```
import streamlit as st
import requests
import threading
import time
import main
from PIL import Image
import pytesseract
import io
import PyPDF2
import os

# --- Start FastAPI in Background ---
def start_fastapi():
    thread = threading.Thread(target=main.run_fastapi, daemon=True)
    thread.start()
    time.sleep(5)  # Wait for FastAPI to start

start_fastapi()

# --- Login System ---
st.sidebar.title("🔐 Login")
username = st.sidebar.text_input("Username")
password = st.sidebar.text_input("Password", type="password")

if username != "doctor" or password != "password123":
    st.sidebar.info("Enter credentials: doctor / password123")
    st.stop()
else:
    st.sidebar.success("Logged in!")

# --- History ---
if "history" not in st.session_state:
    st.session_state.history = []

# --- Page Config ---
st.set_page_config(page_title="🩺 MedVerify AI", page_icon="🩺")
st.title("🩺 AI Medical Prescription Verifier")
st.markdown("Upload or paste a prescription to check safety, dosage, and interactions.")

# --- Tabs ---
tab1, tab2, tab3 = st.tabs(["📝 Text Input", "🖼️ Image Upload", "📄 PDF Upload"])

prescription_text = ""
age = st.number_input("Patient Age", min_value=0, max_value=120, value=30)
```

```
# --- Text Input ---
with tab1:
    prescription_text = st.text_area("Enter Prescription Text", height=150)

# --- Image OCR ---
with tab2:
    uploaded_file = st.file_uploader("Upload prescription image", type=["png", "jpg", "jpeg"])
    if uploaded_file:
        image = Image.open(uploaded_file)
        st.image(image, caption="Uploaded", use_column_width=True)
        try:
            text = pytesseract.image_to_string(image)
            st.text_area("Extracted Text", value=text, height=150)
            prescription_text = text
        except Exception as e:
            st.error(f"OCR Error: {str(e)}")

# --- PDF Parsing ---
with tab3:
    pdf_file = st.file_uploader("Upload PDF", type=["pdf"])
    if pdf_file:
        try:
            pdf_reader = PyPDF2.PdfReader(pdf_file)
            text = ""
            for page in pdf_reader.pages:
                text += page.extract_text()
            st.text_area("Extracted PDF Text", value=text, height=150)
            prescription_text = text
        except Exception as e:
            st.error(f"PDF Error: {str(e)}")

# --- Verify Button ---
if st.button("🔍 Verify Prescription"):
    if not prescription_text.strip():
        st.error("Please provide prescription text, image, or PDF.")
    else:
        with st.spinner("Analyzing with AI..."):
            try:
                response = requests.post(
                    "http://localhost:8000/verify",
                    json={"prescription_text": prescription_text, "patient_age": age}
                )
                result = response.json()
```

```
            # Save to history
            st.session_state.history.append({
                "text": prescription_text[:100] + "...",
                "age": age,
                "drugs": ", ".join(result.get("extracted_drugs", []))
            })

            # Display Results
            st.success("✅ Analysis Complete")

            st.subheader("💊 Extracted Drugs")
            if result["extracted_drugs"]:
                st.write(", ".join(result["extracted_drugs"]))
            else:
                st.write("None detected")

            if result.get("interactions"):
                st.error("⚠️ Potential Interactions")
                for inter in result["interactions"]:
                    st.markdown(f"**{inter['drugs']}**: {inter['description']}")

            st.subheader("📏 Recommended Dosages")
            for drug, dose in result["recommended_dosages"].items():
                st.write(f"**{drug}**: {dose}")

            if result.get("alternatives"):
                st.subheader("🔄 Alternative Medications")
                for drug, alts in result["alternatives"].items():
                    st.write(f"**{drug}**: {', '.join(alts)}")

        except Exception as e:
            st.error(f"❌ Request failed: {str(e)}")
            st.info("Make sure FastAPI is running and API key is set.")

# --- History Sidebar ---
st.sidebar.markdown("---")
st.sidebar.header("📋 Recent Checks")
for i, item in enumerate(st.session_state.history[-5:], 1):
    with st.sidebar.expander(f"Entry {i}"):
        st.write(f"Age: {item['age']}")
        st.write(f"Drugs: {item['drugs']}")

# --- Footer ---
```

```
st.markdown("---")
st.caption("Powered by FastAPI, Streamlit, Hugging Face, DrugBank API & Replit")
```