Student Name: Abinaya.A

Register Number: 422723104003

Institution: V. R. S College of Engineering and Technology

Department: Computer Science Engineering

Source code:

stock_prediction_app.py

```
import streamlit as st import pandas as pd import numpy as np import joblib from
sklearn.preprocessing import MinMaxScaler from tensorflow.keras.models import
load_model import matplotlib.pyplot as plt

st.set_page_config(page_title="Stock Price Predictor", layout="centered")

Title

st.title("Cracking the Market Code: Stock Price Predictor") st.markdown("Predict
future stock prices using AI and time series analysis.")

Upload data

uploaded_file = st.file_uploader("Upload Stock Price CSV", type=['csv'])

if uploaded_file is not None: df = pd.read_csv(uploaded_file) df['Date'] =
pd.to_datetime(df['Date']) df.set_index('Date', inplace=True)
st.subheader("Preview of Uploaded Data") st.write(df.tail())

model_type = st.selectbox("Select Prediction Model", ["XGBoost", "LSTM"])
```

```
if st.button("Predict Next Day Price"):

    if model_type == "XGBoost":
        try:
            model = joblib.load('xgb_stock_model.pkl')
            # Ensure the required features match your model
            features = ['lag_1', 'rolling_mean_7', 'RSI', 'MACD']

            for feature in features:
                if feature not in df.columns:
                    st.error(f"Missing feature in dataset: {feature}")
                    st.stop()

            X_input = df[features].iloc[-1:].values
            prediction = model.predict(X_input)[0]
            st.success(f"Predicted Next Day Close Price: {prediction:.2f}")
        except Exception as e:
            st.error(f"Model or data error: {e}")

    elif model_type == "LSTM":
        try:
            model = load_model('lstm_model.h5')
            scaler = MinMaxScaler()
            scaled_close = scaler.fit_transform(df[['Close']])

            seq_length = 60
            if len(scaled_close) < seq_length:
                st.error("Not enough data for LSTM prediction. Need at least 60
records.")
                st.stop()

            last_seq = scaled_close[-seq_length:]
            X_input = np.expand_dims(last_seq, axis=0)
            prediction = model.predict(X_input)
        predicted_price = scaler.inverse_transform(prediction)[0][0]
        st.success(f"Predicted Next Day Close Price: {predicted_price:.2f}")
    except Exception as e:
        st.error(f"LSTM prediction failed: {e}")

else: st.info("Please upload a CSV file with stock price data including 'Date' and
'Close' columns.")
```

Output:

```
        Predicted Next Day Close Price: 137.52
    Missing Features:
        Missing feature in dataset: RSI
    Insufficient Data for LSTM:
        Not enough data for LSTM prediction. Need at least 60 records.
    Model loading or prediction errors:
        Model or data error: [error message]
```