```
from Shikimori import app,bot_token
from pyrogram import filters
import requests
from urllib.parse import quote_plus
from bs4 import BeautifulSoup
from unidecode import unidecode
from pyrogram.types import InlineKeyboardButton, InlineKeyboardMarkup

# Credit @the_only_god/@Yeah_Am_Kakashi
# -- Requirements --
# pyrogram
# Unidecode~=1.3.6
# requests
# beautifulsoup4


async def Sauce(bot_token,file_id):
    r = requests.post(f'https://api.telegram.org/bot{bot_token}/getFile?file_id={file_id}').json()
    file_path = r['result']['file_path']
    headers = {'User-agent': 'Mozilla/5.0 (Linux; Android 6.0.1; SM-G920V Build/MMB29K)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.98 Mobile Safari/537.36'}
    to_parse = f"https://images.google.com/searchbyimage??
safe=off&sbisrc=tg&image_url=https://api.telegram.org/file/bot{bot_token}/{file_path}"
    r = requests.get(to_parse,headers=headers)
    soup = BeautifulSoup(r.text, 'html.parser')
    result = {
             "similar": '',
             'output': ''
         }
    for similar_image in soup.find_all('input', {'class': 'gLFyf'}):
         url = f"https://www.google.com/search?tbm=isch&q={quote_plus(similar_image.get('value'))}"
         result['similar'] = url
    for best in soup.find_all('div', {'class': 'r5a77d'}):
        output = best.get_text()
        decoded_text =  unidecode(output)
        result["output"] = decoded_text

    return result

async def get_file_id_from_message(message):
    file_id = None
    message = message.reply_to_message
    if not message:
        return None
    if message.document:
        if int(message.document.file_size) > 3145728:
            return
        mime_type = message.document.mime_type
        if mime_type not in ("image/png", "image/jpeg"):
            return
        file_id = message.document.file_id
    if message.sticker:
        if message.sticker.is_animated:
            if not message.sticker.thumbs:
                return
            file_id = message.sticker.thumbs[0].file_id
        else:
            file_id = message.sticker.file_id

    if message.photo:
        file_id = message.photo.file_id

    if message.animation:
        if not message.animation.thumbs:
            return
        file_id = message.animation.thumbs[0].file_id

    if message.video:
        if not message.video.thumbs:
            return
        file_id = message.video.thumbs[0].file_id
    return file_id


@app.on_message(filters.command(["pp","grs","reverse","p"]))
async def _reverse(_,msg):
    text = await msg.reply("** ⇢ wait a sec...**")
    file_id = await get_file_id_from_message(msg)
    if not file_id:
        return await text.edit("**reply to media!**")
    await text.edit("** ⇢ Requesting to Google....**")
    result = await Sauce(bot_token,file_id)
    await text.edit(f'[{result["output"]}]({result["similar"]})\n\n⇢**Credits** -
@The_Only_God',reply_markup=InlineKeyboardMarkup([[InlineKeyboardButton("Open
Link",url=result["similar"])]]))
```