```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import yfinance as yf
from datetime import date, timedelta
ticker = input("Enter the stock ticker symbol: ")
today = date.today()
start_date = today - timedelta(days=365 * 5)
try:
    stock_data = yf.download(ticker, start=start_date.strftime('%Y-%m-
%d'), end=today.strftime('%Y-%m-%d'))
except Exception as e:
    print(f"Error downloading data for {ticker}: {e}")
    exit()
if stock_data.empty:
    print(f"No data available for {ticker}")
    exit()
X = stock_data['Close'][:-1].values.reshape(-1, 1)
y = stock_data['Close'].shift(-1)[:-1].values.reshape(-1, 1)
try:
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
except ValueError as e:
    print(e)
    exit()
model = LinearRegression()
model.fit(X_train, y_train)
train_score = model.score(X_train, y_train)
test_score = model.score(X_test, y_test)
print(f"Training R-squared: {train_score:.2f}")
print(f"Testing R-squared: {test_score:.2f}")
future_days = 5
future_dates = pd.date_range(start=today + timedelta(days=1),
periods=future_days)
future_X = stock_data['Close'][-future_days:].values.reshape(-1, 1)
future_predictions = model.predict(future_X)
print(f"Predicted stock prices for {ticker}:")
for i, date in enumerate(future_dates):
    print(f"{date.date()}: ${future_predictions[i][0]:.2f}")
```

```
Enter the stock ticker symbol:  AAPL

[**********************100%%**********************]  1 of 1 completed

Training R-squared: 1.00
Testing R-squared: 1.00
Predicted stock prices for AAPL:
2024-04-16: $168.38
2024-04-17: $169.60
```

```
2024-04-18: $167.71
2024-04-19: $174.95
2024-04-20: $176.46
```