Placements.io Coding Challenge

Summary:

The challenge is to create an app that manages orders (or “campaigns”) and associated invoices for a digital media publisher. It’s almost like implementing a small slice of Placements.io. The primary user for this particular slice might be a finance person doing end of month billing in the platform, needing to make billing adjustments and generate invoices.

Object Model Background:

A publisher will sell ad campaigns to advertisers and use an order management system (OMS) to manage them. Within a campaign they will distribute budget across “line items” that represent different ads running in the campaign. Each line item would have its own contracted or “booked” amount - which all sum up to a campaign total. At the end of the month they may generate an invoice for a campaign that details the “actual” amounts to bill for each line item, based on ad performance (such as number of impressions or clicks). Any special billing “adjustments” are added to the actual amount to compute the final amount to bill on an invoice. Use the sample data to seed your application and guide your object model. Usually, multiple invoices may be generated per campaign but we’ve simplified the data and have the sample invoice data 1:1 with campaign/line item data. You may model it as you see fit - to suit the objectives you choose to implement.

List of features:

Below are 2 buckets of features. Bucket 1 represents the core use cases - please implement all of these. Bucket 2 is a set of more interesting features that can show off your skill set - pick at least 1 here.

Bucket 1:

- Implement the backend data model and seed it with the entire sample data
- The core objects (campaigns, line items, and potentially invoices) you model should have list views and detail views, with appropriate columns in the list views and appropriate fields and layouts in the detail views
- The only field that needs to support editing is the invoice “adjustments”

Bucket 2:

- A change history for editable objects/fields
- Ability to export invoices to *.CSV, *.XLS, etc.
- The ability to archive objects
- An integration into an external service that makes sense (eg. a currency conversion service, an export to Amazon S3, etc)

- A commenting system for appropriate objects in the app (eg a user might leave instructions on a campaign for another user by tagging them in a comment)
- A feature you think is interesting and can show off your skill set (document the feature in the readme)

Final Instructions and Evaluation

Please submit your assignment with a link to the Github repo (or equivalent) with a readme that outlines which parts you chose to implement, why they interested you, and your thought process on any design decisions or any trade-offs that you had to make. The readme should also include any detailed instructions on how to run your app (even better to host it somewhere).

Your submission will be evaluated on the following criteria:

- We easily we can build and run your app
- How well the app fulfills or exceeds the requirements above
- The level of difficulty in the objectives undertaken
- The quality of the output (code quality, lack of bugs, etc)

The exercise is meant to be a flexible and creative way to demonstrate your development skills. You may use any frameworks or libraries or languages that you wish. Many objectives can be easily knocked off with libraries/frameworks which is great and demonstrates a skill on its own. But also ensure there is enough of your own code somewhere in here to show off that part of your skillset. If you are more of a frontend dev, 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.