Service Cloud Implementation

Use Case Introduction

This Use Case has been developed for ABC Company's new system for managing customer support and self-service using Salesforce Experience Cloud. Based on the design team's gathering of business and functional requirements, the new Experience Cloud implementation will replace the manual customer support processes currently used by ABC Company. The new system aims to reduce response times, improve customer satisfaction, and leverage the existing Salesforce platform for seamless integration with the company's CRM and other tools.
The solution will utilize Salesforce's customizable customer theme layouts to provide users with an intuitive interface for accessing knowledge articles, submitting cases, and tracking support requests. In addition, the system will integrate a chatbot to offer real-time assistance to customers, answering common queries and guiding users to the right resources. By empowering customers with self-service options, relevant support resources, and chatbot assistance, ABC Company is better positioned to enhance the overall customer experience and streamline internal support operations.

Explanation Of Use Case Contents

Name of the Use Case: Experience Cloud implementation for ABC Company's customer support and knowledge management.

Description: The implementation provided ABC Company's support team with the tools to engage customers at the right time, offering quick access to knowledge articles, real-time chatbot support, and efficient case management. This improved both customer experience and internal processes.

Deliverables:

- Customizable customer theme layout for a branded support experience.
- Self-service portal with access to knowledge articles.
- Streamlined case management for efficient customer support.

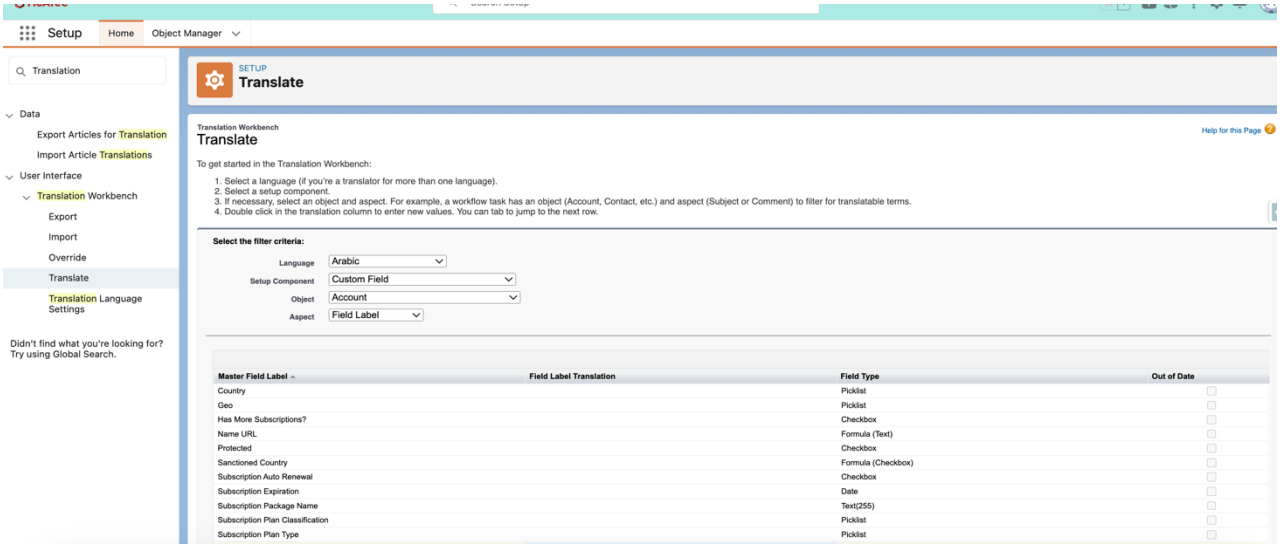
- Real-time chatbot for instant customer assistance.
- Enhanced customer engagement and satisfaction.

For Automated Process entities used: Cases,Knowledge Articles, Trigger, Flows, Batch Class, LWC etc.

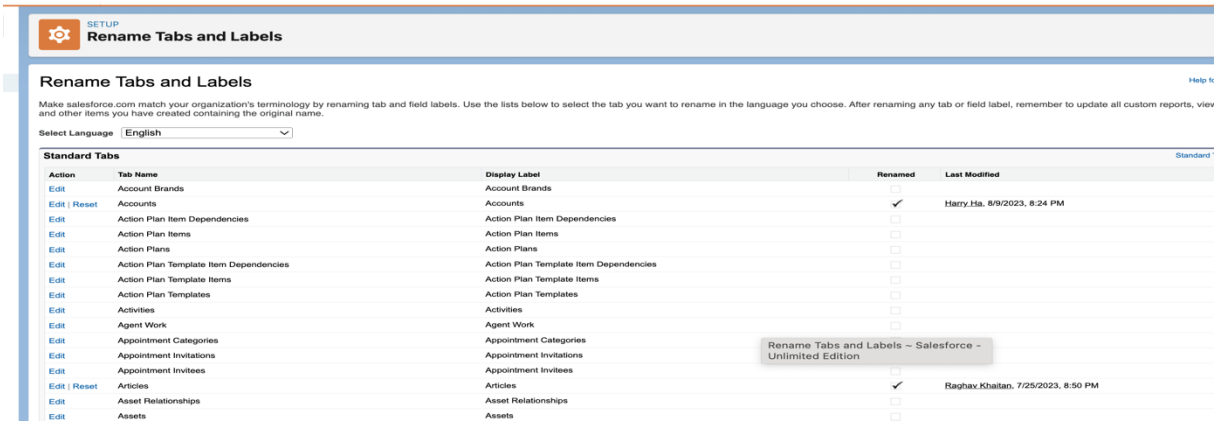
To work this for different languages used Translation-

To do Transaltions used different approaches-

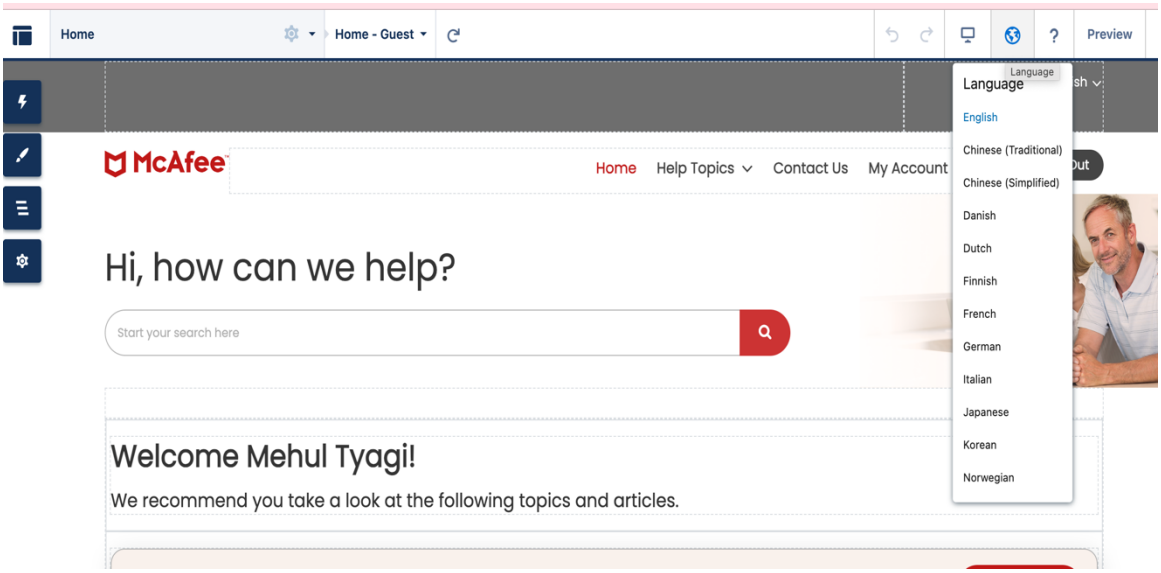
- For custom field label translation, flows, buttons, actions related objects name used Translates. Select the language - Component – Object – Aspect and then put the value in Field Label Translation for the field (name will be different for different components).



- For Standard fields and object label translation is done from Rename Tabs and Labels select the Language and then open the object for which you have to do translations.

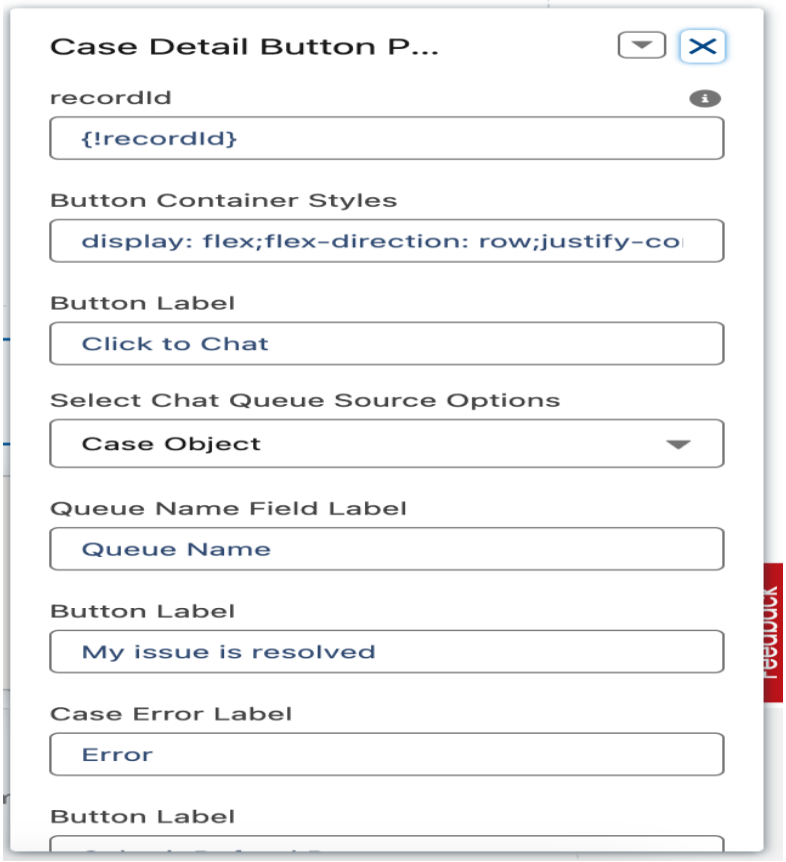


- For the Buttons for which we have given the label from site builder the translation is done from builder site only.

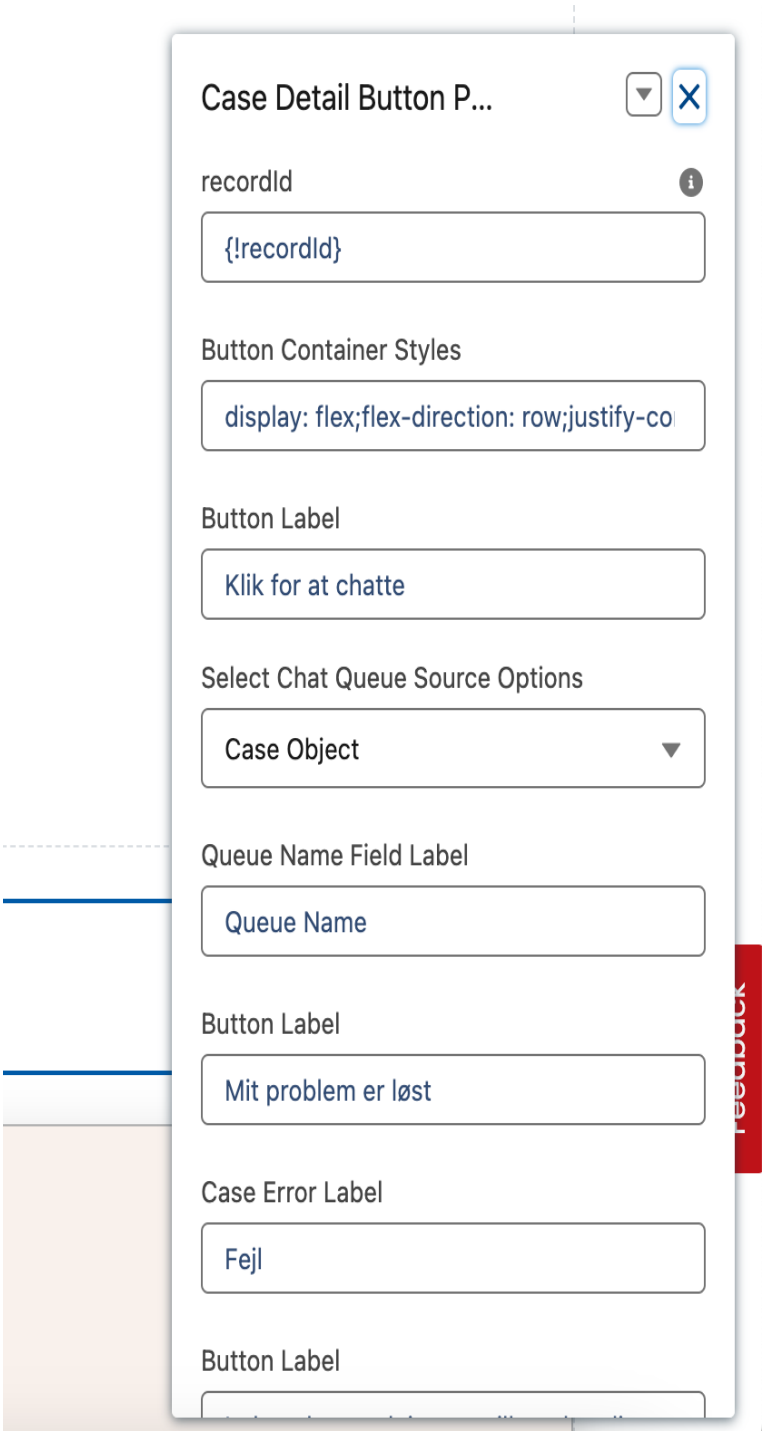


- Select the Language for which translation has to done
- Open the page on which the translation has to be done

The button lables in English



Button labels in Dutch



Use Case Implementation

- Customized the case detail and various other pages using the CSS override and changes in head markup on builder site.
- For Knowledge articles provided Most Recent and Most Popular filter option for sorting.
- For More Popular field updation used batch class and flow to update the number of counts or visits on the article.
- Provided Click To Chat button using LWC on Case Detail page on site to directly start chat from detail page.
- For Chat used Chatbot if agent is not available then chatbot handles the customer during outside business hours and in business hours the chatbot connects the customer with the agent.
- To make this site available for all locales did Translations in many languages.
- Used Apex Triggers for updating the Cases and Knowledge Articles in various scenarios.
- Provided Close Case button (Label – My issue is resolved) using LWC on case detail Page to directly close the case from the case detail page.
- Also provided View All Cases button using LWC to directly go back to all cases from Case detail page.

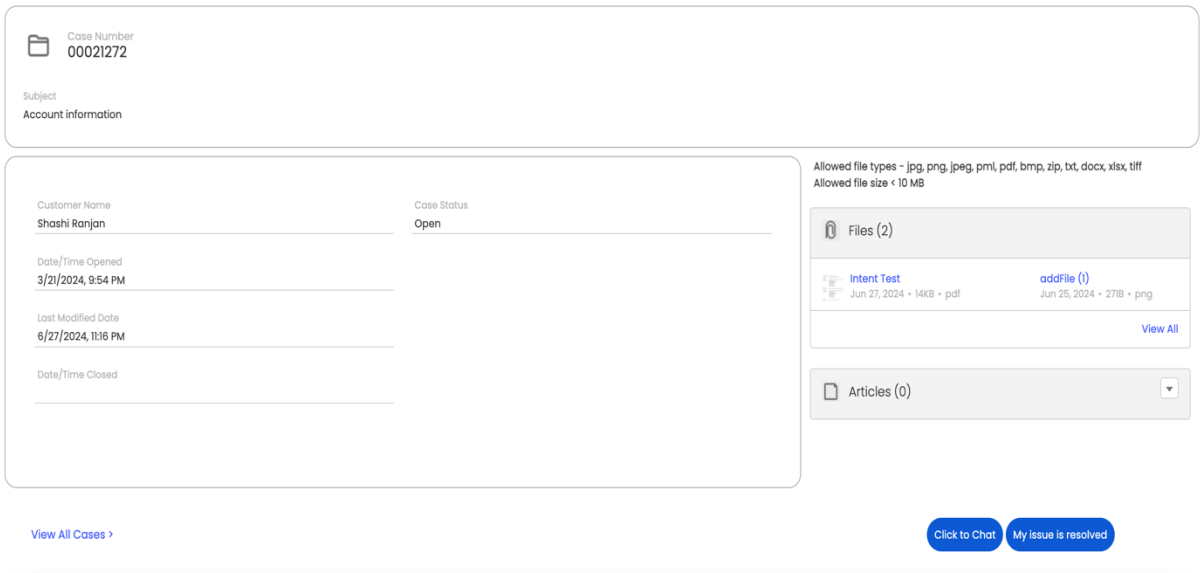
Click To Chat And Close Case Button

- Click To Chat Button on case detail page
- On click on this button an event is dispatched to open the chatbot window and the event is handled in html head tag of site builder.
- Parameters are passed from lwc to chatbot to. Connect with the correct agent for that customer.
- **Code Snippet of LWC to dispatch values from lwc to head markup**

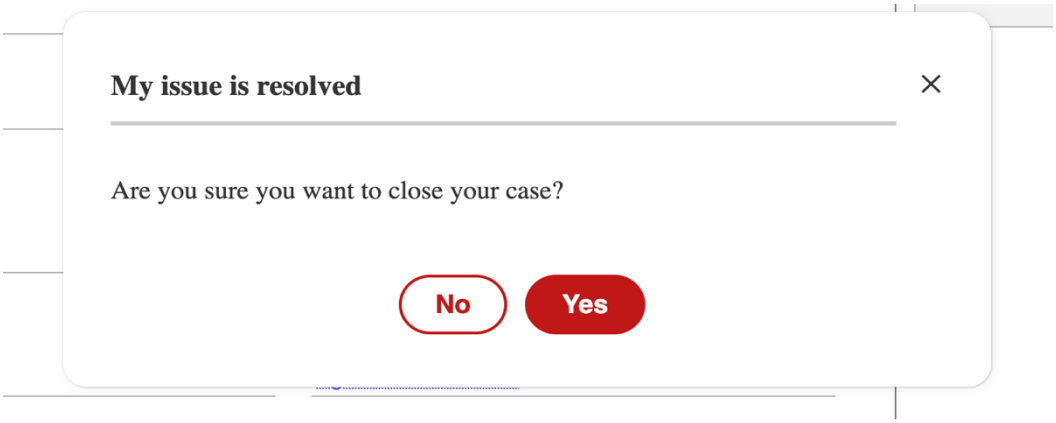
```
/**
 * @description On Button Click fire event to set field in site head tag
 */
startChat() {
    if (this.userInfo && this.userInfo.email) {
        console.log("Value in userinfo, queue----"+this.queueName);
        const chatEvent = new CustomEvent('onEmbeddedMessagingReady', {
            detail: {
                queueApiName: this.queueName,
                userInfo: this.userInfo
            },
            bubbles: true,
            composed: true
        });
        this.dispatchEvent(chatEvent);
        //this.fetchSupport();
         this.launchChat();
    } else if(this.isEmailRequired && (!this.userInfo || (this.userInfo && !this.userInfo.email))){
        console.log("Null userinfo, Email not entered, queue----"+this.queueName);
        this.dispatchEvent(
            new CustomEvent(
                'emailrequired',
                { bubbles: true, composed: true }
            )
        );

    }else {
        console.log("Null userinfo, queue----"+this.queueName);
        const chatEvent = new CustomEvent('onEmbeddedMessagingReady', {
            detail: {
                queueApiName: this.queueName
            },
            bubbles: true,
            composed: true
        });
        this.dispatchEvent(chatEvent);
        this.launchChat();
        //this.fetchSupport();
    }
}
```

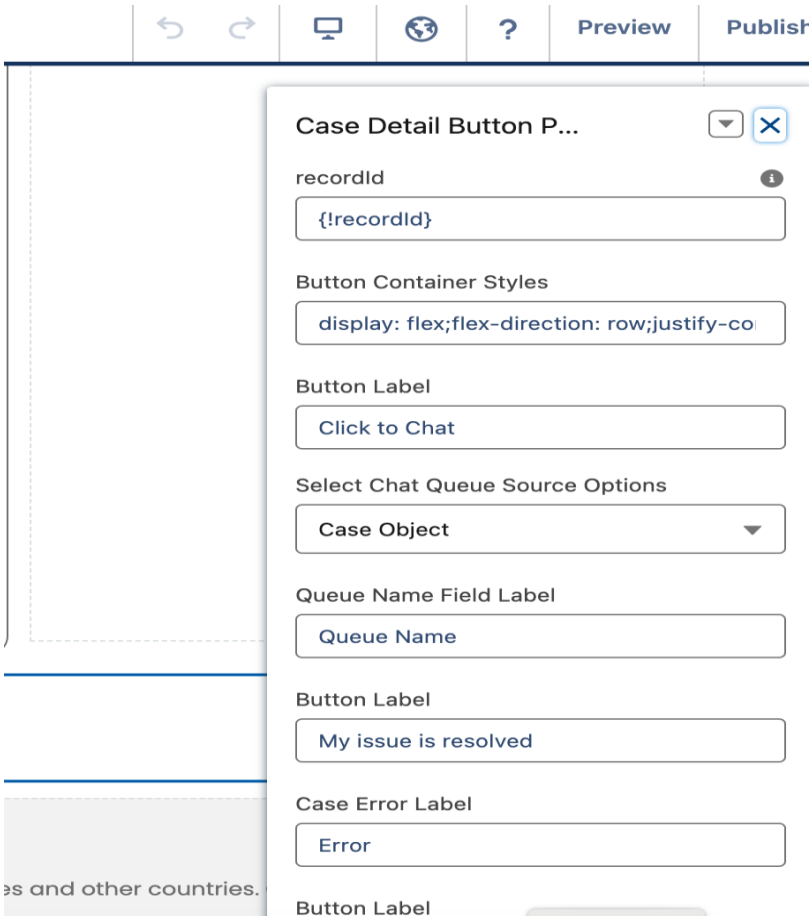
- Click to Chat and Close Case Button Close Case button(Label – My issue is resolved) on case detail page



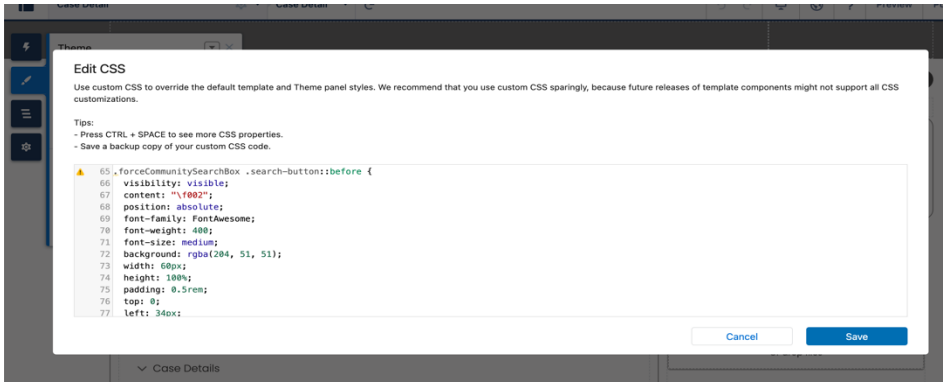
- When we click on **Click to Chat button** then chat window will open.
- When Click on **My issue is resolved** this modal will open After clicking on Yes the case id is sent to apex class to mark the case as closed.



- The Colour , Label and CSS of both the buttons is handled directly from the builder site. We can change these properties directly from builder without making any changes in code.



- **Some part of CSS Codes used for overwrite the standard css**

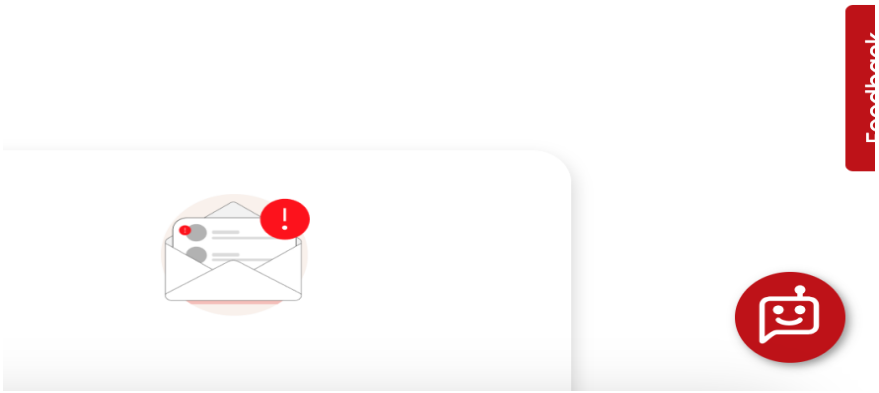


- **Java Script code used to handle the events dispatched from lwc for chatbot**

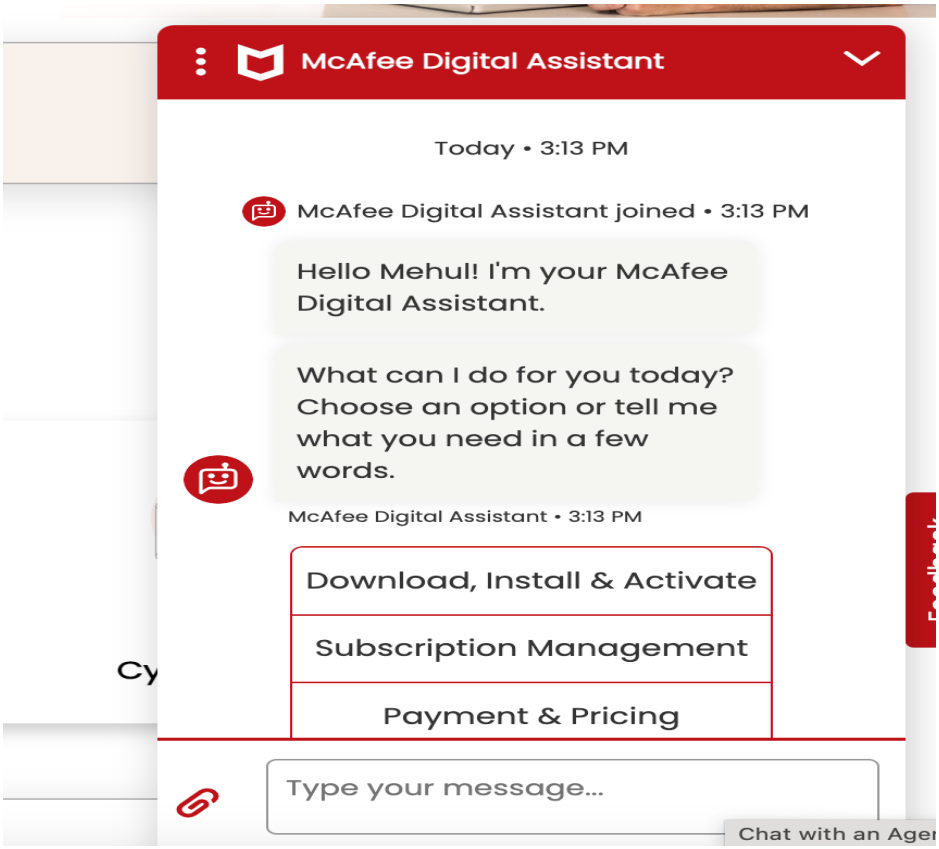
Salesforce And Chatbot Integration

- A chatbot was added to the Experience Cloud portal to provide real-time assistance to customers.
- The chatbot was integrated using Salesforce's Einstein Bot framework, which allowed customers to ask questions, receive instant answers, and be guided through common support queries.

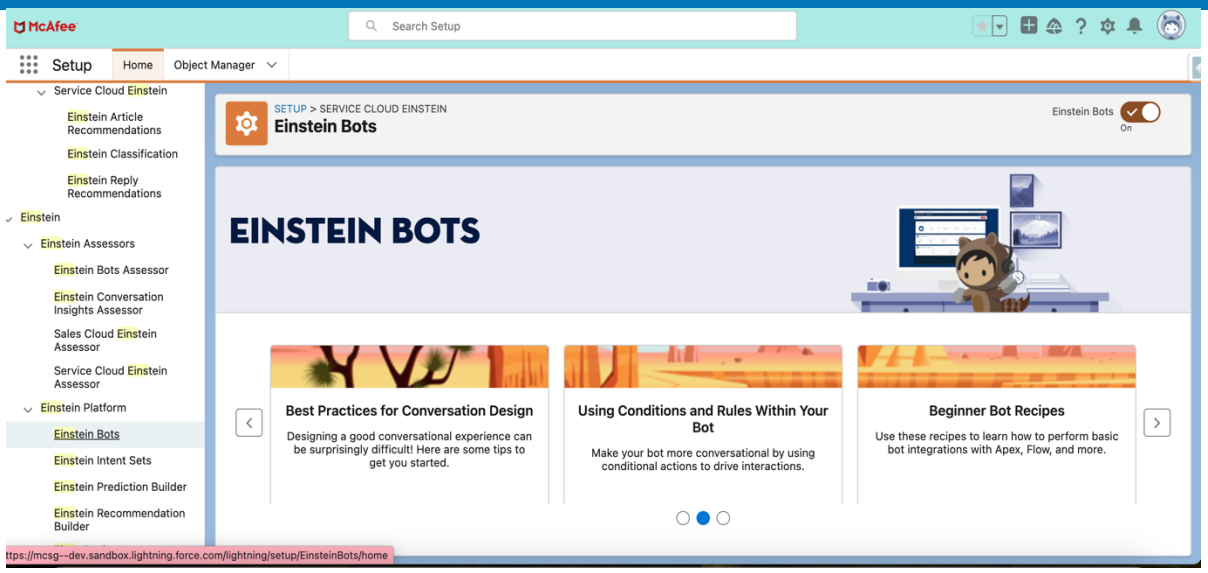
- **Chatbot Icon**



- **ChatBot Window**



- **Enabled Einstein Bots for Chatbot functionality**



Objective:

The goal of the chatbot integration is to enhance customer support by providing 24/7 assistance, reducing the workload on the support team, and offering instant responses to common customer queries. This chatbot is integrated into the Experience Cloud customer portal.

Platform Used: Salesforce's Einstein Bots framework was used for chatbot integration. This platform leverages AI to handle customer interactions and can be customized to meet specific business needs.

Key Features of the Chatbot:

Instant Response to FAQs: The chatbot was designed to answer frequently asked questions (FAQs) by pulling information from the Knowledge base. This feature allowed customers to receive quick answers without waiting for a live agent.

Integration Steps:
Chatbot Setup: Using the Einstein Bots setup, we created a new bot with a conversational flow specifically designed for customer support. The flow was configured to handle a variety of common use cases such as answering questions, providing knowledge articles, and creating cases.
The chatbot was trained with key FAQs and common scenarios gathered from ABC Company's support team to ensure it was knowledgeable about the company's offerings and support processes.

Dialog Design:

We created dialog in the Einstein Bot to handle different types of conversations. Dialog mapped out potential customer questions and responses the bot could give, ensuring that users received the correct answers or were guided to the right action.

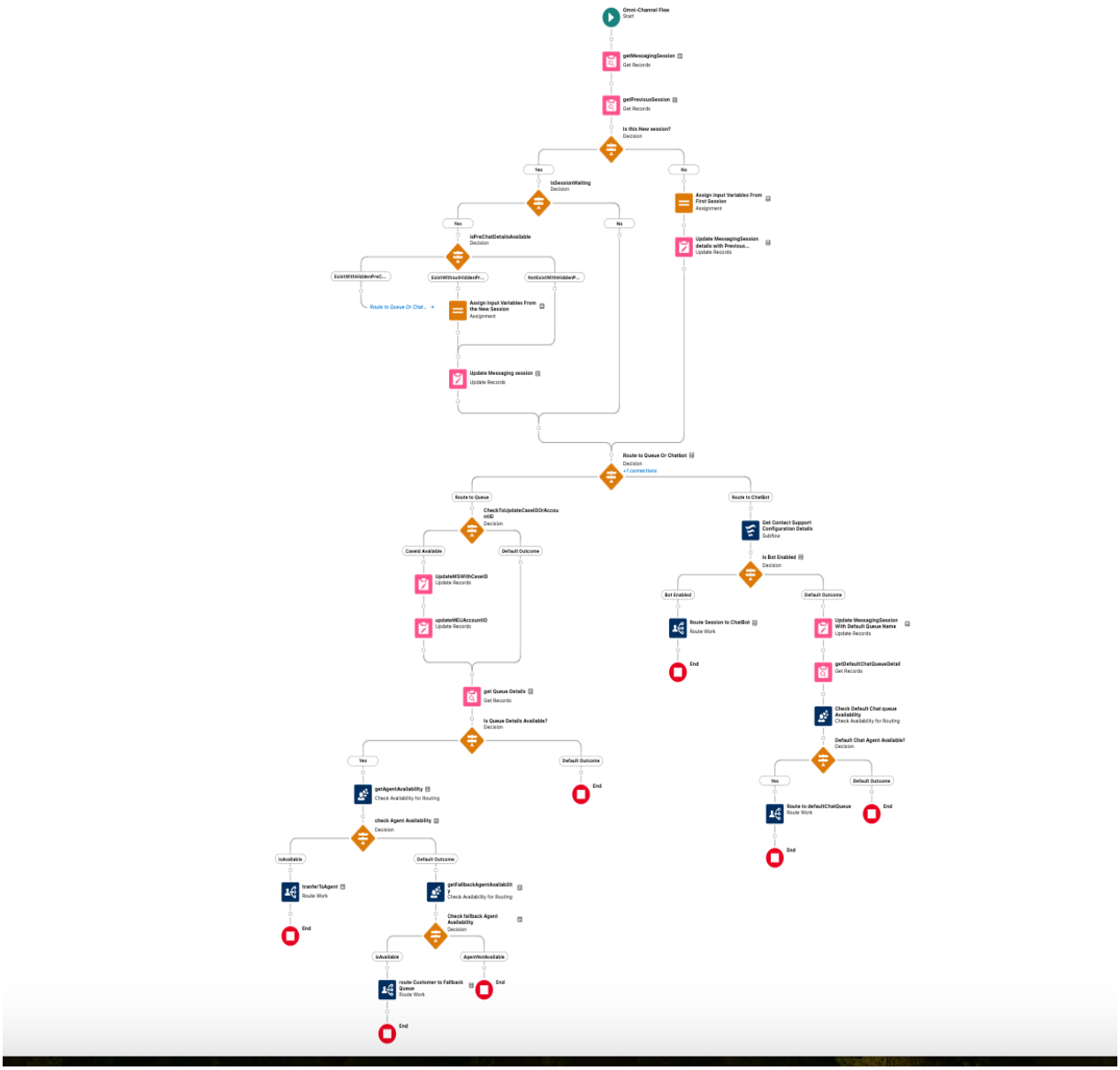
Chat Window Integration:

The chatbot was integrated into the Experience Cloud portal by adding a chat window that appeared on all support-related pages. Customers could easily start a conversation by clicking on the chat icon, which was accessible from both desktop and mobile versions of the portal. The chat window was customized using CSS Override and Java Sript to align with ABC Company's branding and theme layout.

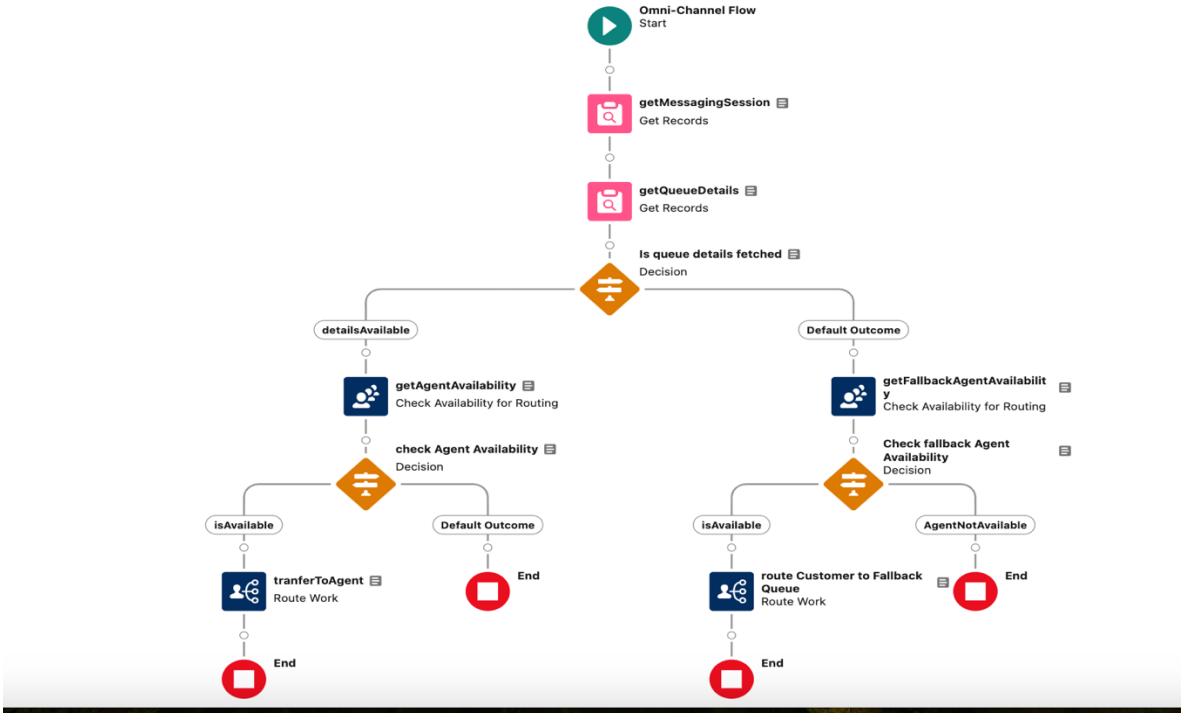
Live Agent Transfer:

Using the Einstein Bot + Live Agent integration, we enabled a seamless handoff between the bot and a live support agent. The handoff was triggered based on conditions (such as user requests or unresolvable issues), and the customer was notified that they would be connected to an agent. The chat history was passed to the agent to ensure a smooth transition.

- Inbound flow - Route to chatbot or Queue based on business usecase



- Outbound flow - Route Session to Available Agent



Analytics and Reporting:

The chatbot's performance was monitored using Salesforce's analytics features. Key metrics like customer satisfaction, number of queries resolved, bot-to-agent handoff rate, and the number of cases created were tracked to ensure the chatbot was effectively assisting customers.

Benefits of the Chatbot Integration:
24/7 Availability: Customers could receive assistance at any time, even outside business hours. This reduced the need for agents to handle repetitive tasks and allowed them to focus on more complex cases.
Quick Issue Resolution: The chatbot provided instant responses to common queries, which improved the overall customer experience and reduced wait times.

Reduction in Agent Workload:
By deflecting simple queries and guiding users to self-service options, the chatbot significantly reduced the number of support requests handled by live agents.