This Streamlit app provides functionalities to convert a given text input into audio, an image with the text, and a downloadable text file. Here's an explanation of the code:

Code Explanation

Imports

```
from PIL import Image, ImageDraw, ImageFont
import streamlit as st
from gtts import gTTS
from io import BytesIO
import base64
```

- **PIL (Python Imaging Library): Used for creating and manipulating images.**
- **Streamlit: A framework for creating web apps.**
- **gTTS (Google Text-to-Speech): Converts text to speech.**
- **BytesIO: A module from `io` library for in-memory binary streams.**
- **base64: Encodes binary data into ASCII text.**

```
st.title("Text to Audio, Image, and Text Converter")
```

Sets the title of the Streamlit app.

Text Input

```
text_input = st.text_area("Enter Text:").
```

Creates a text area where users can input their text.

Language Selection

```
language = st.selectbox("Select Language:", ["en", "fr", "de", "es", "it"])
```

Creates a dropdown to select the language for text-to-speech conversion. Default options include English, French, German, Spanish, and Italian.

Function to Convert Text to Audio

```
def text_to_speech(text, lang='en'):
    tts = gTTS(text=text, lang=lang)
    fp = BytesIO()
    tts.write_to_fp(fp)
    return fp
```

- **text_to_speech: Converts text to speech using the Google Text-to-Speech API.**
- **`gTTS(text=text, lang=lang)`: Creates a gTTS object with the specified text and language.**
- **`fp = BytesIO()`: Creates an in-memory bytes buffer.**
- **`tts.write_to_fp(fp)`: Writes the audio to the buffer.**
- **`return fp`: Returns the audio buffer.**

Function to Create Image with Text

```
def create_image_with_text(text):
    img = Image.new('RGB', (800, 600), color='white')
    draw = ImageDraw.Draw(img)
    font_path = "/home/shubhamchoudharyjr/myenv/Projects/Final Project/Roboto-Light.ttf"
    font_size = 36
    font = ImageFont.truetype(font_path, size=font_size)
    text_width, text_height = draw.textsize(text, font=font)
    text_x = (img.width - text_width) // 2
    text_y = (img.height - text_height) // 2
    draw.text((text_x, text_y), text, fill='black', font=font)
    return img
```

- **create_image_with_text: Creates an image with the given text.**
- **`img = Image.new('RGB', (800, 600), color='white')`: Creates a new white image of size 800x600.**
- **`draw = ImageDraw.Draw(img)`: Prepares the image for drawing.**

- `font_path`: Path to the font file.
- `font_size`: Size of the font.
- `font = ImageFont.truetype(font_path, size=font_size)`: Loads the font.
- `text_width, text_height = draw.textsize(text, font=font)`: Calculates the size of the text.
- `text_x = (img.width - text_width) // 2`: Calculates the horizontal position to center the text.
- `text_y = (img.height - text_height) // 2`: Calculates the vertical position to center the text.
- `draw.text((text_x, text_y), text, fill='black', font=font)`: Draws the text on the image.
- `return img`: Returns the image with the text.

Function to Download Image as PNG

```
def download_image(img):
    buffered = BytesIO()
    img.save(buffered, format="PNG")
    img_str = base64.b64encode(buffered.getvalue()).decode()
    href = f'<a href="{img_str}"
download="image_output.png">Download Image</a>'
    return href
```

- download_image: Creates a downloadable link for the image.
- `buffered = BytesIO()`: Creates an in-memory bytes buffer.
- `img.save(buffered, format="PNG")`: Saves the image to the buffer in PNG format.
- `img_str = base64.b64encode(buffered.getvalue()).decode()`: Encodes the image to base64.
- `href = f'<a href="{img_str}" download="image_output.png">Download Image</a>'`: Creates an HTML link for downloading the image.
- `return href`: Returns the HTML link.

Convert Button for Audio

```
if st.button("Convert to Audio"):
    if text_input:
        audio_file = text_to_speech(text_input, language)
        st.audio(audio_file)
```

- **Checks if the "Convert to Audio" button is clicked.**
- **If there is text input, converts the text to audio and plays it.**

Convert Button for Image

```
if st.button("Convert to Image"):
    if text_input:
        image = create_image_with_text(text_input)
        st.image(image)
```

- **Checks if the "Convert to Image" button is clicked.**
- **If there is text input, converts the text to an image and displays it.**

Download Button for Image

```
if st.button("Download Image"):
    if text_input:
        image = create_image_with_text(text_input)
        st.markdown(download_image(image),
unsafe_allow_html=True)
```

- **Checks if the "Download Image" button is clicked.**
- **If there is text input, converts the text to an image and provides a download link for the image.**

This code creates an interactive Streamlit app that converts user-provided text into audio, an image, and provides a downloadable link for the image. Make sure to replace `font_path` with the actual path to your font file.