

Web app setup and stacks

🕒 Created	@December 19, 2024 9:53 AM
🏷️ Tags	

Frontend (repo)

1. Project Overview

- **Purpose of the Project:**
 - This is a web application for data visualization, allowing retailers to find any outstanding items or discrepancies in their supply chain
 - **Key Technologies:**
 - Mui components
 - ChartJS
-

2. Setup and Environment

- **How to Get Started:**
 - See [Readme](#) for details
 - **Development Commands:**
 - `npm run dev` to run local
 - `npm run build` to test if it builds successfully
 - `pnpm install --no-frozen-lockfile` to install dependencies
-

3. Navigation Implementation

- **Overview:**
 - `Navigation`: Drawer for main pages

- `NavigationDrawerContextProvider` : provide context for whether the main nav drawer is open; used to determine the margin of secondary drawer (filter drawer)
 - `LoginButton` : Login button for navigation; show logout on hover
-

4. MUI Components and Styling

The MUI theme configuration customizes various aspects of the Material-UI library to align with the project's design system. It includes definitions for typography, palette, spacing, breakpoints, and component-specific style overrides. This ensures a consistent look and feel across the application while allowing easy customization.

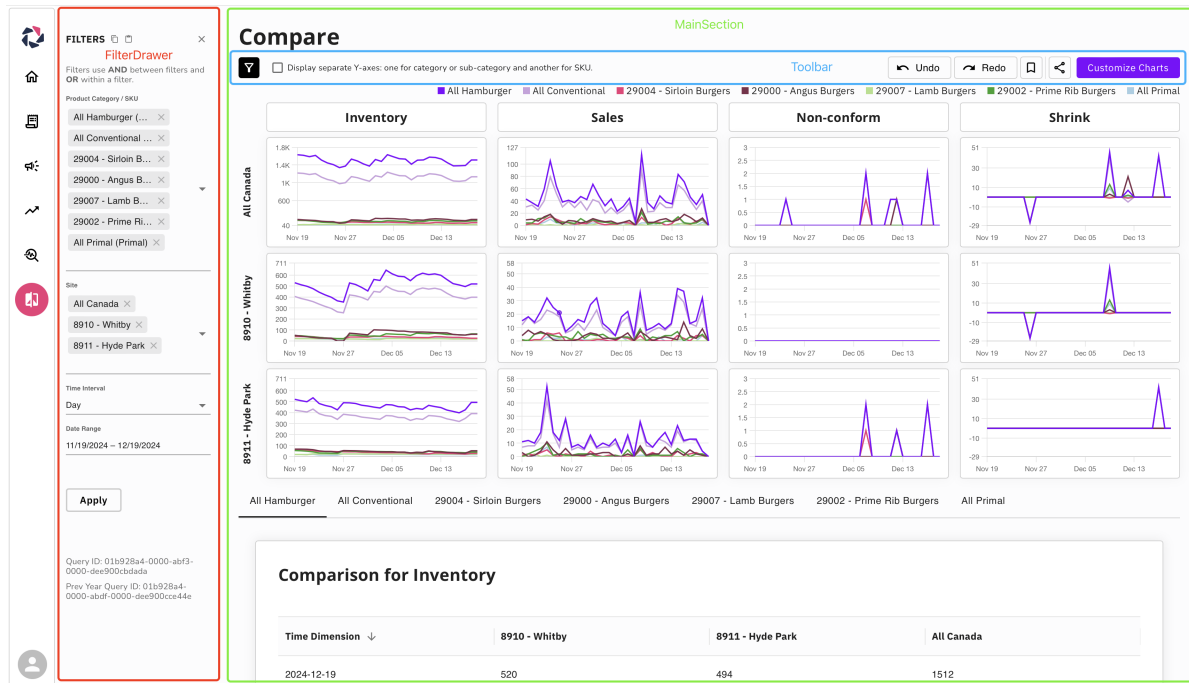
- **Customizations:**

- Typography variant, spacing, border radius, colors (background, text, icon, chart), breakpoints; check `theme.ts` for specific values
 - Globally override styles for `MuiButton`, `MuiAutocomplete`, `MuiTextField`, `MuiGrid`, `MuiContainer` to ensure consistency throughout
 - Override fields for MUI theme in `'@/types/mui.d.ts'`
-

5. Globally Shared Components

- **General Layout**

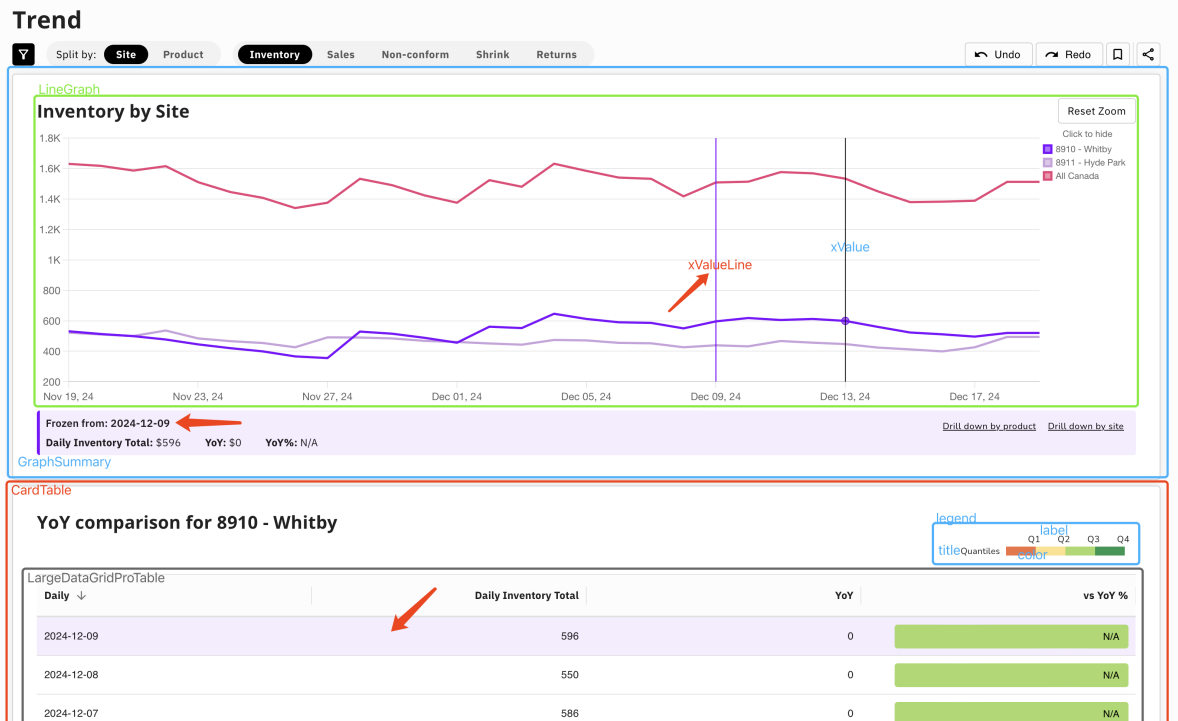
- `FilterDrawer` : Wrapping all the filter selectors
- `MainSection` : Space for main page (use with `FilterDrawer`)
- `FilterSelectionContextProvider` : Remember what user selected for each individual page in this session, should update values when user apply the filters
- `'@/components/filter'` : All the filter selectors
- `'@/components/toolbar'` : collapse on small screen size toolbar



General layout with the above components

• Graphs and Tables:

- **GraphContextProvider**: Wrap this around LineGraph and Table
 - **xValue**: sharing reference line across multiple graphs
 - **xValueLine**: line fixed in graph / selected table row
- **GraphSummary**: Wrap this around graph to show value stored in **xValueLine.data**
- **LineGraph**: see [Chart Section](#)
- **LargeDataGridProTable**: provide style for full size table (see also [Table Section](#))
- **CardTable**: Wrapper for tables for title and legend style
- **Legend**: create a coloured legend for with label and colours provided



GraphContextProvider wrap around all table and graphs; red arrow points to content using xValueLine

6. Data Grid Implementation

- Overview:**

- Provide a more detailed version of data used in graph
- Display report data and allow functions like export, filter and search

- Key Configurations:**

- See [Mui Doc](#)

- Custom Features:**

- Customized column styles in `'@/components/tableColumnCustomizations'`

- Common Issues:**

- `useGridApiRef` : used for scrolling table row into view
 - can't be used in a loop; if you have multiple tables, can create one and pass it into different table as props

7. Chart Implementation

- **Charting Library Overview:**

- ChartJS

- **Key Charts Built:**

- `LineGraph`
 - `data` :
 - Graph each `LineGraphData` as a line on the chart
 - `LineGraphData` : put all x-values in an array (`TimeDimension`) and all y-values in another (`metric`)
 - `size` : ('full', 'small', 'medium', 'large')
 - different number of ticks for each; include year in y-axis for full size
 - uses custom tooltip for `'full'` ; default tooltip for other sizes
 - `yAxisScales` : max & min for `y` and `y1` ; will adjust scale used depending on value of `enableDualY`
 - Tooltip: (see `kpiGraphTooltip.tsx` for sample usage)
 - `CustomTooltip` : customized tooltip component
 - `registerTooltipEvents` : function that register events (i.e. button click) for tooltip; check `RegisterTooltipEventsProps` for details
 - `tooltipData` : any additional data you want to show or use for the tooltip (i.e. siteOptions, searchParams); this gets passed into `registerTooltipEvents`
- **Challenges and Notes:**
 - ChartJS graph sometime does not respond to parent resize (try setting parent width to 99%)

8. Code Structure and Patterns

- **Folder Structure:**

- All the authorized pages are in `'@/app/(authorized)'` ; the `layout.tsx` here check for authorization
 - Put reusable components in `'@/components'` ; and any further customized component for individual pages in their own `./components`
 - Keep reusable / common function and api calls in `'@/lib/services'`
 - **Best Practices:**
 - Create components where possible to maintain good structure and readability
 - Use variables defined in `theme.ts` for easier maintenance
-

9. Known Issues and Future Work

- **Future Improvements:**
 - See TODO comments in repo
 - Might want to work on the `navAndShareToolBarV2` to make it collapse on small screen size
-

10. Documentation Links

- **MUI Documentation:**
 - [Data Grid](#)
 - [MUI Theme](#)
 - [MUI sample dashboard](#)
 - **Chart Library Docs:**
 - [Chart.js](#)
-

API ([repo](#))

1. Project Overview

- **Key Libs:**

- Pandas
- fastapi
- pydantic (validate query params)

2. Setup and Environment

- **How to Get Started:**

- See [Readme](#) for details

3. Code Structure and Patterns

- **Best Practices:**

- Be sure to check for valid parameters in `model.py`; this will ensure we produce valid sql queries 😊
-