

NPS Survey Builder App Architecture Document

Executive Summary

Project Overview

The **NPS Survey Builder Application** is a comprehensive tool designed to help organizations create, customize, and distribute Net Promoter Score (NPS) surveys. The application provides an intuitive interface for designing survey forms, customizing emails, and analyzing customer feedback effectively. Below is an outline of its core features and functionalities:

1. Survey Design and Customization

- **Customizable Form Creation:** Users can create surveys tailored to their needs with a variety of question types, including:
 - Single-choice questions
 - Multiple-choice questions
 - Rating scales
 - Open-ended text responses
- **Scoring Mechanism:** Users can assign scores to specific answers, enabling easy evaluation of responses.
- **User-Friendly Interface:** Drag-and-drop functionality makes the survey creation process seamless and accessible for all users.

2. Survey Distribution

- **Multiple Delivery Channels:** Surveys can be sent via:
 - Email: Users can customize the email template within the application before sending it to recipients.
 - QR Code: A unique QR code is generated for each survey, allowing respondents to easily access and complete the form.
- **Bulk Contact Support:** Surveys can be sent to multiple contacts at once, saving time and effort.

3. Email Customization

- **Built-In Email Editor:** Users can design personalized email templates using a simple editor to match the tone and branding of their organization.
- **Preview Feature:** The email preview feature ensures that users can verify the design and content before sending.

4. Response Management and Analytics

- **Real-Time Results:** Once responses are submitted by customers, they are immediately displayed in the results dashboard.
- **Key Metrics:** The dashboard includes:
 - Response Rate: Percentage of recipients who submitted the survey.
 - NPS Score: Automatically calculated based on responses.
 - Customer Details: A detailed list of customers who submitted responses.
- **Data Export:** Option to export response data for further analysis or reporting.

5. User Experience

- **Accessible Design:** The application ensures a seamless experience across all devices, enabling respondents to complete surveys easily on desktops, tablets, or smartphones.
- **Automation Support:** Built-in features to automate follow-ups for incomplete surveys.

Scope

This document covers the architecture of the **NPS Survey Builder**, including its components, interactions, data flow, and security.

Introduction

Purpose

The purpose of this document is to provide a comprehensive overview of the architecture of the NPS Survey Builder Application, intended for developers, architects, and stakeholders involved in the project.

Background

The **NPS Survey Builder Application** is a comprehensive tool designed to help organizations create, customize, and distribute Net Promoter Score (NPS) surveys. The application provides an intuitive interface for designing survey forms, customizing emails, and analyzing customer feedback effectively.

System Overview

System Description

The NPS Survey Builder Application is a user-friendly platform for creating, distributing, and analyzing NPS surveys. It enables businesses to design customizable surveys, assign scores to responses, and distribute them via email or QR codes. The application also provides real-time analytics and detailed insights to drive customer satisfaction and business growth.

Key Features

1. Custom Survey Builder
 - Drag-and-drop interface with multiple question types (single-choice, multiple-choice, ratings, etc.).
 - Custom scoring for response evaluation.
2. Survey Distribution
 - Multi-channel delivery via email and QR codes.
 - Bulk contact upload for efficient distribution.
3. Email Customization
 - Built-in editor to design and preview personalized email templates.
4. Analytics and Reporting
 - Real-time response tracking, NPS score calculation, and response rate monitoring.
 - Exportable reports for further analysis.
5. Mobile-Friendly Design
 - Fully responsive surveys are accessible across all devices.
6. Data Security
 - Encrypted storage and role-based access control to ensure privacy.

Architecture Description

The NPS Survey Builder Application is designed with a scalable, modular, and secure architecture, ensuring seamless user interaction and efficient processing. The architecture leverages modern technologies and best practices to provide a reliable platform for survey creation, distribution, and analytics.

Architectural Principles

- **Modularity:** Separate components for Object and field.
- **Reusability:** Use of common controllers for multiple LWC components.
- **Scalability:** Designed to handle multiple objects and fields.

Technology Stack

- **Salesforce Platform**
- **Apex**
- **Lightning Web Components (LWC)**
- **Custom Object**

Architecture Diagram

View	Controller	Model	
LWC Components	Apex Controller	Custom Objects	Custom Label
designOptions	ChainedSurveyDistributionQueueable	Field_Mapping__c	BGImageRemove
existingSurveyForm	ExistingFormController	Question_Rule__c	Community_Site_Url
formBuilder	ImageUploadController	Question__c	Data_Not_Found_For_Specify_Record
formPreview	QuestionController	Response__c	DeleteRule
initSurvey	ResponseDataController	Score_Level__c	DeleteScoreLevel
npsHeader	ResponseMappingController	Survey_Distribution__c	EmailFolderName
questionDependency	ScheduledJobCleaner	Survey_Form__c	ErrorBGFileSize

questionEditor	ScheduledSurveyDistributionBatch	Survey_Recipient__c	ErrorDeleteScore
questionPreview	ScoreLevelController	Close_The_Loop__c	ErrorFetchData
		Driver_Feedback__c	
		SurveyFormPage__c	
questionTypeSelector	SurveyDeploymentController	Sharing Rule	ErrorFetchFile
recipientSelector	SurveyDistributionQueueable	Question__c	ErrorFetchRecord
responseDataPage	SurveyFormController	Survey_Form__c	ErrorFileSize
responseMapping	SurveyResultController	Field Mapping	ErrorGetData
scoreLevel	SurveyArchiveBatch	Score Level	ErrorHFSize
settingsSurveyForm	SurveyArchiveScheduler	Account	ErrorImageUpload
surveyDeployment	ObjectPrefixController	Question Rule	ErrorInitilaize
surveyObjectMapping	QuestionController	Response	ErrorLoad
surveyResult	ResponseDataController	Survey Distribution	ErrorLoadField
surveyTabset	ResponseMappingController	Close The Loop	ErrorLoadFile
closeWithResponse	SurveyStatsController	Driver Feedback	
thankYouPage	ResponseRateController	Static Resource	ErrorLoadScore
thankYouPreviewPage	ScheduledSurveyDistributionBatch	CHARTJS	ErrorRecordSave
translation	ScoreLevelController	HideLightningHeader	ErrorRemoveBGImage
surveyResult	StandardsReportsController	gaugeJS	ErrorRequiredFields
responseRate	SurveyArchiveBatch	Permission Set	ErrorSaveField
salesContactDriversChart	SurveyArchiveScheduler	Survey Builder Admin Permission	ErrorSaveResponse
driversLoyalty	SurveyDeploymentController	Survey Builder Guest User	ErrorSaveScoreLevel
accountSurveyStats	SurveyDistributionQueueable		ErrorStoreData
contactMerger	SurveyFormController	ContentAsset	FailedHIRemove
accountMerger	SurveyResultController	NPSSurveyLogo	FailedRemoveFooterImage

npsGaugeChart	SurveyStatsController		FooterImageRemove
npsComments		CustomApplication	HIRemove
npsMergedComments		NPS_Survey_Builder	Not_Data_Found
npsMergedPaginator			ResponseEmailTemplateName
contactSurveyStats		Custom Tab	RuleDelete
npsDashboard		Field_Mapping__c	RuleSaved
npsSummaryChart		NPS_Survey	SaveField
npsContactComments		Question_Rule__c	SuccessImageUpload
scoreLevel		Question__c	SuccessRecordSave
surveyDeployment		Response__c	SurveyArchive
driverLoyaltyQuestions		Score_Level__c	ThankYou_Page_Url
closeTheLoopPage		Survey_Distribution__c	UnknownError
		Survey_Form__c	UpdateScoreLevel
		Survey_Recipient__c	WrongURL
			checkRecordDelete
		FlexiPage	AddAndCheckField
		NPS_Survey	AddDistributionchannel
		New_NPS_Survey_UtilityBar	AddQuestion
		Survey_Form_Record_Page	AddRecipient
		NPS_Dashboard	AllowedTimeToRespond
		Close_Loop_With_Response	
		Close_The_Loop_Details	
		Layout	ArchiveDate
		Field_Mapping__c-Field Mapping	AutocloseDateMessage

		Layout	
		Question_Rule__c-Question Rule Layout	checkFieldMapping
		Question__c-Question Layout	confirmDeleteQuestion
		Response__c-Response Layout	Email_Limit
		Score_Level__c-Score Level Layout	DeleteQuestion
		Survey_Distribution__c-Survey Distribution Layout	Error_cloning_survey
		Survey_Form__c-Survey Form Layout	Error_Remove
		Survey_Recipient__c-Survey Recipient Layout	Error_Save_Record
			Error_While_Delete
		EmailTemplate	ErrorAutoCloseDate
		NPS_Survey	ErrorDeleteQuestion
		Survey_Archive_Completion	ErrorFetchDistribution
			ErrorFetchQuestion
		Custom Setting	ErrorFetchsurveyData
		SurveyFormPage	ErrorFileLoad
			ErrorHandle
			ErrorHandleData
			ErrorHandleHeader
			ErrorHanldeFormData
			ErrorHanldeSave
			ErrorLoadChart
			ErrorLoadContact
			ErrorLoadEmail

			ErrorLoadingDistributionHistory
			ErrorLoadSetting
			ErrorNavigation
			ErrorProcesing
			ErrorResponse
			ErrorResponseData
			ErrorSaveAndContinue
			ErrorSaveSurveySetting
			ErrorSaveTemplate
			ErrorSavingCategory
			ErrorSurveyRecipient
			ErrorTemplateContent
			ErrorUpdateOrder
			FailedCreateDistribution
			FailesDeleteQuestion
			Form_Cloned
			Invalid_configuration
			InvalidData
			InvalidID
			InvalidSurveyData
			Logo_File_Size
			Logo_Image_Remove
			MissingData
			No_Survey_Found

			NotDataRecieved
			NotValidID
			PageView
			Required_Field_Mapping
			RequiredBody
			RequiredSurveyId
			ScheduleDate
			SelectEmailTemplate
			Survey_Delete_Confirmation
			SurveySettingSave
			TemplateSaved
			UnableFileView
			UrlCopied

Component Design

Apex Classes

ChainedSurveyDistributionQueueable

- Description:

This Apex class processes a list of `Survey_Distribution__c` records sequentially using a chainable queueable job. It ensures secure handling of records, logs errors for individual records, and continues processing the remaining records in case of failures.

- **Method:**
Initializes the queueable job with a list of `Survey_Distribution__c` records and the index of the current record. Strips inaccessible fields for security compliance.

ContentVersionTriggerHandler

- **Description:**
This Apex class handles the creation of `ContentDocumentLink` records for `ContentVersion` objects. It ensures that new `ContentVersion` records are properly linked to entities while verifying security and avoiding duplicate links.
- **Method:**
Creates `ContentDocumentLink` records for a list of `ContentVersion` objects, ensuring no duplicate links exist and verifying security permissions before insertion.

ExistingFormController

- **Description:**
The `ExistingFormController` class is an Apex controller designed to handle operations related to the `Survey_Form__c` object and its associated records. This class facilitates secure data access and manipulation while ensuring compliance with field-level security (FLS) and object permissions. It is intended to be used in Lightning Experience or through Lightning Web Components (LWCs) and Aura components.
- **Methods:**
 - **getSurveyData:**
Retrieves the details of `Survey_Form__c` records linked to the current user, enforcing Field-Level Security (FLS) and object-level security before querying the data. Returns a list of records or throws an exception if permissions are insufficient.
 - **deleteSurveyForm:**

Deletes a specific `Survey_Form__c` record identified by its ID. Validates permissions to delete the record and handles DML or other runtime errors gracefully.

- **cloneSurveyFormWithRelatedRecords:**
Clones a `Survey_Form__c` record along with its related records (`Question__c`, `Score_Level__c`, `Field_Mapping__c`, and `Question_Rule__c`) while ensuring field accessibility and object security. The method dynamically handles FLS checks, builds secure queries, and maps related records appropriately. If permissions or conditions are not met, exceptions are raised.
- **handleClone:**
A wrapper for the cloning functionality that sets a savepoint to allow rollback in case of any errors during the cloning process.

ImageUploadController

- **Description:**
This Apex class is responsible for handling the upload, creation, and deletion of image files in Salesforce. It ensures the proper field-level security (FLS) checks and CRUD permissions are in place when interacting with the `ContentVersion` and `ContentDocumentLink` objects. The class allows files to be attached to records, retrieves related files, and manages the deletion of files based on user permissions.
- **Methods:**
 - **uploadFile:**
Uploads a file by creating a `ContentVersion` and linking it to a record using `ContentDocumentLink`.
 - **createContentVersion:**
Creates a `ContentVersion` record from the provided base64 file data and filename.

- **createContentLink:**
Creates a `ContentDocumentLink` to associate the uploaded `ContentVersion` with a specified record.
- **getVersionFiles:**
Retrieves all related `ContentVersion` records associated with a given record ID
- **deleteContentVersion:**
Deletes existing `ContentVersion` and its associated `ContentDocument` based on a provided record ID and file type.

QuestionController

- **Description:**
Handles survey questions and rules, including retrieval, creation, updating, and deletion. Ensures field-level security and user permissions are respected.
- **Methods:**
 - **getQuestionsWithOptions:**
Retrieves questions with non-empty options, excluding multi-select and ranking types.
 - **getAllQuestions:**
Fetches all questions for a form excluding a specified question by ID.
 - **getOptions:**
Returns the options for a given question.
 - **saveQuestionRule:**
Creates or updates a question rule with specified conditions and field-level security.
 - **getQuestionRules:**
Fetches all question rules for a form.
 - **deleteQuestionRule:**
Deletes a question rule by ID with necessary permissions checks.
 - **getQuestions:**
Fetches questions for a form, filtered by rule conditions and answer values.

ResponseDataController

- **Description:**
 - Handles the retrieval of response and file data for surveys, ensuring proper field and content document access checks are performed.
- **Methods:**
 - **getResponseData:**
 - Fetches survey responses for a given form and contact, including attachment counts, with necessary access checks.
 - **getFilesData:**
 - Retrieves files linked to a specific response, including access control for ContentDocument and ContentVersion.
 - **checkContentDocumentAccess:**
 - Checks user permissions for accessing ContentDocument, ContentVersion, and ContentDocumentLink records.
 - **checkFieldAccess:**
 - Validates field-level access for fields related to responses, questions, contacts, and survey forms.
 - **isTextFile:**
 - Determines if a file is a text-based file type (e.g., TXT, CSV, XML).

ResponseMappingController

- **Description:**
 - A controller for managing survey question mappings to the Salesforce object fields, handling field retrieval, mapping saving, and existing mappings retrieval with appropriate security checks and error handling.
- **Methods:**
 - **getQuestions:**
 - Retrieves survey questions for a specific form, ensuring Field-Level Security (FLS) checks.
 - **getFields:**
 - Retrieves the fields of a specified object for mapping, with FLS validation.
 - **saveFieldMappings:**

Saves or updates mappings between survey questions and object fields, with FLS and permission checks.

- **getExistingMappings:**

Retrieves existing question-to-field mappings for a given survey form with FLS validation.

- **saveResponses:**

Saves survey responses, including file uploads, with FLS and permission checks for related objects like Response__c and ContentVersion.

- **getSurveyDistribution:**

Retrieves a Survey_Distribution__c record by its ID.

ResponseTriggerHandler

- **Description:**

Handles the calculation of survey response scores and assigns appropriate labels and colors based on predefined scoring logic, using information from related `Question__c`, `Score_Level__c`, and `Response__c` records.

- **Methods:**

- **calculateScores:**

Iterates over new survey responses, calculates scores, updates field mappings, and applies score level mapping.

- **applyScoreLevelMapping:**

Applies the appropriate score label and color based on the response score and defined score ranges.

- **isStandardSurveyType:**

Checks if the question type is a standard survey type (e.g., NPS, CSAT, CES).

- **calculateStandardScore:**

Calculates a score for standard survey types (NPS, CSAT, CES) based on predefined scoring logic.

- **calculateResponseScore:**

Calculates the score for a response based on the question's specific conditions (e.g., correct answer or keyword match).

ScheduledJobCleaner

- **Description:**

A scheduled job cleaner performs security checks before deleting scheduled jobs (CronTrigger) with no next execution time. Ensures that the user has the necessary permissions to perform these operations.
- **Methods:**
 - **execute:**

Executes the scheduled job cleanup, checking if the user has the necessary permissions before calling deleteEmptyScheduledJobs.
 - **hasRequiredPermissions:**

Check if the user has the required permissions to access and delete CronTrigger records.
 - **deleteEmptyScheduledJobs:**

Queries for scheduled jobs with no next execution time and deletes them, ensuring the user has permissions and handling any exceptions that may arise.
 - **ScheduledJobCleanerException:**

A custom exception class is thrown in cases where permission checks fail or an error occurs during the scheduled job cleanup process.

ScheduledSurveyDistributionBatch

- **Description:**

A batch class that schedules and processes survey distributions based on their scheduled date. It handles security checks and field-level access before performing any actions on Survey Distribution and Survey Recipient records.
- **Methods:**
 - **execute:**

Runs the batch process on a schedule, executing the batch job with the specified batch size (50).
 - **start:**

Checks field-level security and access for Survey Distribution and Survey Recipient records, then returns a query locator to find

scheduled survey distributions that need processing.

■ **execute:**

Processes each survey distribution in the batch, verifies update access, and chains the next set of processing using a Queueable job if necessary.

■ **finish:**

Placeholder method for the finish step in the batch process, with no additional operations.

■ **SecurityException:**

A custom exception is thrown when there are insufficient access rights to the required records or fields.

ScoreLevelController

- **Description:**

A controller class to manage `Score_Level__c` records with field-level security checks. It provides methods to retrieve, create, update, and delete Score Level records while ensuring proper permissions.

- **Method:**

- **deleteScoreLevel:**

Deletes a specific `Score_Level__c` record after validating field and object permissions.

- **saveScoreLevels:**

Creates or updates a list of `Score_Level__c` records linked to a specific survey form and deletes any removed records. Ensures field-level and object-level security.

- **getScoreLevelsByFormId:**

Retrieves all `Score_Level__c` records associated with a given survey form ID, sorted by the minimum score range. Validates field-level security before querying..

SurveyDeploymentController

- **Description:**

A Salesforce Apex controller that manages survey deployment operations with robust security checks, handling survey distribution, recipient management, email templates, and QR code generation.

- **Methods:**

- **checkFieldPermissions:**

Utility method to verify field-level permissions for a given object and set of fields.

- **getContactList:**

Retrieves a list of contacts with their account information, implementing security checks.

- **createSurveyDistribution:**

Creates a new survey distribution record with recipient details and scheduling options. Handles both immediate and scheduled deployments.

- **scheduleDistribution:**

Schedules a survey distribution for a future date using the Salesforce scheduling system.

- **scheduleArchiveResponse:**

Schedules the archiving of survey responses for a specific date.

- **getEmailTemplates:**

Retrieves email templates from a specified folder with security checks.

- **getTemplateContent:**

Fetches the content of a specific email template.

- **saveEmailTemplate:**

Updates an email template's subject and body with proper security validation.

- **getDistributionHistory:**

Retrieves the distribution history for a specific survey.

- **createRecipients:**

Helper method to create survey recipient records.

- **generateQRCode:**

Creates a QR code URL for survey access.

SurveyDistributionQueueable

- **Description:**

SurveyDistributionQueueable is a Salesforce Apex class that implements the

Queueable interface to handle asynchronous processing of survey distributions, managing email sending and recipient updates in batches.

- **Methods:**

- **execute:**

Main method that processes survey distributions by sending emails to recipients in batches of 100, handling both HTML and plain text emails with QR codes or buttons.

- **handleDistributionStatus:**

Manages the distribution status after sending emails, checks for remaining recipients, and initiates the next batch or completes the distribution.

- **scheduleNextRecurrence:**

Creates and schedules the next recurring distribution if configured, cloning the distribution and recipient records with proper settings.

- **handleError:**

Error handling method that updates the distribution status and stores error messages when failures occur.

SurveyFormController

- **Description:**

A Salesforce Apex controller that handles Survey Form and Question operations with field-level security (FLS) enforcement.

- **Methods:**

- **saveSurveyForm:**

Inserts or updates a Survey Form record after validating FLS permissions and strips inaccessible fields.

- **upsertQuestion:**
Creates or updates a Question record associated with a Survey Form, handling properties like title, help text, question type, etc.
- **getQuestions:**
Retrieves all Questions associated with a specific Survey Form ID in order of their question index.
- **getSurveyData:**
Fetches the complete details of a Survey Form record by its ID, including all styling and configuration fields.
- **deleteQuestion:**
Removes a Question record after validating deletion permissions.
- **getSelectedQuestion:**
Retrieves a specific Question record by its ID with all its properties.
- **updateQuestionOrder:**
Updates the order/sequence of questions by modifying their Question_Index__c field.

SurveyResultController

- **Description:**
SurveyResultController: A Salesforce Apex controller that handles Survey Response data and recipient management.
- **Methods:**
 - **getResponseData:**
Retrieves all Response records associated with a Survey Form, including related contact information, questions, and scoring details.
 - **getResponseFiles:**
Helper method that fetches all files/attachments linked to Survey Responses.
 - **getSurveyRecipient:**
Returns a list of contact IDs that have received and completed the survey through Survey Distribution records.

SurveyArchiveBatch

- **Description:**

The **SurveyArchiveBatch** class is a batch Apex implementation that archives

Response__c records related to a specific Survey_Distribution__c if the auto-archive date has passed. It also sends a summary email to the running user upon completion using a predefined email template..

- **Methods:**

- **SurveyArchiveBatch**

Initializes the batch with the provided distribution ID and fetches the email template.

- **fetchEmailTemplate:**

Retrieves the active email template specified by the label.

- **start(Database.BatchableContext bc):**

Returns a query locator for eligible Response__c records to be Archived.

- **execute(Database.BatchableContext bc, List<Response__c> scope):** Updates the status of each response in the batch to "Archived".

-

- **handleSaveResults:**

Handles errors for failed response record updates.

-

- **finish(Database.BatchableContext bc):**

Sends a completion email to the user with archive details.

-

- **handleException:**

Logs contextual exception messages.

-

- **BatchArchiveException:**

Custom exception for errors in the batch process.

SurveyArchiveScheduler

- **Description:**

This apex class schedule archive survey form.

- **Methods:**

- **SurveyArchiveSchedule**

Constructor to initialize the scheduler with a distribution ID.

- **execute:**

Executes the batch process for archiving surveys.

ObjectPrefixController

- **Description:**

The **ObjectPrefixController** class securely retrieves the key prefixes for Contact,

Survey Form, and Survey Distribution objects. It ensures object and field-level

access control before returning prefix values.

- **Methods:**

- **getObjectPrefixes**

Returns a map of object API names to their key prefixes after verifying object and field-level access.

LWC Components

designOptions

- **Description:**

In this component, the user can design their survey form where they can design for

the questions and answers color, text style. Upload header, footer, logo and

background Image uploaded, Also customize the number of questions displayed on the survey Form.

existingSurveyForm

- **Description:**

In this component, all saved survey form records are displayed in the data table

Where users can edit, share, clone, and delete the record.

formBuilder

- **Description:**

In this component, different data types of questions are created, edited and preview the questions.

formPreview

- **Description:**

In this component, customized forms are sent to the user and displayed in the experience site.

initSurvey

- **Description:**

This is the home page of the application where user can fill the survey form name, Survey descriptions and survey form type.

npsHeader

- **Description:**

In this component, header navigation is displayed on each lwc page.

questionDependency

- **Description:**

In this component, add rules for the picklist questions.

questionEditor

- **Description:**

In this component, created questions are preview on the right side of the page.

questionPreview

- **Description:**

In this component, user can preview their customized survey form from the preview icon.

questionTypeSelector

- **Description:**

In this component, user can select different types of data-type.

recipientSelector

- **Description:**

In this component, the user can send the survey form to the selective customer through Email or QR code.

responseDataPage

- **Description:**

In this component, user can see the received responses from their customer.

responseMapping

- **Description:**

In this component, the user can map questions with selected object fields.

scoreLevel

- **Description:**

In this component, the user can set the score color, label, and score of the questions.

settingsSurveyForm

- **Description:**

In this component, customized survey form when form response is ended, in how

many days response is get.

surveyDeployment

- **Description:**

In this component, user can send the survey form to their customers through Email or QR code, also select multiple customers at a time, select the email template and schedule time when form is send and in the sent records are displayed in the data-table.

surveyObjectMapping

- **Description:**

In this component,user choose object to mapped the fields, display current user and also select the survey form is cloned or not.

surveyResult

- **Description:**

In this component, the user can do conditional field mapping of the mapped Salesforce Object.

surveyTabset

- **Description:**

In this component, where different types of LWC pages are called for the survey form customization.

thankYouPage

- **Description:**

In this component,user can customize the thank you message after submit the

survey form.

thankYouPreviewPage

- **Description:**

In this component, User can preview their thank you message on the right side of the thank you page.

translation

- **Description:**

Currently this component is used to display the message for upcoming translation feature.