Project overview: *Create an auction web site, where users can post items for sell and/or bid on items*. (Think “eBay”.)

A. The site must contain the following pages:

1. *Main page*. The main page contains:
    a. A link to the registration page and a sign-in form (if the user is not already logged in), or a log-out link (if the user is already signed in).
        i. The sign-in form is directly in the main page; there is no dedicated “sign in” page.
        ii. The sign-out link signs the user out and leads back to the main screen; there is no dedicated “sign out” page.
    b. A list of the four most-recently-posted items, with thumbnail images, and links to the item pages.
    c. Statistics: Number of items currently available for sale; number of sells completed so far.
    d. If the user is logged in, the main page must also include a list of all bids that the user placed: the item name, what was the user’s bid, if the auction is over, and if so, did the user win.
    e. A link to the “post item” page.
    f. A search form (search box + button), leading to the search-results page.
2. *Registration page*. users can register by choosing a (unique) username and a password. Users can also optionally provide a “display name”, which can be different than the username. (For example, my username can be “tal_cohen” and the display name can be “Tal”.) The display name need not be unique.
    a. The username will be used for logging in; when displaying the user’s name, the display name will be used, if available; otherwise, the username will be used.
3. *Post item* (place an item for sell) *page*. This page presents a form, where the user must provide an item name, a description, a sell time (in minutes; default: 10m), and an opening price (default: $0.01). The user can also include a URL for an image; the server will incorporate that image into the “view item” page.
4. *View item page*. This page will show all details about an item, as posted by the seller. It will also show the current top bid for the item, and allow the user (*unless the user is the seller of that item*) to bid on the item. The page will also show any *comments* left for this item, and allow the user to place additional comments.
    a. To bid, the user must provide a maximum bid. If his maximum bid is higher than the current bid, it does *not* automatically mean that this user is winning the auction; see section B below for details.
    b. Any user (including the seller) can post a comment about the item. Comments will be shown in chronological order, including the poster’s name (display name or username) and time of posting.
    c. If the auction for this item is completed, the page will not allow additional bidding. The final, winning price will be shown.

        i. If the user visiting the view-item page is the seller, then the page will also show the winning user's name (including username, not just display-name, since display name is not unique!)
        ii. If the user visiting the view-item page had placed a bid on this item, the page will indicate if the user won this auction or not.

5. *Search results page.* Used when the user searches from the main page. Shows all items that contain the search-string in their name and/or description.
    a. Search order is: First, all auction-not-yet-closed items, ordered so that lower prices (current) appear first. Next, all auction-already-completed items, ordered so that higher (final) prices appear first.

B. How bidding works

For each item, there is the current highest bid (let's call it *current_bid*). The user that placed the current bid is *winning_user*. There is also winning_user's maximal bid (call it *max_bid*), which is the bid that the user actually specified in the form.

All users can see current_bid. Only winning_user knows what the maximal bid is; other users (including the seller) do not know this value.

If there was no bid so far, current_bid is the opening price specified by the seller, and max_bid equals the same value.

When a user places a new bid, they specify their own maximal bid; let's call this new_bid.

- If new_bid is lower than or equal to current_bid, the bid is disqualified.
- If new_bid is higher than current_bid, but lower than (or equal to) max_bid, then:
    - winning_user remains unchanged. max_bid remains unchanged.
    - current_bid is updated to new_bid + $0.01.
- If new_bid is higher than max_bid, then:
    - current_bid is updated to max_bid + $0.01.
    - max_bid is updated to new_bid.
    - winning_user is updated to the user that placed the new bid.

For example, if the opening price is $0.50, and user U_1 places a bid for $2.00, then:

- current_bid becomes $0.51,
- max_bid is $2.00, and
- winning_user is set to U_1.

Next, user U_2 visits the page. They see the current bid as $0.51. User U_2 places a bid at $1.65. As a result:

- current_bid becomes $1.66,
- max_bid is remains $2.00, and
- winning_user remains U_1.

U_2 can choose to place a new bid, this time for $2.55. As a result:

- current_bid becomes $2.01,
- max_bid becomes $2.55, and
- winning_user is updated to U_2.

In other words, the winning bidder pays at most one cent more than is required to beat all other bidders. When I place a new bid, higher than what I see as the current winning bid, it does not mean I will become the leading bidder: it may just force the current winning bidder to pay a higher price.

C. Auto-Refreshing Pages

The following pages must auto-refresh based either on timer events or on user interactions. A refresh must happen *without* the entire page being reloaded; rather, only the relevant part(s) of the page will be refreshed.

1. The *main page* will refresh the four-most-recent-items, and the site statistics, every 5 seconds.
2. The *view item page* will refresh every 5 seconds, showing:
   a. Changes to the top bid,
   b. If the auction was completed,
   c. Any new comments posted to the page,
   d. Changes to the time left for this auction (naturally, this is *always* updated until the auction is over).
3. The *search results page* will refresh every 5 seconds, only in case the search results were updated. Changes include: new items were listed, top bids were changed, time left (in minutes) to sell completion was reduced, or items just completed their sell.
   a. Note: If an item has less than one minute left, the time left will be shown in seconds.

For example, let's say the user searched for "speakers". The search result page showed:

- Altec Lansing BX1221, $0.01, 5m left
- Logitec Z506, $1.22, 13m left
- Polk Audio CSi 20, $9.51, 0m20s left
- Klipsch KG2.2, sold for $188.30
- Yamaha NX-430P, sold for $32.00

After five seconds, this could update to:

- Altec Lansing BX1221, **$0.10**, 5m left
- Logitec Z506, $1.22, 13m left
- Polk Audio CSi 20, $9.51, **0m15s** left

- Klipsch KG2.2, sold for $188.30
- Yamaha NX-430P, sold for $32.00

(Changes are in bold). After a few more rounds, this could update to:

- Altec Lansing BX1221, $0.10, **4m** left
- Logitec Z506, $1.22, **12m** left
- Klipsch KG2.2, sold for $188.30
- Yamaha NX-430P, sold for $32.00
- **Polk Audio CSi 20, sold for $9.51**

(Small) bonus if you make all changes (to search and other pages) reveal themselves in a nice, animated way rather than a sharp and clear-cut replacement. Animations can include position and/or color indications of change.

D. Misc requirements

- All data, including user data (usernames, password, etc.) and item data (descriptions, bids, comments, etc.) must be stored in a database.
- The UI should not allow the users to make invalid options (e.g., post empty items, place negative or non-numeric bids, etc.).
- Comments can contain a *limited* amount of HTML. In particular, comments must support:
  - <a href> links.
  - <b> (bold), <i> (italic), and <u> (underline).
- Comments entered by the user can contain unbalanced HTML (e.g., <i> with no matching </i>), but annotations must not leak from one comment to the next, on any browser (e.g., if a comment includes <i> and no matching </i>, the comment after it must not appear in italic).
- Users must not be able to pick too-weak passwords. (You get to pick the password-validity rules; e.g., “at least 5 characters, at least one letter, at least one digit”.)
- JSON responses sent to AJAX requests should be minimized if possible. For example, when the search result page is updated, the JSON response need not include all details about all items, since some of the items are already shown, and are not expected to change (e.g., item descriptions).

E. What I will check for

- Clarity and elegance of code. If your code looks bad, your score will suffer -- even if there are no bugs.
- Security. I will try to mess around with your site, in different ways. You will lose points when I will cause the server to misbehave.
- Proper use of HTML, CSS, JavaScript.
- A random set of the requirements above (perhaps all of them, perhaps not).

- Other things...

F. What I will not check for

- How beautiful your graphic design is. (But try not to make it too ugly. I’m an aesthetically-sensitive person.)

G. Browser Support

Your code can potentially be tested on a recent version of Firefox, a recent version of Chrome, and Internet Explorer version 10 or later. Older versions of these browsers need not be supported.

H. Submission

Your code has to be submitted, as a zip file (no RARs, etc.) by February 20, 23:59, Israel time, in an email sent to me. The email *receiving time* counts -- not the send time. Please send the submissions to my personal email address: tal.ayal.cohen@gmail.com. The code you present must be the code you have mailed me; those presenting at a later date do not get a free extra week to polish their code. (They do get an extra week of anxiety and suspense.)

Your email must include your full names in Hebrew, and your ID numbers.

Please send any questions to ctal@cs.technion.ac.il. Include “[webdev]” in your email subject (but also the actual subject, not just “webdev”, please...).