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

# -------------------------------------------------------
# CUSTOM PAGE STYLE
# -------------------------------------------------------
page_bg = """
<style>
.stApp {
    background-color: #0A0F24;
    color: #FFFFFF;
    font-family: 'Arial';
}
h1, h2, h3 {
    color: #14C7FF;
    font-weight: 700;
}
label {
    color: #D2E9FF !important;
}
.stNumberInput > div > div > input {
    color: #000000 !important;
    font-weight: 600;
}
.stButton>button {
    background-color: #14C7FF;
    color: black;
    font-size: 18px;
    font-weight: bold;
    border-radius: 10px;
    border: 2px solid #0EA5E9;
}
.stButton>button:hover {
    background-color: #0EA5E9;
    color: white;
    border: 2px solid white;
}
[data-testid="stDataFrame"] {
    background-color: #11172F;
}
</style>
"""
st.markdown(page_bg, unsafe_allow_html=True)

# -------------------------------------------------------
# LOAD TRAINED MODEL
# -------------------------------------------------------
model = joblib.load("pipeline_xgb_monotonic.joblib")
```

```
st.title("📊 YouTube Ad Revenue Prediction App")
st.markdown("### Predict YouTube video ad revenue using the improved ML model.")

# ---------------------------------------------------------
# USER INPUT UI
# ---------------------------------------------------------
st.header("🎬 Enter Video Details")

views = st.number_input("Views", min_value=0, value=10000)
likes = st.number_input("Likes", min_value=0, value=500)
comments = st.number_input("Comments", min_value=0, value=100)

# Engagement rate BEFORE log-transform
engagement_rate = (likes + comments) / views if views > 0 else 0

watch_time_minutes = st.number_input("Watch Time (minutes)", min_value=0.0,
value=10000.0)
video_length_minutes = st.number_input("Video Length (minutes)", min_value=0.0,
value=10.0)
subscribers = st.number_input("Subscribers Count", min_value=0, value=100000)

# ---------------------------------------------------------
# STRUCTURE INPUT DATA FOR MODEL
# ---------------------------------------------------------
input_data = pd.DataFrame({
    "views": [views],
    "likes": [likes],
    "comments": [comments],
    "Engagement_Rate": [engagement_rate],
    "watch_time_minutes": [watch_time_minutes],
    "video_length_minutes": [video_length_minutes],
    "subscribers": [subscribers]
})

# ---------------------------------------------------------
# APPLY SAME LOG TRANSFORM USED IN TRAINING
# ---------------------------------------------------------
input_data["views"] = np.log1p(input_data["views"])
input_data["subscribers"] = np.log1p(input_data["subscribers"])
input_data["watch_time_minutes"] = np.log1p(input_data["watch_time_minutes"])

st.subheader("▧ Input Summary (After Log Transform)")
st.dataframe(input_data)

# ---------------------------------------------------------
# PREDICTION
# ---------------------------------------------------------
if st.button("Predict Revenue"):
    log_pred = model.predict(input_data)[0]
```

```
revenue = np.expm1(log_pred)

st.success(f"💰 Estimated Ad Revenue: **${revenue:.2f} USD**")
```