Table of Contents

category.php

This script is responsible displaying items in a category listing. It will need to handle the scenarios listed in the following sections.

The script will check the GET variables “view”, “sortby”, “desc”.

The script will make sure the current user is logged in.

- If “view” is empty – display a listing of the available categories with the links (example: category.php?view=art)
- If “view” is non empty, make sure that its value is one of the predefined categories, otherwise display an error message.
- If “sortby” is empty, the listing will be sorted by some default preference. If it is not empty, then sort by the fieldd specified in “sortby”. Also check “desc” to know whether to sort descending or ascending.
- When displaying the listings from a category, refer to the queries from the queries document named “View item listings in a category” to know which things to display.
- Should be able to sort by time remaining, title, seller, or current bid.
- For each entry, there should be a link to view the details of the item listing, this will be a URL with the GET information containing the primary keys of an item listing. For example: itemlisting.php?mode=view&title=$title&seller=$seller&category=$category&end_day=$endday&end_hour=$endhour&end_minute=$endminute
- If the current user is an admin, display an extra 2 columns where one is for “edit” and one is for “delete”.
    - This will be similar to the URL specified above, except “mode” would be “edit” or “delete”.
- Include a URL near the top of the page to create a new item listing. The URL would be something like itemlisting.php?mode=create

common.php

To be included in all other scripts, has common functions like echo_header, echo_footer, etc.

dbinfo.php

Script which contains the dbinfo_t class for handing a lot of the queries on the database with a generic query() function as well as specific functions like user_exists($username).

form_common.php

Functions for generating form html code easily.

index.php

- Display all notifications, feedback, and user activity for current user all ordered by time.
- Other stuff.......

itemlisting.php

This script is responsible displaying items detailed information as well as creation and editing of items.

The script will check the GET variables:

- “mode”
- “title”, “seller”, “category”, “end_day”, “end_hour”, “end_minute”

The script will make sure the current user is logged in.

- If “mode” is empty or “create”, then allow the user to create a new item listing.
  - Should be able to choose from a predefined list of categories
  - Choose picture from a list
  - Choose end time from a list (say a bound from one day in advance to one week in advance)
  - Set some default starting price
  - Allow choosing shipping method and shipping price from a list.
- If “mode” is “view”, then using the GET variables “title”, “seller”, “category”, “end_day”, “end_hour”, “end_minute” display information for the item listing.
  - Also show bid history
  - Also show a bid link if “seller” isn't the same as the current user and the auction hasn't closed. Bid link will be a get query: itemlisting.php?mode=bid&title=$title&seller=$seller&category=$category&end_day=$endday&end_hour=$endhour&end_minute=$endminute
  - Also show a edit link and a delete link, forming the correct GET query in the URL.
- If “mode” is “edit”, and current user is not an admin, allow editing of the entry defined by “title”, current user, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.
- If “mode” is “edit” and current user is an admin and “seller” is empty, allow editing of the entry defined by “title”, current user, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.
- If “mode” is “edit” and current user is an admin and “seller” is not empty, allow editing of the entry defined by “title”, “seller”, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.
- If “mode” is “delete”, and current user is not an admin, allow deleting of the entry defined by “title”, current user, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.

- If “mode” is “delete” and current user is an admin and “seller” is empty, allow delete of the entry defined by “title”, current user, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.
- If “mode” is “delete” and current user is an admin and “seller” is not empty, allow delete of the entry defined by “title”, “seller”, “category”, “end_day”, “end_hour”, “end_minute”. Follow rules listed in queries document.
- If “mode” is “bid”, and the POST “bid_amount” is empty, show a bid form with a selection box ranging from higher than current value to “a lot” higher than current value. The link will be itemlisting.php?mode=bid&title=$title&seller=$seller&category=$category&end_day=$endday&end_hour=$endhour&end_minute=$endminute
- If “mode” is “bid”, and the POST “bid_amount” is not empty, take care of the rules to introduce a new bid.

list_common.php

Functions for generating html list code easily.

login.php

Handles user login to website.

logout.php

Handles user logout of the website.

profile.php

This script is responsible handling all things related to user profiles. It will need to handle the scenarios listed in the following sections.

The script will check the GET variables “mode”, “username”.

The script will make sure the current user is logged in.

Default view

If it is “mode” is empty, behave the same as “Display current logged in user's profile”.

Display a listing of all user profiles

- The value of “mode” is “browse”
- Display a table that has the columns of information:
    - Username (which is a link like “profile.php?mode=view&username=$username”)
    - Username's registration date.

  - Username's last time to visit the site (activity = 'Logged Out')
- (Verify from the requirements that we don't need to display anything more for this view)

Display current logged in user's profile

- The value of “mode” is (“view” or empty) and “username” is empty.
- Display the current logged in user's profile picture and description.
- Also show a listing of all feedback for the given user, either as buyer feedback for seller or vice versa. This listing is ordered by time.
- Also show a listing of all bidding activity for the user, sorted by time.
- Display a link for editing the current user's profile “profile.php?mode=edit”

Display another registered user's profile

- The value of “mode” is (“view” or empty) and “username” is not empty.
- Check that “username” exists in the database. ($dbinfo->user_exists(get(“username”)))
- If “username” does not exist, display some form of error.
- If “username” does exist:
  - Display “username”'s profile picture and description.
  - Also show a listing of all feedback for the “username”, either as buyer feedback for seller or vice versa. This listing is ordered by time.
  - Also show a listing of all bidding activity for “username”, sorted by time.
  - If the current logged in user is an administrator:
    - Display a link for editing the current user's profile “profile.php?mode=edit&username=$username”

Edit the current user's profile

- The value of “mode” is “edit” and the value of “username” is empty.
- Allow changing of the profile picture and profile description.
- When the user clicks save, send information through post.
- The form action should be the same as the get query you received: “mode” = “edit”.
- Update the database.
- Save activity

Edit another user's profile

- Current user must be an admin ($dbinfo->is_admin())
- The value of “mode” is “edit” and the value of “username” is not empty.

- Verify “username” is a valid user in the database.
- Perform similar steps for “Edit the current user's profile”.

Delete/reset profile picture and description

- The value of “mode” = “delete”. If the value of “username” is not empty, then verify the current user is an admin.
- Update the profile picture to default.jpg and description to 'N/A'.
- Save activity (if an admin, save activity for both users).

registration.php

This script is responsible for handling user registrations.

The script will check the GET variables “mode”, “username”.

- If the current user is not logged in, the script will allow the registration of a new unique username.
- If the current user is logged in and not an admin:
  - If “mode” is “edit”, allow the user to edit their registration information according to the requirements document. Save activity.
  - Else, display current user's registration information.
- If the current user is logged in and is an admin:
  - If “mode” is “view” & “username” is empty, display the current user's registration information with a link to be able to edit it.
  - If “mode” is “view” & “username” is not empty, verify user exists and display the user's registration information with a link to be able to edit it.
  - If “mode” is “edit” & “username” is empty, allow the user to edit the registration information of the current user according to the requirements document. Save activity.
  - If “mode” is “edit” & “username” is not empty, verify user exists and allow the user to edit the registration information according to the requirements document. Save activity.
  - If “mode” is “delete”, verify “username” is not empty and follow the queries document for rules of deletion.
  - If “mode” is “viewall”, display a list similar to the profile listing.

table_common.php

Functions for generating html tables and associated style name easily.

viewtable.php

This script is responsible for displaying the data from the database tables:

- user
- user_activity
- item_listing
- bids_on

This script should check that the user is logged in and that the user is an administrator. Both of these should be true, otherwise the user should be redirected to index.php

This script will check the GET variables “name”, “order”, and “desc”.

Default view

The GET variable “name” → if it is empty, then the script should display a means for the user to view one of the tables from the database (“user”, “user_activity”, “item_listing”, “bids_on”). This can be done with a list of urls that form the get query with “name” equal to the correct table name.

User table view

The value of “name” should be “user”. A query will be formed to get information from the “user” table and display it to the page.

The query is dependent on the values of the “order” and “desc” GET variables. If “order” is empty then the query will have some default ordering of the data. Otherwise, the query should order by the field name given. If the GET variable “desc” is empty or 0, then the query will order by the specified field ascending. Otherwise, order by descending. (For example: “select * from $table order by $order desc" vs. “select * from $table order by $order").

The user should be able to sort by the username field.

The username fields should be a link where the url is like: profile.php?
mode=view&username=$username

User Activity table view

The value of “name” should be “user_activity”. A query will be formed to get information from the “user_activity” table and display it to the page.

The query is dependent on the values of the “order” and “desc” GET variables. If “order” is empty then the query will have some default ordering of the data. Otherwise, the query should order by the field name given. If the GET variable “desc” is empty or 0, then the query will order by the specified field ascending. Otherwise, order by descending. (For example: “select * from $table order by $order desc" vs. “select * from $table order by $order").

The user should be able to sort by the username field and the time fields.

Auction table view

The value of “name” should be “item_listing”. A query will be formed to get information from the “item_listing” table and display it to the page.

The query is dependent on the values of the “order” and “desc” GET variables. If “order” is empty then the query will have some default ordering of the data. Otherwise, the query should order by the field name given. If the GET variable “desc” is empty or 0, then the query will order by the specified field ascending. Otherwise, order by descending. (For example: “select * from $table order by $order desc" vs. “select * from $table order by $order").

The user should be able to sort by the following fields: title, seller, category, “end time”, current_price, starting_price, buyerfeedbackforseller_rating.

Bidding table view

The value of “name” should be “bids_on”. A query will be formed to get information from the “bids_on” table and display it to the page.

The query is dependent on the values of the “order” and “desc” GET variables. If “order” is empty then the query will have some default ordering of the data. Otherwise, the query should order by the field name given. If the GET variable “desc” is empty or 0, then the query will order by the specified field ascending. Otherwise, order by descending. (For example: “select * from $table order by $order desc" vs. “select * from $table order by $order").

The user should be able to sort by the following fields: username, item_title, item_seller, item_category, item “end time”, bid “time”, bid amounts, display_notification.