ("End Time / Bid Time").
                                         $table->td ("Amount").
                                         $table->td ("Notify").
                                         $table->td ("Modify"));
                        echo $table->table_head_end ();
                        echo $table->table_body_begin ();

                        // Get each row of the result, assign a variable to each
                        // attribute in the row, and echo the data with labels
                        while (list($username, $title, $seller, $category, $end_day,
                                    $end_hour, $end_min, $bid_day, $bid_hour, $bid_min,
                                    $bid_amt, $disp_notif)
                               = mysql_fetch_row($results_id))
                        {
                                if ($disp_notif != "n")
                                {
                                        $can_delete = $table->td (href ("$current_script?
name=$name&order=$order&desc=$desc&mode=delete&username=$username&title=$title&seller=$
seller&category=$category&end_day=$end_day&end_hour=$end_hour&end_minute=$end_min&bid_
day=$bid_day&bid_hour=$bid_hour&bid_minute=$bid_min", "X"));
                                }
                                else
                                {
                                        $can_delete = $table->td ("-");
                                }

                                // Start writing table to page
                                echo $table->tr ($table->td (href ("profile.php?
mode=view&username=$username", $dbinfo->get_realname ($username))).
                                                 $table->td (href ("itemlisting.php?
mode=view&title=$title&seller=$seller&category=$category&end_day=$end_day&end_hour=$end_h
our&end_minute=$end_min", $title).
                                                                 "<br/>(".
                                                                 href ("category.php?view=$category",
$category).")").
                                                 $table->td (href ("profile.php?
mode=view&username=$seller", $dbinfo->get_realname ($seller))).
                                                 $table->td (format_time ($end_day, $end_hour,
$end_min).
                                                                 "/<br/>".
                                                                 format_time ($bid_day, $bid_hour, $bid_min)).
                                                 $table->td ("\$$bid_amt").
                                                 $table->td ($disp_notif).
```

```
                                        $can_delete);
                }
                echo $table->table_body_end ();

                echo $table->table_end ();
                echo "<br><br>";
            }
            else if ($dbinfo->debug ())
            {
                // Display the query and the MySQL error message
                print "<br><br>QUERY FAILED !!! <br><br>QUERY = $query <br>
        <br>ERROR = ";
                die (mysql_error());
            }
        }
        else
        {
            echo "Errors were detected:<br/>";
            echo ul (li ("Invalid table name '$name'."));
            cout ("Back to your ".href ("index.php", "home").".");
        }
}
else
{
        redirect ("index.php");
}

echo_footer ($dbinfo);
?>
```

```
drop table if exists bids_on;
drop table if exists item_listing;
drop table if exists user_activity;
drop table if exists user;

create table user(
	username		varchar(50),
	password		varchar(50),
	is_admin		boolean,
	realname		varchar(100),
	birth_date	varchar(10),
	shipping_street	varchar(100),
	shipping_city	varchar(50),
	shipping_state	varchar(25),
	shipping_zip	varchar(10),
	phone		varchar(12),
	email		varchar(50),
	card_type	varchar(20),
	card_number	varchar(16),
	card_expire	varchar(5),
	picture		varchar(100),
	description		varchar(250),
	primary key (username)
);

create table user_activity(
	username		varchar(50),
	day		int(5),
	hour		int(2),
	minute		int(2),
	activity		varchar(250),
	primary key (username, day, hour, minute),
	foreign key (username) references user (username)
);

create table item_listing(
	title		varchar(50),
	seller		varchar(50),
	category		varchar(30),
	end_day		int(5),
	end_hour		int(2),
	end_minute		int(2),
	description		varchar(250),
	shipping_cost	float(7,2),
	shipping_method	varchar(100),
	starting_price	float(7,2),
	current_price	float(7,2),
	picture		varchar(100),
```

```
    buyer                varchar(50),
    buyerfeedbackforseller_description      varchar(250),
    buyerfeedbackforseller_rating           int(2),
    sellerfeedbackforbuyer_description      varchar(250),
    primary key (title, seller, category, end_day, end_hour, end_minute),
    foreign key (seller) references user (username)
);

create table bids_on(
    username            varchar(50),
    item_title varchar(50),
    item_seller         varchar(50),
    item_category       varchar(30),
    item_end_day        int(5),
    item_end_hour       int(2),
    item_end_minute int(2),
    bid_day             int(5),
    bid_hour            int(2),
    bid_minute          int(2),
    bid_amount          float(7,2),
    display_notification        char(1),
    primary key (username, item_title, item_seller, item_category,
                item_end_day, item_end_hour, item_end_minute,
                bid_day, bid_hour, bid_minute),
    foreign key (username) references user (username),
    foreign key (item_title) references item_listing (title),
    foreign key (item_seller) references item_listing (seller),
    foreign key (item_category) references item_listing (category),
    foreign key (item_end_day) references item_listing (end_day),
    foreign key (item_end_hour) references item_listing (end_hour),
    foreign key (item_end_minute) references item_listing (end_minute)
);
```