How to implement Accessability Feature.

1. Paste accessability folder in your project directory. Let’s assume I placed in **my\_project\_folder/public** directory.
2. Add css of accessability in your file. E.g.- **public/accessability/css/accessability.css**

<link rel="stylesheet" href="/gvc/public/accessability/css/accesibility-style.css">

1. Add js of accessability in your file. E.g.- **public/accessability/js/accessability.js. Also do the configuration there is a function - RenderAccessability(options); This is the core function of render.js.**

In this function you can pass button location and button color and widgets languages (in my case I’m determining the language based on language toggle button, but you can pass language directly also)

<!-- Accessibility Widget Script -->

<script src="public/accessability/js/accessability.js?v=<?= time() ?>"></script>

<script>

    var isPageLoaded = isAllAjaxComplete = isAccessabilityRendered = false;

    $(document).ajaxStop(function(){

        //WHEN ALL AJAX REQUEST COMPLETED

        isAllAjaxComplete = true;

    });

    document.addEventListener('DOMContentLoaded', function() {

        //WHEN PAGE LOADED SUCCESSFULLY

        isPageLoaded = true;

    });

    function renderAccessabilityWidget(params) {

        let languageElement = document.getElementById("language-toggle");

        let selected\_lang = 'en';

        if(languageElement){

            selected\_lang = (languageElement.checked)?'en':'gu';

            //ONCHANGE ON LANGUAGE RE-RENDER ACCESSABILTIY WITH SELECTED OPTION

            languageElement.addEventListener('click', function(e){

                selected\_lang = (this.checked)?'en':'gu';

                WIDGET\_SPEECH\_LANGUAGE = selected\_lang;//SET LANGUATE TO WIDGET GLOBAL VARIABLE

            })

        }

        const accessabilityBtnHtml = `<button class="btn btn-primary" type="button" title="Accessability Features">

        <i class="fa fa-wheelchair" aria-hidden="true"></i></button>`;

        const options = {

            backgroundColor: '#FF9800', //BACKGROUND COLOR LIKE red, blue, #000(black)

            accessabilityBtnContainer: 'accessabilityBtnSection', //ID OF CONTAINER WHERE YOU WANT TO RENDER START ACCESSIBILITY BUTTON

            accessabilityBtn : accessabilityBtnHtml, //pass any html or text (If you'll not pass anything then default button will appear)

            speechLanguage: selected\_lang, //default(en-english), you can pass gu-gujarati, hi-hindi

        }

        RenderAccessability(options); //DEFINED IN ACCESSABILITY.JS LIBRARRY

    }

    function intervalFunction(){

        if(isPageLoaded && isAllAjaxComplete ){

            //  IF PAGE AND AJAX REQUEST COMPLETED

            clearInterval(renderAccessInterval);//CLEAR INTERVVAL

            renderAccessabilityWidget(); //RENDER ACCESSABILITY WIDGET

            isAccessabilityRendered = true;//FLAG TO ADD DRAGGABLE FEATURE IN ACCESSABILITY WIDGET

        }

    }

    let renderAccessInterval = setInterval(intervalFunction, 500);//execute function - intervalFunction every 0.5 sec

</script>

1. If you want draggable feature in accessabilty then add draggable js folder (If you add draggable then you must have to run your project on production or a localhost like - **http://localhost/documentation/accessability\_imp/index.html**

<!-- Draggable Js Start -->

<script type="module" src="public/draggable/draggable.js"></script>

<!-- Draggable Js End -->

1. Inside dragglable folder there is a file draggable.js. In this you’ve to pass your main container id of your accessabilty element. However already configured if later you changes the id of accessability.js then you’ve change the draggable.js file also.

import { FDraggable } from './FDraggable.js';

let renderDraggableInterval = setInterval(() => {

    if(isAccessabilityRendered){

        clearInterval(renderDraggableInterval);

        const options = {

            'elementId' : 'main\_w\_container', // myDraggleElement - DRAGGLE ELEMENT ID

            'elementHeaderId' : 'widget\_header' //THERE MUST HEADER ELEMENT ALSO

        }

        let myDraggable = new FDraggable(options);

    }

}, 500);