```
!pip install streamlit pyngrok --quiet
```

```
!pip install SpeechRecognition pydub
```

```
Collecting SpeechRecognition
  Using cached speechrecognition-3.14.3-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from SpeechRecognition) (4
Using cached speechrecognition-3.14.3-py3-none-any.whl (32.9 MB)
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.14.3
```

```
pip install googletrans==4.0.0-rc1
```

```
Requirement already satisfied: googletrans==4.0.0-rc1 in /usr/local/lib/python3.11/dist-packages (4.0.0rc1)
Requirement already satisfied: httpx==0.13.3 in /usr/local/lib/python3.11/dist-packages (from googletrans==4.0.0-rc1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletrans==4.0
Requirement already satisfied: hstspreload in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletrans=
Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletrans==4.0
Requirement already satisfied: chardet==3.* in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletrans
Requirement already satisfied: idna==2.* in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletrans==4
Requirement already satisfied: rfc3986<2,>=1.3 in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletr
Requirement already satisfied: httpcore==0.9.* in /usr/local/lib/python3.11/dist-packages (from httpx==0.13.3->googletr
Requirement already satisfied: h11<0.10,>=0.8 in /usr/local/lib/python3.11/dist-packages (from httpcore==0.9.*->httpx==
Requirement already satisfied: h2==3.* in /usr/local/lib/python3.11/dist-packages (from httpcore==0.9.*->httpx==0.13.3
Requirement already satisfied: hyperframe<6,>=5.2.0 in /usr/local/lib/python3.11/dist-packages (from h2==3.*->httpcore=
Requirement already satisfied: hpack<4,>=3.0 in /usr/local/lib/python3.11/dist-packages (from h2==3.*->httpcore==0.9.*
```

```
%%writefile app.py
import streamlit as st
import pandas as pd
import re
from googletrans import Translator

# Sample product catalog
products = pd.DataFrame([
    {"Product": "Red T-shirt", "Color": "Red", "Available": 12, "Mfg Date": "2024-01-10", "Price": 25},
    {"Product": "Blue Jeans", "Color": "Blue", "Available": 8, "Mfg Date": "2023-11-05", "Price": 40},
    {"Product": "Green Jacket", "Color": "Green", "Available": 5, "Mfg Date": "2024-02-20", "Price": 75},
    {"Product": "Black Shoes", "Color": "Black", "Available": 10, "Mfg Date": "2023-12-01", "Price": 90},
    {"Product": "White Shirt", "Color": "White", "Available": 15, "Mfg Date": "2024-03-15", "Price": 35},
    {"Product": "Yellow Hat", "Color": "Yellow", "Available": 7, "Mfg Date": "2024-04-05", "Price": 20},
    {"Product": "Brown Belt", "Color": "Brown", "Available": 9, "Mfg Date": "2023-10-10", "Price": 30},
    {"Product": "Orange Scarf", "Color": "Orange", "Available": 4, "Mfg Date": "2024-01-25", "Price": 18},
    {"Product": "Purple Socks", "Color": "Purple", "Available": 20, "Mfg Date": "2023-09-12", "Price": 10},
    {"Product": "Grey Sweater", "Color": "Grey", "Available": 6, "Mfg Date": "2024-05-01", "Price": 60},
])

translator = Translator()

st.title("🛍️ ABC Store - Real-Time Product Catalogue")

query_tamil = st.text_input("உங்கள் வினாவை உள்ளிடுங்கள் (e.g., 'Rs.50 க்குக் கீழ் சிவப்பு பொருட்கள்')")

def translate_query(tamil_text):
    try:
        translated = translator.translate(tamil_text, src='ta', dest='en')
        return translated.text
    except:
        return tamil_text  # fallback if translation fails

def filter_products(query):
```

```
        df = products.copy()
        if not query:
            return df

        query_lower = query.lower()

        # Filter by color
        for color in df["Color"].unique():
            if color.lower() in query_lower:
                df = df[df["Color"].str.lower() == color.lower()]

        # Filter by price
        price_match = re.search(r'under\s*\Rs.?(\d+)', query_lower)
        if price_match:
            price_limit = float(price_match.group(1))
            df = df[df["Price"] < price_limit]

        return df

    if query_tamil:
        translated_query = translate_query(query_tamil)
        st.write(f"🔎 Tamil Query: `{query_tamil}`")
        st.write(f"🌐 Translated to English: `{translated_query}`")

        result = filter_products(translated_query)

        if not result.empty:
            st.subheader("📦 Matching Products")
            st.dataframe(result)
        else:
            st.warning("No matching products found.")
```

```
Overwriting app.py
```

```
from pyngrok import ngrok
import os

# Kill previous tunnels
ngrok.kill()

# Start streamlit app in background
!streamlit run app.py &> logs.txt &

# Create public URL
public_url = ngrok.connect("http://localhost:8501")
print(f"🌐 Your app is live at: {public_url}")
```

```
🌐 Your app is live at: NgrokTunnel: "https://f265-35-243-130-222.ngrok-free.app" -> "http://localhost:8501"
```

Start coding or generate with AI.