```python
import streamlit as st
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
import joblib
import os

# Training the model (runs only once)
def train_chatbot():
    data = {
        'query': [
            "How can I reset my password?",
            "I want to return my order.",
            "Where is my order?",
            "How do I track my order?",
            "I received a damaged product.",
            "I want to cancel my order.",
            "Can I speak to a human?",
            "Tell me about your return policy.",
            "I need help with my account.",
            "Do you ship internationally?"
        ],
        'intent': [
            "account_help",
            "returns",
            "order_status",
            "order_status",
            "complaint",
            "cancel_order",
            "human_support",
            "returns",
            "account_help",
            "shipping"
        ],
        'response': [
            "You can reset your password from the login page.",
            "Sure, you can return your order within 10 days.",
            "Please check your order status on your profile page.",
            "Track your order here: [Order Tracking Page]",
            "Sorry about that! You can request a replacement or refund.",
            "Your order can be canceled within 24 hours of placing it.",
            "Connecting you to a human agent...",
            "You can return most items within 10 days of delivery.",
            "I can help with your account issues. What do you need?",
            "Yes, we do ship internationally to selected countries."
        ]
    }
    df = pd.DataFrame(data)
    model = Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('clf', LogisticRegression())
    ])
    model.fit(df['query'], df['intent'])
    response_map = df.groupby('intent')['response'].first().to_dict()
    joblib.dump(model, 'chatbot_model.pkl')
    joblib.dump(response_map, 'response_map.pkl')

# Train if not already trained
if not os.path.exists('chatbot_model.pkl') or not os.path.exists('response_map.pkl'):
    train_chatbot()

# Load model and responses
model = joblib.load('chatbot_model.pkl')
responses = joblib.load('response_map.pkl')

# Streamlit UI
st.title("Revolutionizing Customer Support with an Intelligent Chatbot")
st.write("Ask me anything related to orders, returns, shipping, or your account.")
user_input = st.text_input("You:", "")
if user_input:
    intent = model.predict([user_input])[0]
    response = responses.get(intent, "I'm not sure how to help with that yet.")
    st.write(f"Bot: {response}")
```