Challenge Objectives

Build the following pages using ReactJS v17:

- Landing
- Search Deals
- Employee Lookup

Project Background

- The Eclipse Suite provides the capability to create and approve special pricing discount requests and manage the associated workflows and systemic distribution of approved prices. The suite consists of several modules for different purposes.
- When there is a deal that HP would like to win due to for instance competition on the market, their customers can request for special pricing of products. In this case HP's sales support employees will enter the deal details into the Eclipse application. The details will be passed through different team members, who will analyze the price requests if they can authorize the deal or not. If the deal gets authorized a quote will be created for the customer.

Technology Stack

- ReactJS v17 (latest stable version)
- Material UI Framework (REQUIRED)
- HTML5
- Typescript
- CSS

Challenge Requirements

For this challenge, we are going to build the UI prototype that covers the mentioned pages, please refresh this page after registering to the challenge to see the page specific details.

Please check the following sections for general challenge requirements which also need to be followed.

General Requirements

- Must use functional components only (No class based components)
- You'll be working with data from local json files, you must NOT hardcode any data in the source code
- Minimum supported width should be 1440px
- Must use ReactJS v17 and Material UI Framework
- Must properly implement the UI with grid system (that follows the design)
- The main content should have fluid width
- Must follow ReactJS best coding practices
- No lint errors and no errors with production build
- You are expected to create a detailed README file including information on how to setup, run and verify your application
- Show loading spinners when populating data from API / local JSON to UI
- Performance objectives on UI in general:
  - Landing page load < 250 ms
  - Search Details page load: < 500 ms
  - Employee Lookup page load: < 300 ms

HTML Requirements

- HTML should be valid HTML5 compliant.
- Provide comments on the page elements to give a clear explanation of the code usage. The goal is to help future developers understand the code.
- Please use clean INDENTATION for all HTML code so future developers can follow the code.
- All HTML code naming should not have any conflicts or errors.
- Element and Attribute names should be in lowercase and use a "-" or camel naming to separate multiple-word classes (i.e.. "main-content", or "mainContent)
- Use semantically correct tags- use H tags for headers, etc. Use strong and em tags instead of bold and italic tags.
- No inline CSS styles- all styles must be placed in an external stylesheet.
- Validate your code- reviewers may accept minor validation errors, but please comment your reason for any validation errors. Use the validators listed in the scorecard.

Code Requirements

- Provide comments on the CSS code. We need CSS comments to give a clear explanation of the code usage. The goal is to help future developers understand the code.
- Please use clean INDENTATION for all CSS so developers can follow the code.
- Do not create a single .css/.scss file for the whole app. Each component should have its own stylesheet file.
- Ensure that there are no lint errors.
- All CSS naming should not have any conflicts.
- You’re free to choose between CSS & SCSS but you need to use flex instead of float.
- Follow a clean folder structure.
- Use CSS3 Media Queries to load different styles for each page and don't build different page for different device/layout.
- Use CSS to space out objects, not clear/transparent images (GIFs or PNGs) and use proper structural CSS to lay out your page.
- Make sure npm run build works without any errors.

Javascript / TypeScript Requirements

- All JavaScript / TypeScript must not have a copyright by a third party. You are encouraged to use your own scripts, or scripts that are free, publicly available and do not have copyright statements or author recognition requirements anywhere in the code
- Use typescript and linter for code quality

Licenses & Attribution

- Third-party assets used to build your item must be properly licensed and free for commercial use.
- MIT and Apache v2 licenses are ok, for any others please check with us to get approval first.
- Sufficient information regarding third-party assets must be present in your documentation. This includes the author, license info and a direct link to the asset online.

Browser Requirements

- Windows: Chrome & Edge Latest
- Mac: Chrome Latest & Edge Latest

Registered User Additional Information

Page Details

Table Component

We'll provide a reusable table component that will be used on the Deal Search and Employee Lookup pages. The table component supports many features and you should properly configure it to cover the requirements for different pages.

Please keep the code for the table component in its own folder and use it like a library, do NOT modify the code of the component unless there’s a bug in it.

Landing Page

The landing page was already done, with a mock GraphQL backend (both will be provided in the challenge forum), you should use these as the base for your work. However for this challenge you don't have to create new GraphQL mock backend, you can just use the provided json files in the UI code to simulate api responses. You might need to expand the data to have more records so you can test pagination on Deal Search / Employee Lookup pages.

On successful login you need to cache the json response since you’ll need to use the data from the login response to do some validation on other pages.

Deal Search

- When the user clicks on Open Deal from the landing page, the user will be redirected to the Deal Search page
- The Deal Search page is a full page, NOT a popup page
- The Deal Search page will open in blank state, with the CANCEL button enabled, clicking CANCEL will navigate the user back to the Landing Page.
- Clicking the close icon will also navigate the user back to the Landing Page.
- The user can Click SHOW OPTIONS to expand the search panel, and then click HIDE OPTIONS to hide collapse the panel
- The various search option values will be loaded from json (they will come from the GraphQL backend later)

- The SEARCH button is initially disabled, and it will be enabled if any of the following happens:
    - Text entered into search input field
    - At least one country added
    - At least one Employee added
    - For the country search input, a maximum of 2 rows or 8 countries can be added, whichever comes first
    - Recent deals (last 50) is selected from deal filters
    - Version 0 deals is selected from deal filters
    - Pending deals is selected from deal filters
    - Won deals (last 365 days) is selected from deal filters
    - Active deals (last 365 days) is selected from deal filters
    - Deals with unsubmitted changes is selected from deal filters
- Based on the “Search by” value, the search input boxes will change:
    - If the option is Ext. quote no.: this will have two input boxes as shown in design
    - For any other options we'll have only one input box
    - The default option when the user opens the page should be All
- Clicking ADD under the Employee Names section will open the Employee Lookup page (explained below)
- The user can select multiple Employee Names and then REMOVE them (multi-select and bulk delete)
- Clicking SEARCH will trigger a search API call to the backend, in this case you'll be calling a local method to get the JSON response and populate the table with data
    - For the deal search result, you need to do the pagination on the client side
- The table will show the following columns in order (columns not mentioned should not be part of the table and shouldn’t be available in the column chooser either, but they may be used for future API calls):
    - Deal# (dealNumber)
    - Version# (dealVersionNumber)
    - Deal Country (dealCountry)
    - Freeze FL (freezeFlag)
    - Lock Flag (dealLockFlag)
        - Note for this column there needs to be an icon indicating the lock status
    - Stop Reason (stopReason)
    - End User Company (endUserCompanyName)
    - Deal Description (dealDescription)
    - Begin Date (dealBeginDate)
    - End Date (dealEndDate)
    - Version Status (versionStatus)
        - Note this column shall show a mapping text based on the value:
          -----Mappings are------

UNN = Open
FNN = Authorized
FSN = Approved
FSP = Pending Validation
FSR = Invalid
FSQ = Partially Quoted
FSA = Quoted
UDN = Pricing Denied

- Prog Code (dealProgCode)
    - Note this column needs to be shown as “icon mapping_name_text” something like “X Not Authorised” where X will be an icon.
      Cross_icon = cross icon like X in Red
      Right_Tick = Right tick mark in green
      -----Mappings are------
      F = “**Right_Tick** Fully Authorized”
      U = “**Cross_icon** Not Authorized”

- BDME Code (bdmeCode)
    - Note this column needs to be shown as “icon mapping name text” something like “X Not Authorised” where X will be an icon.
      Cross_icon = cross icon like X in Red
      Right_Tick = Right tick mark in green
      -----Mappings are------
      N = “**Cross_icon** Not Approved”
      A = “**Right_Tick** Approved”
      S = “**Right_Tick** Approved for Distribution”
      D = “**Cross_icon** Pricing Denied”

- Dist Code (quoteDistCode)
    - Note this column shall show a mapping text based on the value:
      -----Mappings are------
      N = Quote Not Sent
      Q = Quote Sent
      A = All Quotes Sent
      P = Quote Pending (SAP)
      R = Quote Rejected (SAP)

- W/L Stat (wonOrLossStatus)
- BD ID (bdId)
- Lock User (dealLockUserId)
- BusModel Name (businessModelName)

    - End User Country (endUserCountryName)
    - End User St/Prov (endUserState)
    - Driving Sales Rep (drivingSalesRep)
    - Submitted Date (submitDate)
    - Last Saved Date (lastSavedDate)
    - Deal Creator (dealCreator)
    - Opp ID (cRMOpportunityId)
    - Ext Quote Nr (dealExtQuoteNumber)
    - Ext Quote Ver (dealExtQuoteVersion)
    - Tier Level (tierLevel)
    - Tier Name (tierName)
    - Program ID (sFProgramId)
    - OPG # (oPGNumber)
    - Sales Territory ID (salesTerritoryId)
    - Deal Source (dealSource)
    - Deal Type (dealType)
    - MC Code (mCCode)
    - Tenant ID (tenantId) - hidden by default, can be re-enabled in column chooser
    - Last Message (lastMessage) - hidden by default, can be re-enabled in column chooser
    - RequestID (requestId) - hidden by default, can be re-enabled in column chooser
    - Watson Quote ID (watsonQuoteId - hidden by default, can be re-enabled in column chooser
- The table shall be implemented using the common table component, and the following features should be working:
    - Sorting / Clear Sorting
    - Reordering columns
    - Choose columns
    - Fit columns
    - Filter
    - Searching
    - Resetting table
    - Export
    - For records with isExpired=true, highlight the row with red color as shown in design
    - Hover on row will show the row in light blue color as shown in design
    - Selected row will show in dark blue as shown in design
    - Note that this table is readonly, and most of the above features are accessible through context menus, and they are all supported by the table component already
    - Please follow the provided design to apply styling to the table component

- When a row is selected the OK button will be enabled, but clicking it will do nothing for now
- When a row is selected, the user can double click and the deal will be opened, since we are not implementing the deal open feature now, please implement an empty event handler that writes a text to the log (with row data)
- Please implement the following UI validation rules:
    - If Search button is disabled and user hovers on the button, show a tooltip that says "Please enter some search criteria or filters"
    - If Search By Deal Number, the max input length is 10, and it needs to be numeric only
    - ExtQuoteNr needs to be AlphaNumeric, and ExtQuote Version needs to be numeric only
    - Business model is AlphaNumeric and max length is 20
    - MC Code is alphanumeric and must be exactly 3 chars
    - Partner pro ID is AlphaNumeric and can allow the special character hyphen(-), max length is 16, sample value: 3-2SS-4543
    - If Search Result is not null and search result rows count is equal to defined max number of rows (default to 200 but make it configurable) and deal filter contains (Current version of deal or all version of deal or Version 0 deals), show the following message in an alert:
        - More than max record(s) found for the given search criteria. The deal and version you are searching for may not appear in the returned results. Please refine your search criteria.
- Please implement the following UI controls
    - If Deal filter contains (Version 0 deals or Deals W/UnSubmitted Changes or Pending Deals or Recent Deals Last 50)
        - Disable Search By panel, Check box Deal date and set Date range visible False.
        - Enable Dealtype label, Deal type drop down
    - If Deal filter contains (“Current version of deals” or “All version of deals”),
        - Disable Dealtype label, Deal type drop down
    - If Deal filter contains (Won Deals (Last 365 Days) or Active Deals (Last 365 days))
        - All the filters shall be active
- Perform the below operation for Tenant drop down based upon "**tenantId**" , "**tenantIdDealOverrideFlag**","**federalAccessFlag** values from Authorization response:
    - If tenantId is “HPF or HPI” and tenantIdDealOverrideFlag = Y and has federalAccessFlag = Y
        - set "ALL" as default and remove "HPI,HPF" from Tenant drop down.
    - If tenantId = HPI and tenantIdDealOverrideFlag = N and federalAccessFlag = Y
        - set "HPI,HPF" as default and remove ALL from Tenant drop down.
    - If tenantId = HPF and tenantIdDealOverrideFlag = Y,

        - set "HPI,HPF" as default and remove ALL from Tenant drop down
    - ELSE: default case, just show deals matching user's tenantID in Tenant Drop down and it shall be read only.

Employee Lookup

- The Employee Lookup page will open as a popup to the current page
- The default view will display the following message “Enter at least 3 characters to search by. Search by last name first. You can do a match of both first and last name; to do this, enter last name than the first name, separated by space.”
- The search input field will have the following placeholder “Search by last name first”
- Once user enters something (non empty text & minimum 3 characters) the SEARCH button will be enabled
- Clicking SEARCH will trigger an API call to the backend, in this case you'll be calling a local method to get the JSON response and populate the table with data
- If SEARCH action doesn’t return any results, a temporary error will be displayed stating “No results found”
- All columns shown in the design need to be shown in the table in the same order by default. You might see extra details in response like “employeeNumber”, pls ignore such data for UI display, however cache it for future uses.
- The table shall be implemented using the common table component, and the following features should be working:
    - Sorting / Clear Sorting
    - Reordering columns
    - Choose columns
    - Fit columns
    - Filter
    - Searching
    - Resetting table
    - Export
    - Hover on row will show the row in light blue color as shown in design
    - Selected row will show in dark blue as shown in design
    - For this table pagination is done on the server side (as you can see from the response json)
    - Note that this table is readonly, and most of the above features are accessible through context menus, and they are all supported by the table component already
    - Please follow the provided design to apply styling to the table component
- Once the user selects a row the OK button will be enabled, clicking OK will close the popup and add the selected user to the Employee Names list of the Deal Search page

- Once the user selects a row, double clicking on it, will close the popup and add the selected user to the Employee Names list of the Deal Search page