

```
import random

import streamlit as st

from textblob import TextBlob


# Extended Fashion Knowledge Base

COLOR_COMBINATIONS = {

    "black": ["white", "gold", "red", "grey", "silver"],

    "white": ["blue", "black", "pastel shades", "olive green"],

    "beige": ["pastel pink", "brown", "rust"],

    "navy blue": ["mustard yellow", "white", "grey"],

    "lavender": ["mint green", "grey", "white"],

    "red": ["denim blue", "black", "cream"],

    "olive green": ["cream", "rust", "white"],

    "grey": ["black", "red", "navy blue"],

    "brown": ["beige", "cream", "mustard"],

    "pastel pink": ["white", "beige", "light grey"]

}


FASHION_TRENDS = [

    "Y2K aesthetic",

    "Oversized blazers",

    "Baggy jeans & crop tops",

    "Statement belts",

    "Chunky sneakers",

    "Matching coords",

    "Minimalist neutral tones",

    "Sporty streetwear",

    "Utility fashion with pockets",

    "Denim on denim"

]
```

```
OOTD_SUGGESTIONS = [  
    "High-waist jeans + relaxed tee + sneakers",  
    "Oversized shirt + straight pants + crossbody bag",  
    "Flared trousers + plain tee + boots",  
    "Denim jacket + cargo pants + basic tee",  
    "Co-ord set + minimal accessories",  
    "Satin shirt + wide-leg jeans + heels",  
    "Crop top + midi skirt + sandals"  
]
```

```
STYLE_TIPS = [  
    "Add a belt to instantly elevate a simple dress.",  
    "Layering is key for a stylish and dynamic outfit.",  
    "Mix textures like denim, leather, and cotton for contrast.",  
    "Always balance oversized and fitted pieces.",  
    "Accessories can change the entire vibe of your outfit."  
]
```

# SlayBot with Improved Logic

```
class SlayBot:  
    def __init__(self):  
        pass  
  
    def extract_color(self, prompt):  
        blob = TextBlob(prompt)  
        for word in blob.words:  
            word_lower = word.lower()  
            if word_lower in COLOR_COMBINATIONS:  
                return word_lower  
        return None
```

```

def respond(self, prompt):

    prompt = prompt.lower()

    if "color" in prompt or "goes with" in prompt or "match with" in prompt:

        color = self.extract_color(prompt)

        if color:

            options = COLOR_COMBINATIONS[color]

            return f"{color.title()} pairs well with {' '.join([c.title() for c in options])}. Try mixing and matching!"

        else:

            return "I couldn't detect the base color. Try rephrasing your question."

    elif "trend" in prompt or "trending" in prompt or "what's in" in prompt:

        return f"Hot trend alert! {random.choice(FASHION_TRENDS)} is super in right now."

    elif "ootd" in prompt or "outfit" in prompt or "wear" in prompt:

        return f"Here's an outfit idea for you: {random.choice(OOTD_SUGGESTIONS)}"

    elif "tip" in prompt or "advice" in prompt or "style me" in prompt:

        return f"Style tip: {random.choice(STYLE_TIPS)}"

    elif "help" in prompt or "what can you do" in prompt:

        return ("SlayBot can help you with:"

                "- Color combinations (e.g., 'What goes with red?')\n"

                "- Latest fashion trends (e.g., 'What's trending?')\n"

                "- Outfit of the day ideas (e.g., 'Give me an OOTD')\n"

                "- Styling tips (e.g., 'Give me a fashion tip')")

    else:

        return "Hmm... I'm not sure I understood that. Try asking about colors, trends, outfits, or styling tips."

```

```
# Streamlit UI
```

```
bot = SlayBot()
```

```
st.set_page_config(page_title="SlayBot - AI Fashion Assistant", page_icon="👗")
```

```
st.title("👗 SlayBot - Your AI Fashion Assistant")
```

```
st.markdown("_Talk to your AI stylist! Ask about colors, trends, outfits or tips._")
```

```
if "history" not in st.session_state:
```

```
    st.session_state.history = []
```

```
user_input = st.chat_input("Ask me something fashionable...")
```

```
if user_input:
```

```
    response = bot.respond(user_input)
```

```
    st.session_state.history.append((user_input, response))
```

```
for q, a in st.session_state.history:
```

```
    with st.chat_message("user"):
```

```
        st.write(q)
```

```
    with st.chat_message("assistant"):
```

```
        st.write(a)
```