

Calendar widget

General appearance

Behavior

- Aesthetic designed to look like example

Omissions

- Not responsive - all positions/sizes set with pixels

Task 1: Add and Remove Events (without metadata)

Behavior

- Click and drag to create event (uses react-resizable)
- Snaps to half-hour blocks
- Each meeting is given a unique, consecutive id (e.g., if Meeting 1 is created then deleted, the next meeting would be Meeting 2)

Omissions

- Cannot resize in the upward direction
- No bounds on resizing downward (i.e., breaks if dragged off calendar)
- Resizing doesn't work perfectly - sometimes mouse gets ahead/behind of meeting

Task 2: Persist Events between reloads

Behavior

- Calendar should look the same when page is reloaded (uses redux-localstorage)
- Meeting ids should continue to be incremented correctly

Omissions

- State not stored as `immutable` types so that it can be more easily saved in local storage

Task 3: Allow the user to provide and edit event titles

Behavior

- After creating initial meeting, default name is "New meeting"
- Goes into edit mode and selects name after initial meeting is created
- Pressing enter while name is focused or clicking the check mark saves the name and disables the input
- When not in edit mode, an edit button shows; clicking it enters edit mode and selects the name (and changes the edit button to a check mark)
- The textarea expands as the user writes

Omissions

- Styling not handled when height of meeting textarea is greater than height of meeting

Task 4: Allow users to drag events with the mouse

Behavior

- Dragging meetings snaps them to the nearest half hour (uses react-draggable)
- Dragged elements always have highest z-index

Omissions

- Resizing events after creation is not possible
- No bounds on dragging (i.e., breaks if dragged off the calendar)

Task 4: Handle event overlap when multiple events occupy the same time span

Behavior

- Meetings line up next to each other when they overlap
- When being dragged, meetings expand to full width and resume normal positioning, then immediately adapt when dropped

- Order of meetings based on id

Omissions

- Sizing and positioning do not work correctly when one meeting overlaps with two meetings that do not overlap with each other
- Meeting names can be very cramped when meetings get narrow enough

This project was bootstrapped with [Create React App](#).