در این قسمت می خواهیم به افزونه قابلیتی دهیم که با استفاده از آن کابر قادر باشد تا متون موجود در کامپوننت انتخاب شده را ترجمه کند. بعضی موقع ها کامپوننت هایی که مورد استفاده کاربر قرار میگیرد دارای متونی هستند که متفاوت با زبان اصلی پروژه است و ترجمه و تغییر این متون کار دشواری است حتی با استفاده از سایت های ترجمه مانند مترجم گوگل نیاز به زمان زیادی است که روند پیشرفت پروژه را کند می کند. این قابلیت در جهت کمک به این امر طراحی شده و کاربر بدون نیاز به نمونه برداری از متن و مراجعه به سایت ترجمه قادر باشد به راحت ترین روش و با کمترین زمان ممکن متون را به هر زبانی که نیاز دارد ترجمه کند.

در مرحله اول ما باید یک گزینه در افزونه اضافه کنیم که کاربر با فعال کردن این گزینه به برنامه این پیام را ارسال کند که قصد ترجمه متون را دارد :

Html

 <div class="form-check form-switch">

      <input class="form-check-input" type="checkbox"  id="translate">

      <label class="form-check-label" for="flexSwitchCheckDefault">ترجمه متن</label>

    </div>

با فعال شدن این چک باکس مجوز ترجمه متون صادر می شود.

سپس در اسکریپت داخلی فایل html زمانی که دکمه اعمال کلیک می شود ، باید مقدار "ترجمه متن" ارسال شود. مقدار آن برابر با true یا false است. کد های افزوده شده با رنگ قرمز مشخص شده اند

      document.getElementById('submit').onclick=()=>{

        const targetlanguage= mycombo.value;

const textAlignHorizontal=document.getElementById('textAlignHorizontal').checked;

        const reversenode=document.getElementById('reverse-node').checked;

        const translatetext=document.getElementById('translate').checked;

        parent.postMessage(

{

pluginMessage:

{textAlignHorizontal,reversenode,translatetext,targetlanguage}},'\*')

      }

    </script>

سپس ما نیاز به یک لیستی از زبان ها داریم که کاربر زبان مورد نیاز خود را برای ترجمه متن انتخاب کند

پس از چک باکس ترجمه متن ، یک لیست باکس قرار می دهیم.

 <div id="langcombo" style="width:100px;height:20px;"></div>

پس از ایجاد لیست باکس، نوبت به گزینه های مورد انتخاب است.در قسمت اسکریپت داخلی html آپشن های لیست باکس را تعریف میکنیم. برای سادگی کد اینجا از سه زبان فارسی ، انگلیسی و عربی استفاده می کنیم.

 let val='';

      let langcombo=Combo.New(document.getElementById('langcombo'));

      langcombo.options(

        [

         'فارسی',

         'العربیه',

         'English'

        ]);

        langcombo.value='فارسی'

        langcombo.onchange=function(e, combo) {

        val=combo.value;

      }

در اینجا لیست باکس بر اساس id دریافت می شود. سپس اپشن ها تعریف شده و مقدار پیشفرض را هم برابر با "فارسی" قرار می دهیم. با تغییر دادن مقدار لیست باکس توسط کاربر، تابع حساس به event فعال شده و مقدار جدید را در متغیر val ذخیره می کند.

در اخر بعد از زدن دکمه اعمال مقدار آن به همراه بقیه اطلاعات ارسال خواهد شد :

      document.getElementById('submit').onclick=()=>{

        const targetlanguage= langcombo.value;

        const textAlignHorizontal=document.getElementById('textAlignHorizontal').checked;

        const reversenode=document.getElementById('reverse-node').checked;

        const translatetext=document.getElementById('translate').checked;

        parent.postMessage({pluginMessage:{textAlignHorizontal,reversenode,translatetext,targetlanguage}},'\*')

      }

کار ما در قسمت ui به اتمام رسید و در عملیات مربوط به قسمت فایل js باقی مانده است. در مرحله اول مقادیر دکمه ترجمه با زبان انتخابی دریافت میشود.

var reversenode=false;

var textAlignHorizontal=false;

var targetlanguage="";

var translate=false;

var resultdata="";

figma.ui.onmessage=pluginMessage=>

{

  textAlignHorizontal=pluginMessage.textAlignHorizontal;

  reversenode=pluginMessage.reversenode;

  translate=pluginMessage.translatetext;

  targetlanguage=pluginMessage.targetlanguage;

  main()

}

در تابع main شرط به همراه فراخوانی تابع ترجمه درج می شود:

if(translate==true)

      translatetext(node)

شی انتخاب شده به تابع ترجمه ارسال می شود تا بررسی شود. تابع به صورت زیر نوشته می شود:

function translatetext(node)

{

  const children = node.children;

for (const child of children)

{

  if(child.type==="TEXT")

  {

    (async () => {

      const text = child

      await figma.loadFontAsync(text.fontName)

     var text\_for\_translate=text.characters

     var langAbbreviation=targetlanguage\_to\_Abbreviation(targetlanguage)

     var translated\_text= sendRequestToGoogle(text\_for\_translate,"auto",langAbbreviation)

      text.characters=(await translated\_text)

    })()

  }

  if(haschild(child))

    {

      translatetext(child);

    }

}

}

سپس تمام فرزندان node در هر مرحله داخل حلقه قرار میگیرد و اگر فرزندی هم دارای فرزندان دیگری باشد، با تابع haschild برسی می شود، در صورتی که مقدار تابع true باشد تابع ترجمه به صورت بازگشتی فراخوانی می شود. در داخل حلقه ویژگی نوع node بررسی می شود که آیا برابر با text است یا خیر. در صورت برقرار بودن شرط نوبت به ویرایش متن node موجود در حلقه می شود که برای تغییر محتوای متن نیاز است که فونت متن بارگزاری شود. محتوای متن داخل ویژگی characters قرار دارد.

برای ترجمه متن از سایت مترجم گوگل استفاده می کنیم. ما نیاز به مخفف زبانی که قرار است ترجمه شود داریم بر فرض مثال مخفف زبان فارسی در مترجم گوگل fa است. کار تابع targetlanguage\_to\_Abbreviation به این صورت است که مخفف زبان انتخاب شده را به دست می آورد. تابع sendRequestToGoogle اطلاعات جمع اوری شده از جمله متن و زبان ترجمه را به سمت سایت مترجم گوگل ارسال میکند و محتوای متن ترجمه شده را بر می گرداند. عمل اصلی ترجمه کردن در این تابع انجام می پذیرد. سپس در متغیر translated\_text ذخیره می شود و در ویژگی متن node انتخابی جایگزین می شود.

تابع targetlanguage\_to\_Abbreviation :

 function targetlanguage\_to\_Abbreviation(lang)

 {

  let Abbreviation;

  switch(lang) {

    case "فارسی":

      Abbreviation="fa"

      break;

    case "العربیه":

      Abbreviation="ar"

      break;

      case "English":

        Abbreviation="en"

        break;

    default:

      Abbreviation="fa"

  }

  return Abbreviation;

 }

تابع sendRequestToGoogle :

async function sendRequestToGoogle(word,sourceLang,targetLang)

  {

    const url = `https://translate.googleapis.com/translate\_a/single?client=gtx&sl=${sourceLang}&tl=${targetLang}&dt=t&dt=bd&dj=1&q=${encodeURIComponent(

    word

  )}`;

    let result = await fetch(url);

    let resultdata = await result.json();

    let textsum="";

   resultdata.sentences.forEach(element => {

    textsum+=element['trans']

   });

    return textsum;

  }