MVC-Database Assignment

# CSCI 4300

**Introduction**: Create a Web application for an online auction. As discussed in class, there will be two user roles in this application:

* **Guest**: a guest does not need to be authenticated. Guests can view all the auction items and place bids on the item.
* **Admin**: the administrative user must be authenticated. The admin can add new auction items and view the current top bids on all items.

I have provided a partial mockup of the Guest view. Check out the sliding index in the left panel! To get full credit, your solution must follow this pattern, in which the user simultaneously sees the brief titles of all the items plus the full description and input form for a single item.

Note that the items are divided into categories. The easiest way to implement this is to have a separate DB table for the categories.

**Scope Limitations**: The following useful functionality will NOT be part of this assignment:

* Payment processing
* Image file uploading (assume the admin can upload files via SCP)
* Protection against frivolous bids
* Protection against non-numeric bids
* Adding new users
* Changing the admin password
* Managing the categories

Your application should, however, reject bids that are not larger than the current largest bidder (or the minimum bid) and bids for which the input form is not completely filled out *(e.g.* user does not provide their email address).

Before adding additional cool features, please ask me! The best solution is the minimal solution that supports the example view.

**Implementation**: use the Model-View-Controller framework. Provide a single controller class **servlets.AuctionController** and a single helper class **helper.AuctionHelper**.

**Deliverables**: as part of the project deliverables , provide a database schema document and the HTML formatted Javadocs for AuctionController and AuctionHelper.

**Deployment**: This app must be built and run on your VM to get full credit. 50 points partial credit will be given for an Eclipse-only solution.

# Grading Rubric:

|  |  |
| --- | --- |
| **Accomplishment** | **Points** |
| Complete guest view | 10 |
| MySQL data model | 10 |
| Admin view | 10 |
| HTML documentation for AuctionController and AuctionHelper | 10 |
| JUnit test code for AuctionHelper | 10 |
| Handles incomplete or too-low bids | 10 |
| Records legitimate bids | 10 |
| Admin login procedure | 10 |
| Admin add item procedure | 10 |
| Admin view bids | 10 |
| **Total** | **100** |

## Grading Deductions:

* Up to 10 points will be deducted for failure to follow the course Coding Guidelines on eLC. Frequent causes are leaving out spec or loop comments.
* 10 points will be deducted if you copy code from your own work or from the book and it the copied code is inappropriate in its new location. (This penalty is intended to discourage you from copying your own code. We learn by writing code, not by copying code).

# Academic Honesty Rules for This Assignment

This is an individual assignment. You may discuss ideas about this assignment with others. You may not help another student with their code, copy code from any other source, work on code together with other students, show your code to other students, or get help from any other person besides the instructor.

Suspected violations of this policy will be dealt with through the UGA Academic Honesty program. In this event, I will request a penalty of an F in the course.