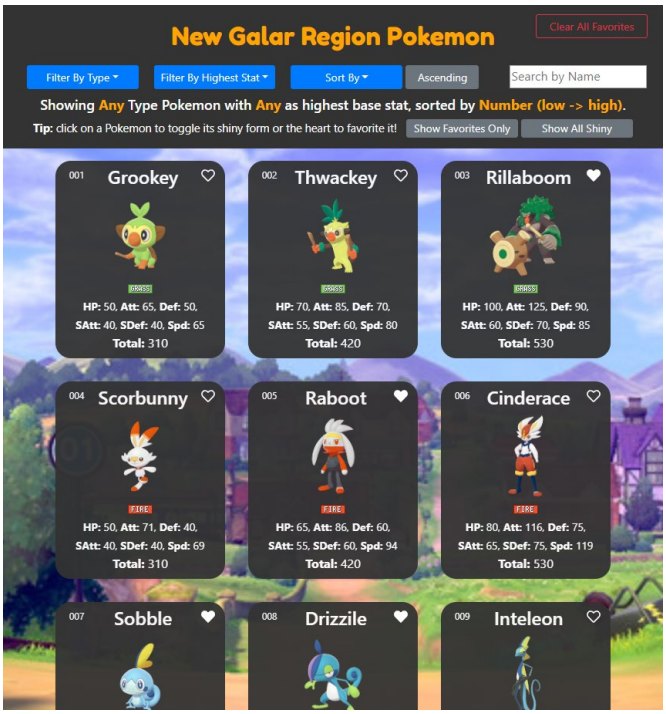
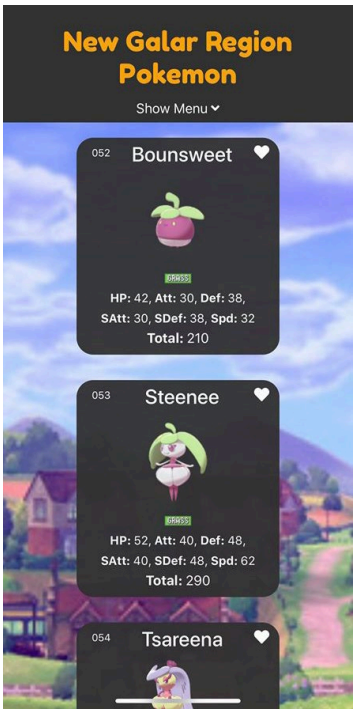
Leon Lei's CSCI 1300 Capstone

For my capstone for the User Interfaces/User Experience course, I extended the *Development* assignment to have substantial design and development components. The application is implemented with React and displays all 400 Pokémon from the Galar region (introduced in the recent Pokémon Sword and Shield video games). The application was ultimately deployed using GitHub Pages on my personal domain.

Link: https://leonlei.com/galarpokemon/





How was the **design** of this project extended?

- With many filtering options available and a substantial number of items for the user to view, careful thought was put into the design and user experience of the app. Several sketches and simple hand-drawn wireframes were used to brainstorm different layouts of the app and prioritize certain aspects of the user experience. Sketches are provided at end of document.

- Attention was paid to the differing desktop and mobile experiences of the app. One main concern was how to display the filters and sorting nav bar to the user in a mobile view given how much space they take up. The ultimate solution

used was to allow the user to collapsing the menu bar to save on vertical space when they have already applied their filters and just want to see the list of Pokémon. Some other examples of details that were paid attention to are where to break lines for different page sizes and whether the ability to scroll the main content of the page while hovering over the menu bar at the top should be disabled.

- Design was improved over an iterative process which involved soliciting feedback from other users. The app was deployed early on by the time of Development assignment deadline so that others could easily use it and provide feedback. Two separate deployments were used to allow for some hand-run A/B testing capabilities. One of these deployments (an older version) can be found on Heroku: https://galar-pokemon.herokuapp.com/. The current version is deployed on GitHub Pages.

- Different designs and layouts of the Pokémon list-item cards were considered as well as how much information would be presented to the user. One example of this was whether it was necessary to display a total base stat number to the user or whether the stats that made up that number were enough. I asked 10 users of the app whether they wanted this additional information, and all of them said yes. However, to not make the cards significantly more overloaded with information, I modified the text arrangement and the shape of the card to make it more compact and take up nearly the same amount of space as it did before.

- The resulting app is polished overall, and its design draws from many of the principles that we learned in the course.

How was the **development** of this project extended?

- Went beyond the 12-item requirement by using 400 items. This involved a great deal of data collection through web scraping and other means. Over 800 images were downloaded for this project since a normal and shiny form of the image is provided for each Pokémon. Within each item, we also dynamically load images rather than text for the types of the Pokémon to improve the overall feel of the app. Each Pokémon can have anywhere between 1-2 types.

- Added greater filtering and sorting capabilities to make the app more powerful and useful for users. For instance, the “filter by type” dropdown

dynamically maps all the available types from the data into options, and the same is done for "filter by highest stat". Furthermore, the "sort by" feature was redesigned to pull out the "ascending/descending" option, allowing the sort-by list itself to support sorting by each of the available Pokémon stats as well as their Pokédex number. These filters and sorts all behave consistently with other functionalities in the menubar, the result of extensive debugging and testing.

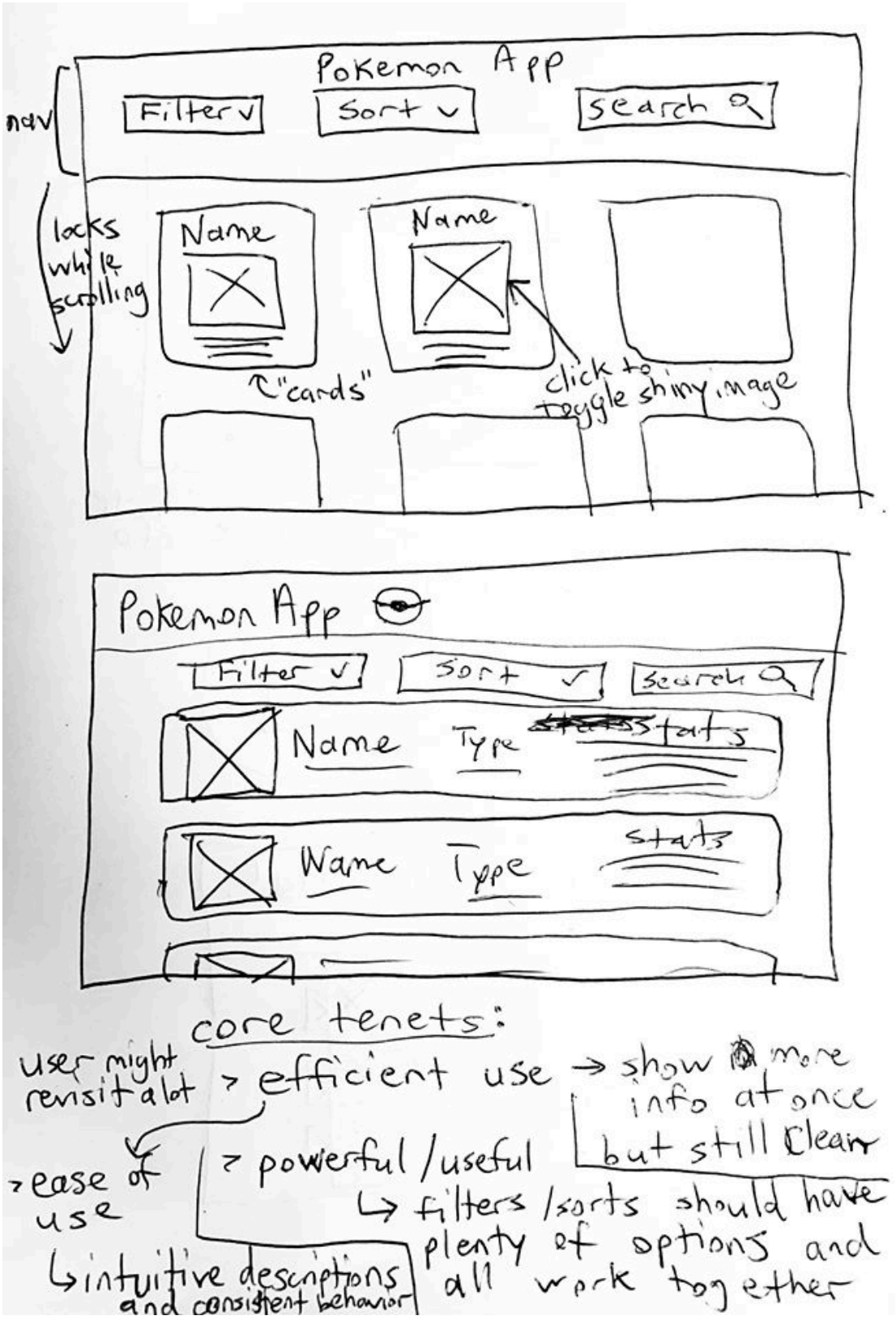
- Implemented a **favoriting** feature using localStorage so that it has some level of permanence. Extensive testing was done to ensure that the coloring behavior of the heart is consistent even when filters and sorts are applied, or if the page is refreshed. Furthermore, an ability to show only the favorited items was implemented to work alongside all other filter/sort/search functionality.

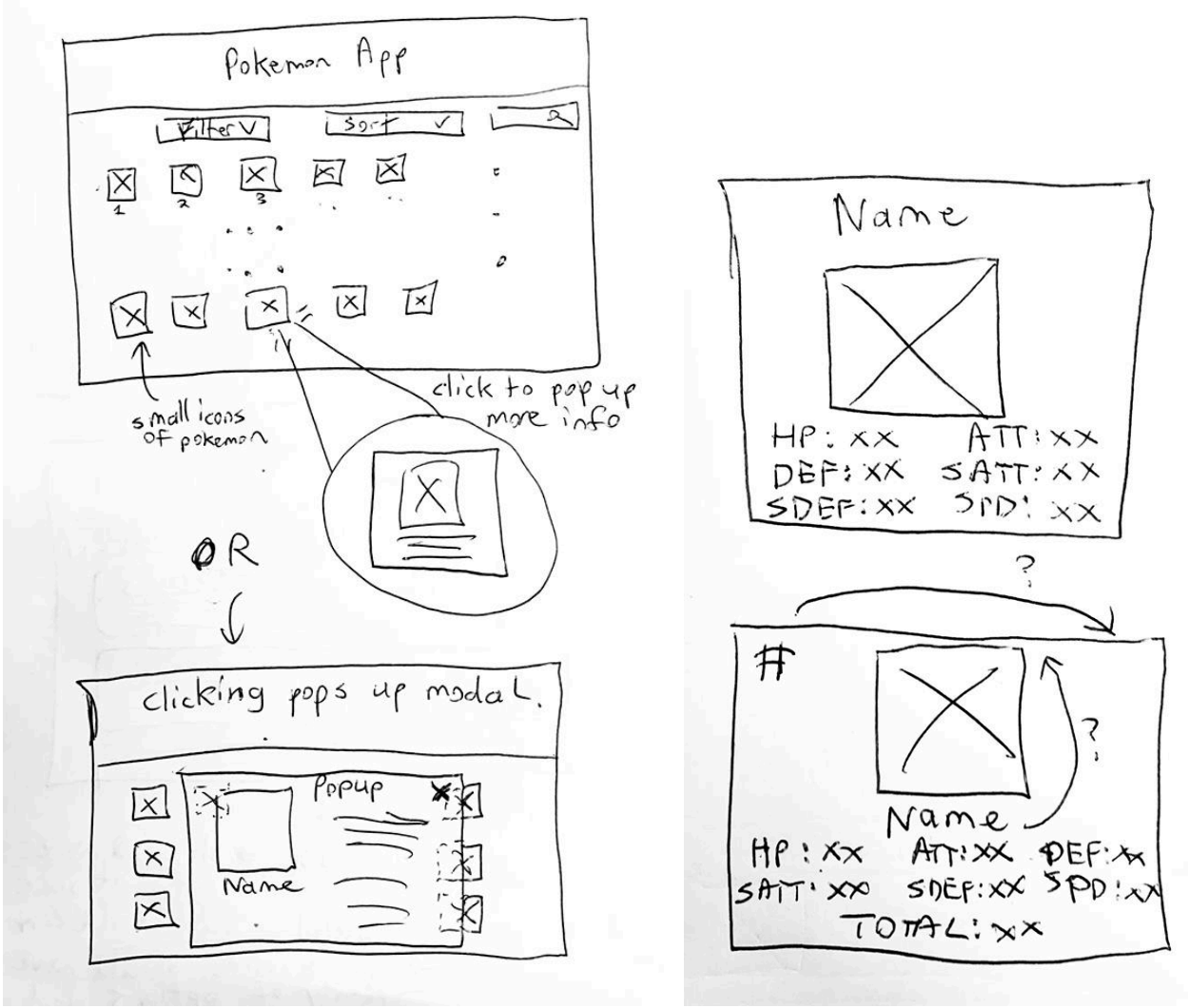
- Other nice options for the user were provided, such as a button to toggle all the shiny forms, and a button to clear all the favorites from localStorage. Implemented logic for many edge-cases, e.g. appropriately updating the HTML elements when the corresponding React component's state changes, or displaying a message in the main body if the user's applied filters do not return any matching Pokémon.

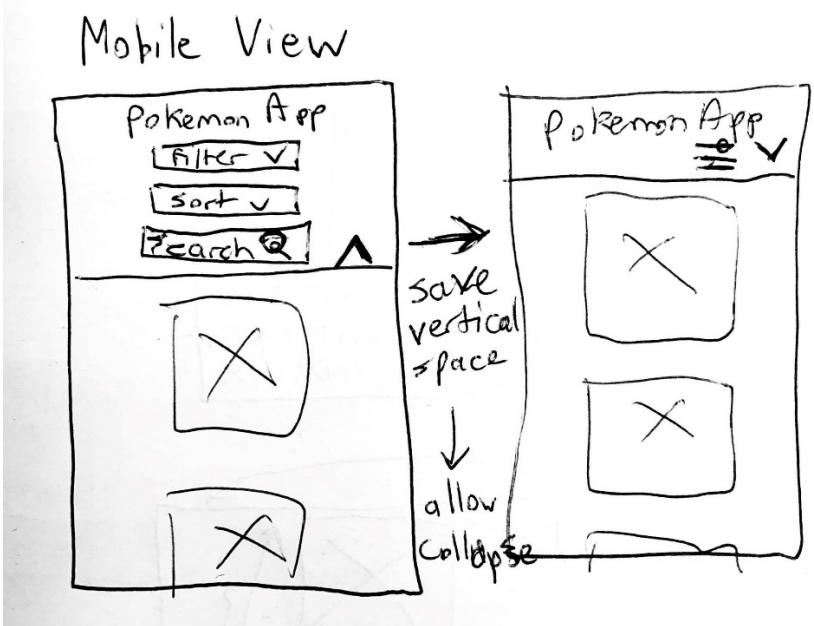
- Careful attention to the styling/design and user experience of the app through JavaScript and CSS, as well as a few dynamic changes made directly in React. Tried to make the app look as nice and compact as possible, as well as work smoothly. This includes the styling for the Pokémon cards in the main content of the page as well as the filter/sort buttons in the top menu bar. Customizations are also consistent and save in localStorage, i.e. when you click the header at the top of the page to change the color of some of the menu text. This is the same for toggled shiny forms, so if you prefer to look at a Shiny version of a Pokémon it will remember it for when you next open the app. Implementation-wise, this involved the tracking of many state changes and appropriate handling of corresponding logic.

- Great efforts expended on the responsiveness of the app. Implemented many different styling rules with media queries. Tested with several browsers and different screen widths/devices. While there are still some minor kinks and more extensive testing can be done, the app is highly responsive overall.

Sketches


Pokemon App
nav
Filter
Sort
search
locks while scrolling
Name
Name
"cards"
click to toggle shiny image
Pokemon App
Filter
Sort
search
Name
Type
stats
Name
Type
stats
core tenets:
user might revisit a lot
> efficient use
> ease of use
> powerful / useful
↳ intuitive descriptions and consistent behavior
→ show more info at once but still clean
↳ filters / sorts should have plenty of options and all work together


Pokemon App
Filter
Sort
small icons of pokemon
click to pop up more info
OR
clicking pops up modal
Popup
Name
Name
HP: XX
ATT: XX
DEF: XX
SATT: XX
SDEF: XX
SPD: XX
#
Name
HP: XX
ATT: XX
DEF: XX
SATT: XX
SDEF: XX
SPD: XX
TOTAL: XX


Mobile View
Pokemon App
Filter
Sort
Search
save vertical space
allow collapse
Pokemon App