Homework 4

Due: Tuesday May 28, 11:59pm

The assignment

We will create a site where users can view a list of orders, and leave product reviews.
The site will have two pages: /default/index and /default/products .

The products page

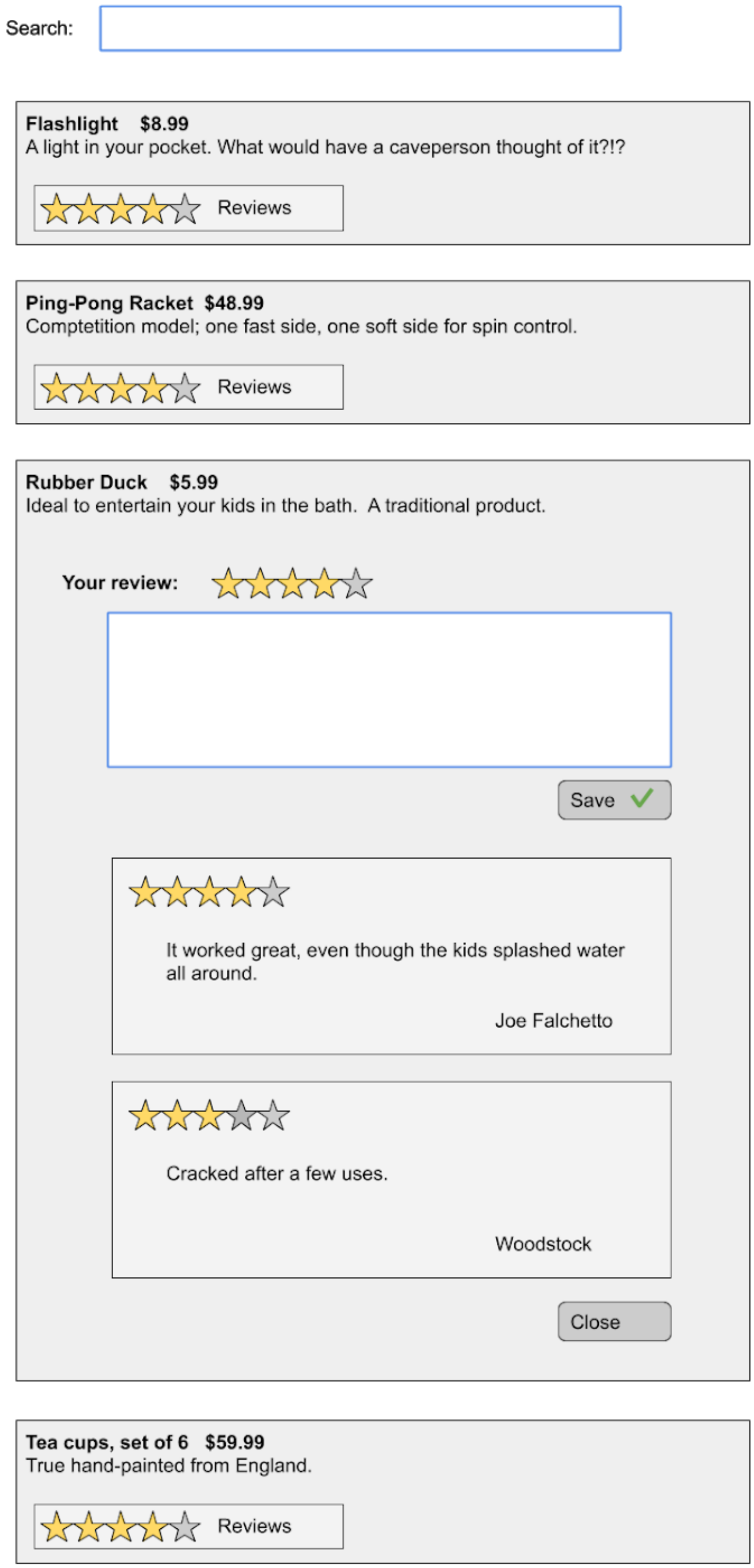
The page at /default/products is a page where, with the help of a SQLFORM.grid, logged in users can maintain (add/edit/delete) a list of products. For each product there has to be:

- Product name
- Product description
- Product price

(and other fields if you wish, but they are not strictly necessary).

The main page (/default/index)

This page *must* be developed via vue.js as a dynamic page, just like the pages we have been seeing recently in class. A template for the page is as follows:



Search:

Flashlight $8.99
A light in your pocket. What would have a caveperson thought of it?!?

★★★★☆ Reviews

Ping-Pong Racket $48.99
Comptetition model; one fast side, one soft side for spin control.

★★★★☆ Reviews

Rubber Duck $5.99
Ideal to entertain your kids in the bath. A traditional product.

Your review: ★★★★☆

Save ✓

★★★★☆

It worked great, even though the kids splashed water all around.

Joe Falchetto

★★★☆☆

Cracked after a few uses.

Woodstock

Close

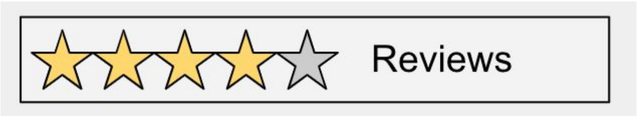
Tea cups, set of 6 $59.99
True hand-painted from England.

★★★★☆ Reviews

Search bar. At the top is a search bar. If one types “Rub”, only products starting with “Rub” should be displayed. Everytime someone adds or removes a character, the list of products must be updated. It’s ok to close the review details when this happens.

Product display. Products are displayed in rectangles, along with the average rating they got. You can round to the nearest star.

Reviews. If someone clicks on the “Reviews” button next to the stars (you can make the stars and the Review, the whole rectangle behave like a button), then the list of reviews is expanded. When one presses on the Close button at the end of the expanded list, the list of reviews is collapsed again.



If the user is logged in, at the top of the list of reviews is a zone where the user can enter or edit his or her review.

- If the user has not reviewed the product yet:

  In the expanded list of reviews, the user can click on stars and enter a review. The stars are saved as soon as they are clicked. The review is saved when the user clicks on Save; upon successful save in the database, a green checkmark flashes in the Save button to provide confirmation that the review has been saved. Thus, reviews and star ratings can be saved independently.

- If the user has already reviewed the product:

  The behavior is similar to above, but the star rating and the input for the product review come pre-filled with the user’s review. The user can edit it and save it.

After the region for the user’s own review, is a region where the reviews by other users are displayed. As username, you can just use the user’s name. Finally, there is a close button that, when pressed, collapses all the reviews for a product.

If the user is not logged in, the special first review where one can enter/edit one’s own review is not there. The list of reviews by users, with their star ratings, is still present.

Note that, if a user sets no star value, then all stars need to be displayed gray, and they should not be counted for the average rating (so the number of stars for a product can be None (indicating no vote), or a number from 1 to 5).

At most one product should have its reviews displayed, so that when you check the reviews for another product, the reviews of the first product should collapse.

The reviews for a product should be loaded dynamically, only when someone clicks on the “Reviews” button for that product. Caching them in the browser is extra credit.

Implementation

Start from the clean vue branch of the code. First define database tables for products, for stars, and for reviews; since you are going to write stars and reviews separately, it’s easier to use separate tables for stars and reviews, so you don’t have to worry about timing of writes. Please ensure that the tables have appropriate names; if you call “posts” the table for products, points will be taken off.

Once you have that, develop the grid page so you can insert products for testing.

After that, start work on the main page. First get it to display the list of products, with their average star rating (you can compute an arbitrary average for testing; just pick a random number on the server if you don’t have reviews yet).

Then, work at making the list of reviews expand/collapse. Create for each product a flag called _details, which is true for only one product at a time, and use it in v-if=”...” to decide which view of a product to show.

Finally, work at populating the list of reviews, and at implementing the behavior for the user to be able to save a review.

At the end come the finishing touches, like making sure the average is the real average, the checkmark for the Save button, etc.

Finally, test the behavior, using two users: one logged in normally, the other logged in via an incognito window.