import streamlit as st

import time

import numpy as np

import pandas as pd

import matplotlib

matplotlib.use('Agg') # Use 'Agg' backend for rendering in Streamlit

import matplotlib.pyplot as plt

def check\_for\_fraud(transaction\_id, amount, transaction\_type):

time.sleep(2)

if amount > 1000 and transaction\_type == "Withdrawal":

return "Suspicious Activity Detected!"

elif amount < 0:

return "Invalid Transaction"

else:

return "Transaction is Legitimate."

def generate\_transaction\_history(num\_transactions):

transaction\_types = ["Purchase", "Withdrawal", "Transfer"]

transaction\_data = {

"Transaction ID": np.random.randint(1000, 9999, size=num\_transactions),

"Amount ($)": np.random.uniform(10, 5000, size=num\_transactions).round(2),

"Type": np.random.choice(transaction\_types, size=num\_transactions),

"Status": np.random.choice(["Legitimate", "Suspicious"], size=num\_transactions, p=[0.9, 0.1])

}

return pd.DataFrame(transaction\_data)

def main():

st.set\_page\_config(page\_title="Fraud Detection System", layout="wide")

st.title("🚨 Fraud Detection System for Online Transactions")

st.sidebar.title("Navigation")

menu = st.sidebar.selectbox("Menu", ["Fraud Detection", "Transaction History", "About"])

if menu == "Fraud Detection":

st.subheader("Transaction Fraud Checker")

with st.form(key="transaction\_form"):

transaction\_id = st.text\_input("Transaction ID", max\_chars=10, help="Enter the transaction ID (max 10 characters)")

amount = st.number\_input("Transaction Amount", min\_value=1.0, help="Enter the transaction amount")

transaction\_type = st.selectbox("Transaction Type", ["Purchase", "Withdrawal", "Transfer"], help="Select the type of transaction")

submit\_button = st.form\_submit\_button(label="Check for Fraud")

if submit\_button:

if transaction\_id == "":

st.error("Transaction ID is required.")

elif amount <= 0:

st.error("Amount must be greater than 0.")

else:

with st.spinner("🔍 Checking for fraud..."):

result = check\_for\_fraud(transaction\_id, amount, transaction\_type)

st.success(f"Transaction Result: {result}")

elif menu == "Transaction History":

st.subheader("📊 Transaction History Overview")

num\_transactions = st.slider("Select Number of Transactions to Generate", min\_value=10, max\_value=100, value=50)

transaction\_data = generate\_transaction\_history(num\_transactions)

st.dataframe(transaction\_data, height=400)

st.markdown("### Transactions by Type")

type\_counts = transaction\_data["Type"].value\_counts()

fig, ax = plt.subplots()

ax.pie(type\_counts, labels=type\_counts.index, autopct="%1.1f%%", startangle=90, colors=["#007bff", "#28a745", "#ffc107"])

ax.axis("equal")

st.pyplot(fig)

st.markdown("### Suspicious Transactions")

suspicious\_transactions = transaction\_data[transaction\_data["Status"] == "Suspicious"]

if not suspicious\_transactions.empty:

st.dataframe(suspicious\_transactions)

else:

st.info("No suspicious transactions detected.")

elif menu == "About":

st.subheader("ℹ️ About this Application")

st.markdown("""

\*\*Fraud Detection System\*\* is a web-based application designed to identify potentially fraudulent transactions.

This demo allows users to input transaction details and receive a fraud risk assessment, with an additional view of simulated transaction history.

- \*\*Technologies Used\*\*: Python, Streamlit, Matplotlib, Pandas

- \*\*Detection Logic\*\*: Basic heuristics (withdrawals > $1000 marked as suspicious)

""")

st.sidebar.markdown(f"""

---

Developed by Sharath (Registration No: 22BCE1802)

""")

if \_\_name\_\_ == "\_\_main\_\_":

main()