



Building a Name Pronunciation Database

With Power Apps and Power Automate

Lindsay Shelton
Application Programmer

January 18,
2022



Lindsay Shelton, Application Programmer
From Kansas City Metro Area, US

Microsoft Certified Trainer

Spent 10 years as a teacher before switching
careers

Works with everything from Power Platform
adoption to SPFx web parts to Azure APIM



 <https://www.linkedin.com/in/lindsayshelton>

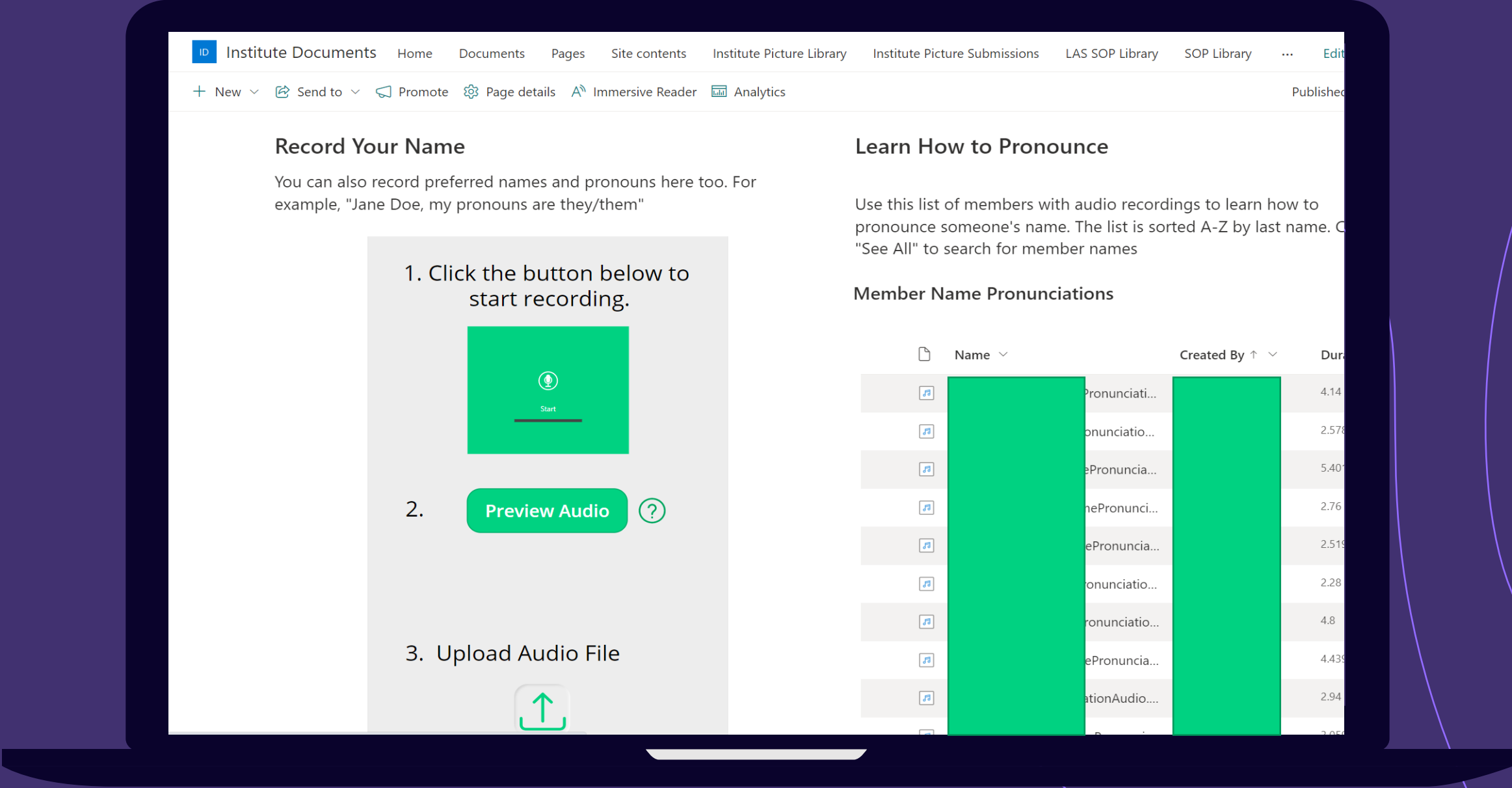
 <https://lindsayshelton.com>

 @lshelton_tech

“Our names are our identities...closely linked to our heritage. Mispronouncing a name can be experienced as a kind of microaggression, as the person may feel put down, invisible, or unworthy.”

- Marie-Therese Claes, Ph.D

Being a medical research institute, we have employees from all over the globe. We celebrate diversity, so what we were doing to support all of the wonderfully unique users we have?

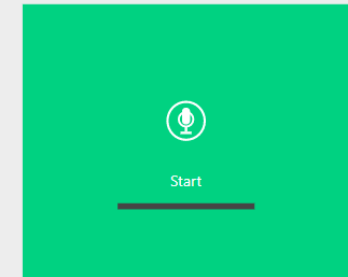


So, we created a place where our users could record and store name pronunciations!

What is this solution actually made up of?

Power App with a Power Automate flow sitting on a SharePoint page and a document library

1. Click the button below to start recording.

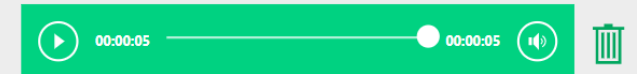


2.

Preview Audio



Lindsay Shelton-NamePronunciationAudio
4.62

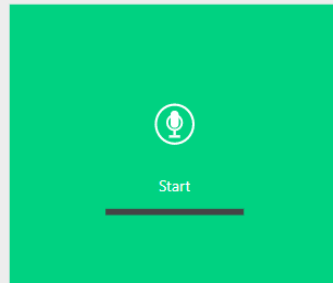


3. Upload Audio File



How does it work on the front-end?

1. Click the button below to start recording.



2.

Preview Audio



Lindsay Shelton-NamePronunciationAudio
4.62



3. Upload Audio File



1. User clicks to start recording, records name, and clicks the same button to stop the recording.
2. User clicks Preview Audio button, which populates the Play Audio control, which they can use to play and then delete the audio if they don't like it.
3. Repeat steps 1 and 2 as many times as desired.
4. When the user is happy with the audio, they click the Upload Audio File button.
5. The file is uploaded via a Power Automate flow to a SharePoint document library, which is surfaced on the same page that hosts this Power App.

The Power App

Uses some basic media controls –

- the microphone
- the audio control to play the recording back

Plus –

- a gallery that stores the audio recordings, displaying the most recent recording
- a button that populates the gallery with the latest recording done
- an upload button that is hooked up to a Power Automate workflow (it's neumorphic to be extra)

Power Apps | GenericNamePronunciationAudioRecorder

Environment
Personal Productivity

LS

Back | Undo | Redo | Insert | Add data | New screen | Theme | Settings | ...

OnStop = fx

Tree view

Screens Components

Search

New screen

App

Screen1

- Image4
- Group3
 - Step2
 - Step3
- Group1
 - Step1
 - QuestionMark1
 - PreviewAudio
 - TextInput1
 - Audio1
 - Microphone1
- Gallery1

Screen2

Format text | Remove formatting | Find and replace

Start

2. Preview Audio

3. Upload Audio File

Display mode Edit

Visible On

Position X: 177 Y: 169

Size Width: 287 Height: 240

Color A

Border 0

Image None

Image position Fill

Disabled color A

Pressed color A

Hover color A

Tooltip Click here to start and stop your recording. Keep it short with minimal background

Tab index 0

Screen1

Microphone1

60 %

```
Set(NowVar, Now());
Set(TempRecordingVar, Microphone1.Audio);
Set(tempJSON, JSON(Microphone1.Audio, IncludeBinaryData));
Set(StringB64Var, Mid(tempJSON, 25, Len(tempJSON)-25));
Set(HideTempAudioVar, true)
```

OnStop

Tree view

Screens

Components

Search

+

New screen

App

Screen1

Image4

>

Group3

Step2

Step3

>

Group1

Step1

QuestionMark1

PreviewAudio

TextInput1

fx

=

Set(NowVar, Now());

Set(TempRecordingVar, Microphone1.Audio);

Set(tempJSON, JSON(Microphone1.Audio, IncludeBinaryData));

Set(StringB64Var, Mid(tempJSON, 25, Len(tempJSON)-25));

Set(HideTempAudioVar, true)

Format text

Remove formatting

Find and replace

Microphone Control - OnStop ?.

Prev

00:00:00

Power Apps | GenericNamePronunciationAudioRecorder

Environment
Personal Productivity

Back | Undo | Redo | Insert | Open Sans | 24 | Semibold | Color | Background color | Settings

OnSelect =

Collect(TempSpeechRecCol,{
Audio: TempRecordingVar,
Duration: Audio1.Duration,
RecordingName: TextInput1.Text,
Recording: StringB64Var
});
Reset(TextInput1);

Format text | Remove formatting | Find and replace

Tree view

Screens Components

Search

New screen

App

Screen1

- Image4
- Group3
 - Step2
 - Step3
- Group1
 - Step1
- QuestionMark1
- PreviewAudio
- TextInput1
- Audio1
- Microphone1
- Gallery1

Screen2

Start

2. Preview Audio

00:00:00 00:00:00

3. Upload Audio File

Display mode Edit

Visible On

Position X 177 Y 475

Size Width 284 Height 82

Padding Top 5 Bottom 5 Left 5 Right 5

Color A

Border 2

Border radius 25

Font Open Sans

Font size 24

Font weight B Semibold

Font style / U

Text alignment

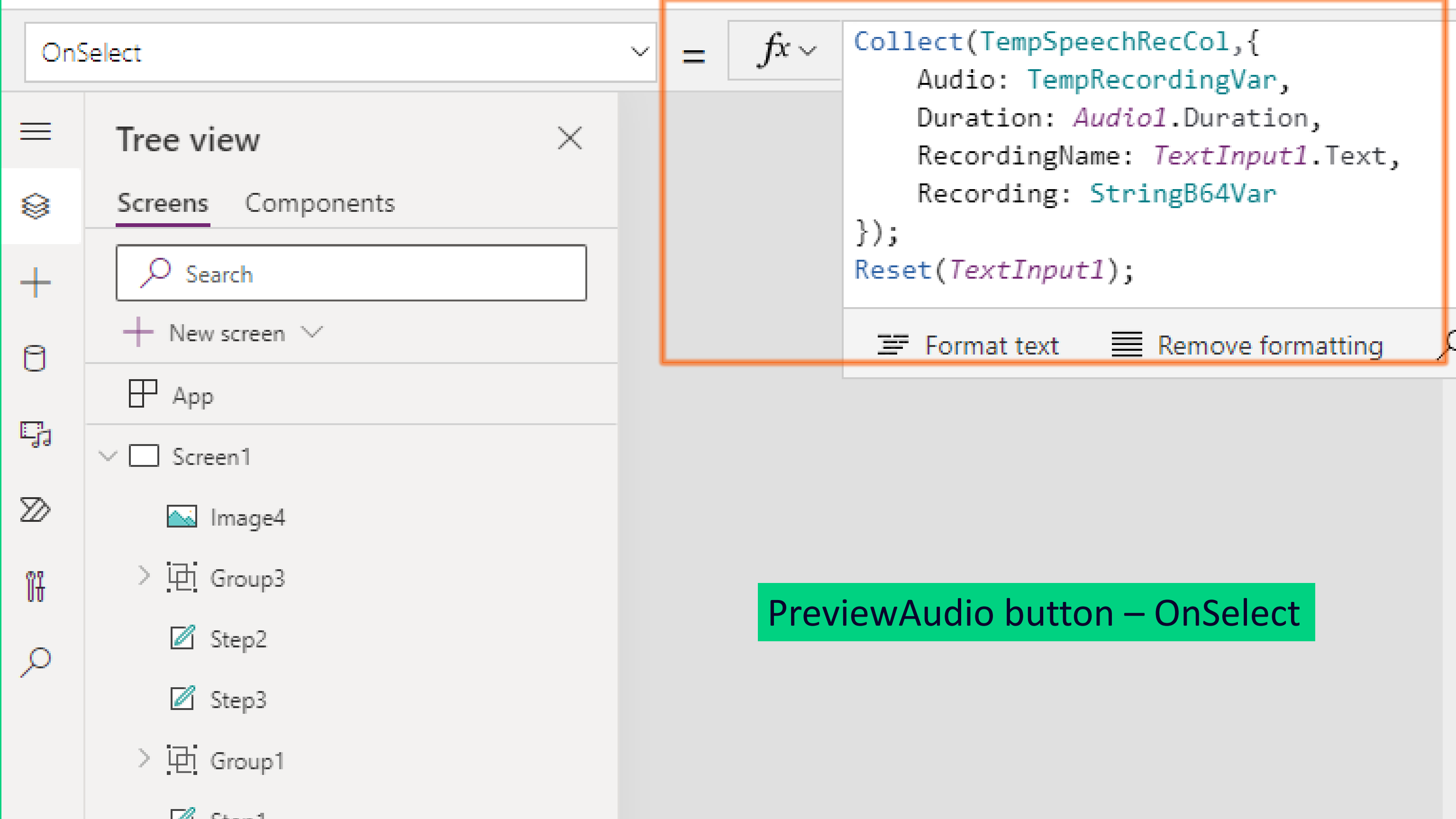
Vertical align Middle

Auto disable on select On

Disabled color

Pressed color

Screen1 PreviewAudio 60 %



OnSelect

=

fx

```
Collect(TempSpeechRecCol,{  
    Audio: TempRecordingVar,  
    Duration: Audio1.Duration,  
    RecordingName: TextInput1.Text,  
    Recording: StringB64Var  
});  
Reset(TextInput1);
```



Format text

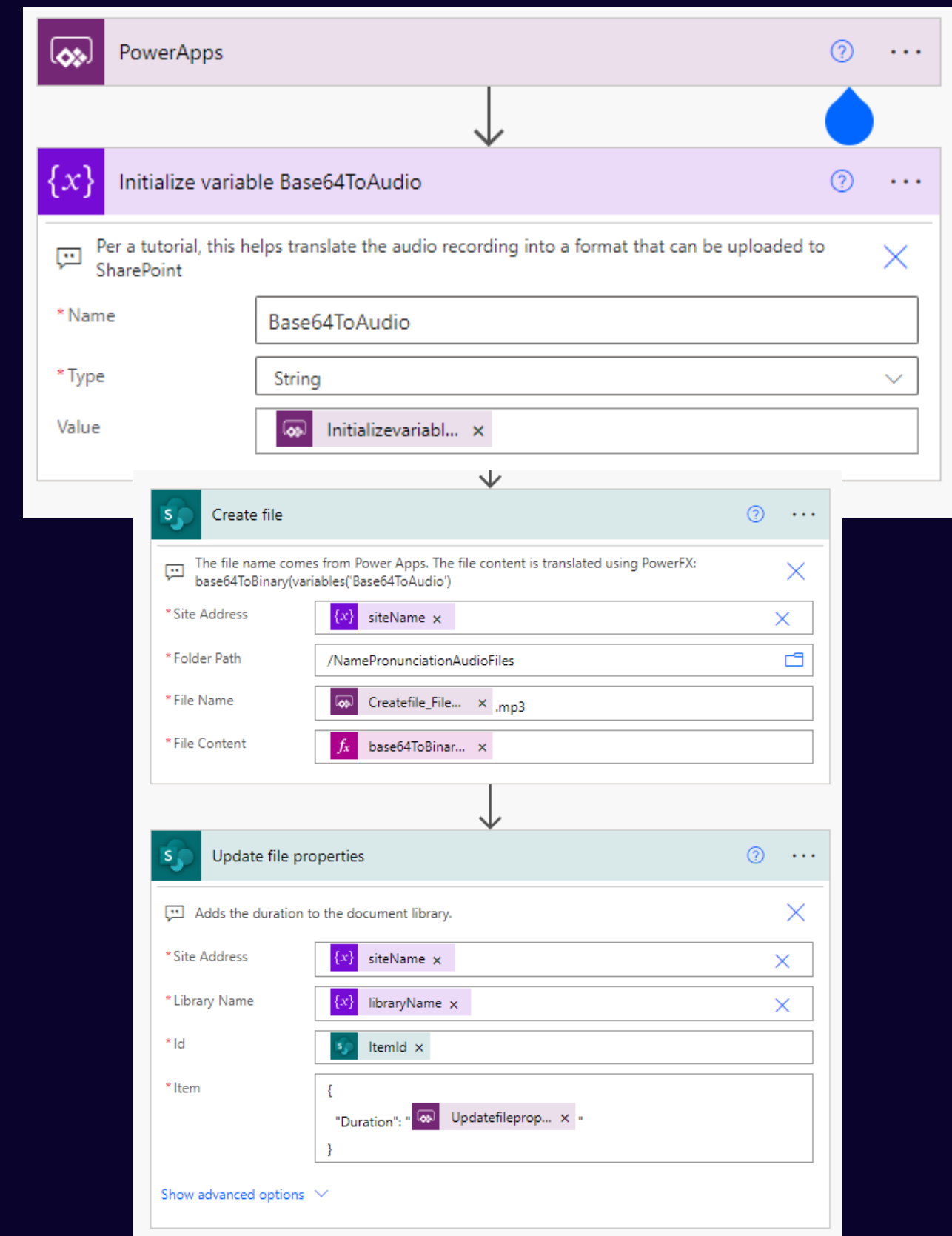


Remove formatting

PreviewAudio button – OnSelect

Power Automate

- Prior to making this, make a SharePoint doc library and add a number column called “Duration”
- Trigger is Power Apps – instant flow
- Convert that data from a string to an audio file in a variable using Power Apps data
- Create the file in SharePoint – file name is the recording name.mp3, and the file content is an expression – Base64toBinary(Base64ToAudio variable)
 - Note that this is different than Daniel’s tutorial – this is what I got working
- Update file properties to add in the Duration
- So, getting three things from Power Apps – the file itself, the file name, and the duration



Power Apps | GenericNamePronunciationAudioRecorder

Environment
Personal Productivity

LS

← Back ↶ ⌵ ⌵ + Insert ⌵ Open Sans 24 Semibold ⌵ ≡ ⌵ Color ⌵ Background color ⌵ Settings ⋮

OnSelect ⌵ = fx ⌵ UpdateContext({pressedbutton: !pressedbutton});
ForAll(TempSpeechRecCol,GenericSaveAudioFiletoSP0.Run(Recording, RecordingNameGal.Text, DurationGal.Text));

Format text Remove formatting Find and replace

Tree view

Screens Components

Search

+ New screen ⌵

App

Screen1

Image4

Group3

Button2

Image1

HtmlText1

Step2

Step3

Group1

Step1

QuestionMark1

PreviewAudio

TextInput1

Audio1

Microphone1

Gallery1

Screen2

Start

2. Preview Audio ?

00:00:00 00:00:00 🔊 🗑️

3. Upload Audio File

📶

🦊

Display mode Edit ⌵

Visible On ☒

Position 261 844
X Y

Size 100 100
Width Height

Padding 5 5
Top Bottom
5 5
Left Right

Color A 🎨

Border — 2 🎨

Border radius 25

Font Open Sans ⌵

Font size 24

Font weight B Semibold ⌵

Font style / U ➡

Text alignment ⌵ ⌵ ⌵ ⌵

Vertical align Middle ⌵

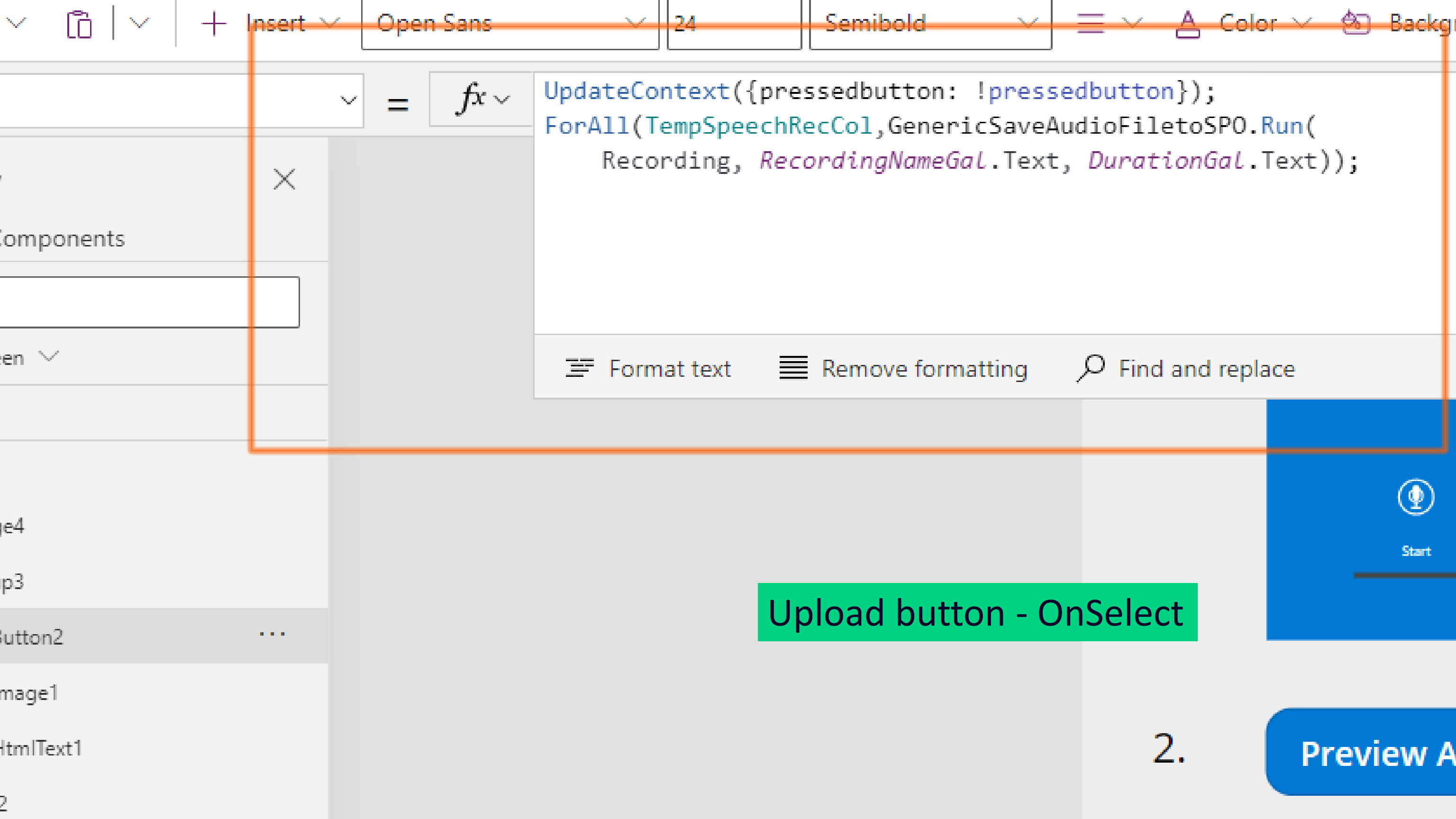
Auto disable on select Off ☐

Disabled color A 🎨

Pressed color A 🎨

Screen1 ⌵ Button2

60 % ↗



Upload button - OnSelect

2.

Preview A

Thank You!

This whole project wouldn't be possible without the amazing Youtube guidance from Daniel Christian – Save Audio Recording to SharePoint Online <https://www.youtube.com/@DanielChristian19> !

For help making it at least a little bit pretty - Kristine Kolodziejski – Neumorphic Buttons

<https://www.kristinekolodziejski.com/blog/building-beautiful-buttons-in-canvas-apps-part-1-neumorphic-buttons>