```
import streamlit as st
import numpy as np
import joblib

# Load trained ML model
model = joblib.load("accident_severity_model.pkl")

# Title and subtitle
st.set_page_config(page_title="Traffic Accident Severity Predictor", layout="centered")
st.title("🚦 Road Safety AI: Accident Severity Predictor")
st.markdown("Predict the severity of a traffic accident based on environmental and road conditions using a machine learning model.")

# Sidebar - Input Features
st.sidebar.header("Enter Traffic Conditions")

time = st.sidebar.selectbox("Time of Day", ["Morning", "Afternoon", "Evening", "Night"])
weather = st.sidebar.selectbox("Weather Condition", ["Clear", "Rain", "Snow", "Fog", "Windy"])
road = st.sidebar.selectbox("Road Condition", ["Dry", "Wet", "Icy", "Slippery", "Gravel"])
location = st.sidebar.selectbox("Location Type", ["Urban", "Rural"])

# Encoding maps
time_map = {"Morning": 0, "Afternoon": 1, "Evening": 2, "Night": 3}
weather_map = {"Clear": 0, "Rain": 1, "Snow": 2, "Fog": 3, "Windy": 4}
road_map = {"Dry": 0, "Wet": 1, "Icy": 2, "Slippery": 3, "Gravel": 4}
location_map = {"Urban": 0, "Rural": 1}
```

```
# Prepare input
input_data = np.array([[time_map[time], weather_map[weather], road_map[road], location_map[location]]])

# Predict Button
if st.button("🚨 Predict Accident Severity"):
    prediction = model.predict(input_data)[0]
    probability = model.predict_proba(input_data).max()  # Confidence of prediction

    st.subheader("📊 Prediction Result")
    severity_label = {
        0: "Low Severity",
        1: "Moderate Severity",
        2: "High Severity"
    }

    st.success(f"**Predicted Severity:** {severity_label.get(prediction, 'Unknown')} ({prediction})")
    st.info(f"🔎 Model Confidence: {probability * 100:.2f}%")

    # Explanation Tip
    st.markdown("---")
    st.caption("Note: Predictions are based on historical accident trends. Use with caution for real-world decisions.")
```