

Placements.io Coding Challenge

Given a set of line-items, how would you go about writing a web-app that allows users to maintain an adjustable invoice?

Please see attached for example line-item data.

Below is a list of some expected behaviors and use cases. Pick two from bucket 1 and one from bucket 2:

Bucket 1:

- The user should be able to browse through the line-item data as either a list or table (ie. pagination or infinite-scrolling).
- The user should be able to edit line-item "adjustments".
- The user should be able to see each line-item's billable amount (sub-total = actuals + adjustments).
- The user should be able to see sub-totals grouped by campaign (line-items grouped by their parent campaign).
- The user should be able to see the invoice grand-total (sum of each line-item's billable amount).
- Multiple users should be able to edit the same invoice concurrently.
- The user should be able to sort the data.
- The user should be able to browse/filter/sort the invoice history, as well.
- The user should be able to output the invoice to *.CSV, *.XLS, etc.
- The user should be able to customize the layout.
- The user should be able to flag individual line-items as "reviewed" (meaning they are disabled from further editing).
- The user should be able to flag "campaigns" as being reviewed, as well.
- The user should be able to archive line-items
- The user should be able to add comments on an individual line-item.

Bucket 2:

- An integration into an external service that makes sense (eg. a currency conversion service, an export to Amazon S3, etc)
- The user should be able to filter the data (ie. by campaign name, etc., should affect the grand-total).
- The user should be able to share and reuse filters between users.
- The user should be able to see a history of all the adjustments/ comments/changes/etc. made to the invoice by different users.

Please respond with a link to your work checked-in under github/bitbucket along with a brief statement as to which parts you chose to implement and why they interested you, and/or your thought process on any design decisions vs. trade-offs that you had to make.

Your submission will be evaluated on the following criteria:

- We can easily build and run your web-app because you documented how to do so!
- The web-app satisfies one or more of the stated use-cases/expected-behaviors.
- If you think your skills lie on the frontend, go ahead and make the UI dazzle, if you are more comfortable on the backend, focus your efforts on best practices and functionality there - it's up to you.
- (Bonus Points) You wrote specs AND you documented how to run them against your work!