WageBase High-Fidelity Prototype README

How to use:

Go to https://wagebase.netlify.app/. That's it!

* Note that our app is not responsive for different devices/screens, and works best on a standard 13-inch laptop (e.g. Macbook Pro) with Google Chrome.

Tools Used

We built the prototype using ReactJS and used HTML/CSS for layout and styling. React helped us easily build interactive, stateful UIs, enabling us to iterate quickly. We used multiple React libraries, which helped us operate at a higher level of abstraction. We used React-Router to manage app routes. We used a React wrapper around Chart.js to display our charts in a responsive way, We used other react libraries to help us build autocomplete features for search inputs and also dropdowns. Some of these tools were hard to customize in terms of style which made it a little hard to perfectly conform to our Figma designs. Some of these libraries were also not being actively worked on which could pose issues in the long-term.

Wizard of Oz

On the Wizard of Oz side of things, the location for any rankings/wage comparisons is automatically assumed since there is no way to actually get a user's location in our hi-fi prototype. Secondly, when you submit a wage and engage in the verification procedure, we immediately simulate that it's been uploaded and verified to show the end modals. This verification procedure will actually occur 'behind the scenes' via a secure email verification program. Finally, we also simulate the social media sharing feature by simply linking you to the social media website. The actual creation of a post with our data embedded in it was beyond the scope of our prototype stage.

Hard-Coded Data

We hardcoded a mock dataset in the front-end of our hi-fi web prototype since we didn't have an easy way to collect data on restaurant wages - this includes, restaurant names, locations, wage data, etc. We make use of this mock data in all parts of our website, from the restaurant rankings to individual restaurant pages and graphical visualizations/comparisons.

Limitations

- Our Demo App is deployed on a free dyno on Netlify. While it fully works, the response times may sometimes appear to be sluggish (especially for images) since the free dyno isn't particularly fast. If an image doesn't load, refreshing the page will usually fix it.
- We only have a small number of fake restaurants
- Wage data uploaded by service workers isn't actually added to our "database" since there is no database at this stage
- Location search for restaurants is mocked. We don't actually have a search index - search results are always the same no matter the query. For the ideal/representative use case, please use 'pizza' as your search term.
- In the Compare Wages task-flow, you can only add restaurants for comparison if the restaurant name and location matches the mock data in the database. An easy way to do this is to select the same order of items from the dropdowns in both fields. For instance, if you select 'Pizza Palace', the third item in the Restaurant Name dropdown, then you must select the third item in the Location Dropdown, 'Mountain View', for a successful addition. This is once again a constraint that comes from the lack of actual data behind our platform.
- Some of the extra violations from our heuristic evaluation could not be addressed in the hi-fi due to time constraints