

Approach for Quote Lifecycle -

View Quote -

- To view the quote record on the UI, I created a wrapper class and getQuotebyRecId method in quoteDTO class.
- getQuotebyRecId method is used to fetch the current Quote__c record by passing the recordId to it and wrapper is used to bind the data provided by SOQL and return it to JS controller for displaying it on UI.
- Designed lightning-layouts for displaying the startDate and endDate in the grid pattern.
- The wire method was used to retrieve the apex data.

Edit Quote -

- For editing the startDate and endDate fields, I had designed the lightning-layouts with two lightning-inputs in HTML for displaying in the grid format.
- For both the lightning-input, the handleChange method is used in JS to store the changed startDate and endDate in local variables.

Save Quote-

- A lightning-button was created inside the lightning-layout to save the data provided by lightning-inputs.
- handleSave method is used to save the data of lightning-inputs ie of startDate and endDate to database.
- Upon changing the quoteData variable with new startDate and endDate values, updateQuote method from apex is called and the whole quoteData variable is passed into it.
- updateQuote method takes the wrapper data as input and creates a local Quote__c variable to map the wrapper data to the Object variable for update.
- Upon successful update of the Quote record, a Success toast event is fired and the apex data which we queried from database is then refreshed by using refreshApex method so that UI data also refreshes and if any error occurs while updating the record then the Error toast event is fired.

Approach for Launch Dialog box -

- For creating a dialog box or modal which will be fired from editQuotePage (given in assessment) LWC component a custom event has been fired upon clicking on the "Adjust Quote" button from the quoteTotalSummary LWC component to the parent component editQuotePage stating to open the modal.
- As per winter'23 release of LWC, we have new attributes of lightning-modal-header, lightning-modal-body and lightning-modal-footer which can be used in component after extending the JS class with LightningModal. Hence I thought to use this, as it is a new enhancement and can be used more efficiently instead of the old method of creating sections / divs along with the lightning design system classes.

- Therefore, quoteModal LWC component was created as per the wireframe provided so that it can be used to invoke from parent (editQuotePage) component.
- Now, as soon as the event is received from the child component (quoteTotalSummary) to open modal, then the boolean value is passed to the asynchronous function openAdjustQuoteModal and in this function the child component (quoteModal) is being invoked by using open method and recordData (which is wired data) is passed as attribute to get it displayed in modal.
- Once the modal is invoked, the recordData variable is passed to the adjustQuotePrice child component to display the information.
- After entering the adjusted amount, the handleSave method is called from the Save button on quoteModal component and adjustQuotePrice component's saveData method is invoked to save the data to the database.
- Upon successful update, the closeModal event is fired to parent the component.
- At last, after the event is received in the editQuotePage component, the refreshApex method is called to refresh the wired data.