# **COMPLETE GITHUB ACTIONS TRADING BOT SETUP**

## **Daily Email with BUY/SELL/WAIT Signals**

## **PART 1: Create Gmail App Password**

## **Step 1: Enable 2-Step Verification**

1. Go to:
2. <https://myaccount.google.com/security>
3. Find "2-Step Verification"
4. Click it and turn it ON (follow prompts)
5. Complete setup (add phone number, etc.)

## **Step 2: Generate App Password**

1. Go to:
2. <https://myaccount.google.com/apppasswords>
3. Sign in if prompted
4. Under "App name" field, type: GitHub Trading Bot
5. Click "Create" button
6. Google shows a 16-character password (like: abcd efgh ijkl mnop)
7. COPY IT (you can only see it once!)
8. Keep it somewhere safe for now

## **PART 2: Create GitHub Repository**

## **Step 3: Create New Repo**

1. Go to:
2. [https://github.com](https://github.com/)
3. Click the "+" icon (top right) → "New repository"
4. Repository name: TRADING-BOT (or any name)
5. Privacy: Choose Private (recommended)
6. Check "Add a README file"
7. Click "Create repository"

## **PART 3: Add Your Trading Code**

## **Step 4: Create signal.py File**

1. In your new repo, click "Add file" → "Create new file"
2. Filename: Type signal.py
3. Paste this code:

import yfinance as yf

import pandas as pd

import numpy as np

from datetime import date

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

import os

# YOUR STOCKS

STOCKS = ["SPY", "AMZN", "MSFT", "NVDA", "TSLA", "GOOGL"]

def get\_signal\_for\_stock(symbol):

"""Get trading signal for a single stock"""

END\_DATE = str(date.today())

START\_DATE = "2025-09-01"

try:

data = yf.download(symbol, start=START\_DATE, end=END\_DATE, progress=False)

if data.empty:

return symbol, "ERROR", 0.0

# Calculate indicators

data['Short\_MA'] = data['Close'].rolling(window=10).mean()

data['Long\_MA'] = data['Close'].rolling(window=30).mean()

# RSI

delta = data['Close'].diff()

gain = (delta.where(delta > 0, 0)).rolling(window=14).mean()

loss = (-delta.where(delta < 0, 0)).rolling(window=14).mean()

rs = gain / loss

data['RSI'] = 100 - (100 / (1 + rs))

latest = data.iloc[-1]

price = float(latest['Close'])

short\_ma = float(latest['Short\_MA'])

long\_ma = float(latest['Long\_MA'])

rsi = float(latest['RSI'])

# Determine signal

if pd.isna(short\_ma) or pd.isna(long\_ma):

return symbol, "WAIT", price

elif short\_ma > long\_ma and 50 < rsi < 65:

return symbol, "BUY", price

elif short\_ma < long\_ma:

return symbol, "SELL", price

else:

return symbol, "WAIT", price

except Exception as e:

return symbol, "ERROR", 0.0

def create\_beautiful\_email():

"""Create beautiful formatted email"""

results = []

for stock in STOCKS:

symbol, signal, price = get\_signal\_for\_stock(stock)

results.append({

'symbol': symbol,

'signal': signal,

'price': price

})

# Count signals

buy\_count = sum(1 for r in results if r['signal'] == 'BUY')

sell\_count = sum(1 for r in results if r['signal'] == 'SELL')

wait\_count = sum(1 for r in results if r['signal'] == 'WAIT')

# Create email

email\_body = f"""

╔══════════════════════════════════════════════════╗

║ 📊 DAILY TRADING SIGNALS 📊 ║

║ {date.today()} ║

╚══════════════════════════════════════════════════╝

📈 Summary: {buy\_count} BUY | {sell\_count} SELL | {wait\_count} WAIT

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

"""

# Add each stock

for r in results:

symbol = r['symbol']

signal = r['signal']

price = r['price']

# Signal emoji

if signal == 'BUY':

emoji = '🟢'

elif signal == 'SELL':

emoji = '🔴'

elif signal == 'WAIT':

emoji = '⚪'

else:

emoji = '⚠️'

# Format price

if price >= 1000:

price\_str = f"${price:,.2f}"

else:

price\_str = f"${price:.2f}"

email\_body += f"""

{emoji} {symbol:<6} → {signal:<4} | Price: {price\_str}

"""

email\_body += """

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

Strategy: 10/30 MA Crossover + RSI (50-65)

⚠️ MESSAGE FROM REINDOLF: This is no financial advice. Trade responsibly!

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

"""

return email\_body

def send\_email(message):

"""Send email"""

sender\_email = os.environ.get('EMAIL\_USER')

sender\_password = os.environ.get('EMAIL\_PASSWORD')

receiver\_email = os.environ.get('EMAIL\_USER')

if not sender\_email or not sender\_password:

print("⚠️ Email credentials not set")

return

msg = MIMEMultipart()

msg['From'] = sender\_email

msg['To'] = receiver\_email

msg['Subject'] = f'📊 Trading Signals - {date.today()}'

msg.attach(MIMEText(message, 'plain'))

try:

server = smtplib.SMTP('smtp.gmail.com', 587)

server.starttls()

server.login(sender\_email, sender\_password)

server.send\_message(msg)

server.quit()

print("✅ Email sent!")

except Exception as e:

print(f"❌ Email failed: {str(e)}")

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

email\_message = create\_beautiful\_email()

print(email\_message)

send\_email(email\_message)

1. Click "Commit new file" (green button at bottom)

## **Step 5: Create GitHub Action Workflow**

1. Click "Add file" → "Create new file"
2. Filename: Type exactly .github/workflows/daily-signal.yml
3. Paste this code:

name: Daily Trading Signal

on:

schedule:

- cron: '0 11 \* \* \*' # Runs at 11:00 AM UTC daily

workflow\_dispatch: # Allows manual trigger

jobs:

send-signal:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v3

- name: Set up Python

uses: actions/setup-python@v4

with:

python-version: '3.11'

- name: Install dependencies

run: |

pip install yfinance pandas numpy

- name: Run trading signal

env:

EMAIL\_USER: ${{ secrets.EMAIL\_USER }}

EMAIL\_PASSWORD: ${{ secrets.EMAIL\_PASSWORD }}

run: python signal.py

1. Click "Commit new file"

## **PART 4: Add Email Secrets**

## **Step 6: Add EMAIL\_USER Secret**

1. In your repo, click "Settings" tab (top right)
2. Left sidebar → "Secrets and variables" → "Actions"
3. Click "New repository secret" (green button)
4. Name: EMAIL\_USER
5. Value: Your Gmail address (e.g., yourname@gmail.com)
6. Click "Add secret"

## **Step 7: Add EMAIL\_PASSWORD Secret**

1. Click "New repository secret" again
2. Name: EMAIL\_PASSWORD
3. Value: Paste the 16-character App Password from Step 2
4. Click "Add secret"

You should now see 2 secrets:

* EMAIL\_USER
* EMAIL\_PASSWORD

## **PART 5: Test It Now**

## **Step 8: Manually Run the Workflow**

1. Click "Actions" tab (top of repo)
2. Left sidebar → Click "Daily Trading Signal"
3. Click "Run workflow" dropdown (right side)
4. Click green "Run workflow" button
5. Wait 30-60 seconds → Refresh page
6. Click on the workflow run to see status
7. Check your Gmail inbox!

## **PART 6: Verify Daily Automation**

## **Step 9: Confirm Schedule**

* Your workflow is now set to run automatically every day at 11:00 AM UTC
* Convert to your timezone:
  + UK (BST): 11 AM UTC = 12 PM (noon)
  + UK (GMT in winter): 11 AM UTC = 11 AM

## **Step 10: Change Time (Optional)**

To run at different time, edit .github/workflows/daily-signal.yml:

Examples:

text

- cron: '0 10 \* \* \*' # 10 AM UTC (11 AM BST)

- cron: '30 8 \* \* \*' # 8:30 AM UTC (9:30 AM BST)

- cron: '0 14 \* \* \*' # 2 PM UTC (3 PM BST)

## **CUSTOMIZATION OPTIONS**

## **Change Stocks:**

Edit line 10 in signal.py:

python

STOCKS = ["AAPL", "TSLA", "BTC-USD", "ETH-USD", "SPY"]

## **Change Strategy (Advanced):**

* Line 30-31: Moving averages (currently 10/30 days)
* Line 48-54: Signal logic (BUY/SELL/WAIT conditions)

## **TROUBLESHOOTING**

## **No Email Received?**

1. Check spam/junk folder
2. Verify secrets are correct (Settings → Secrets)
3. Check Actions logs for errors
4. Make sure 2-Step Verification is ON in Google

## **Wrong Time?**

* GitHub uses UTC time
* BST = UTC + 1 hour
* GMT = UTC (no difference)

## **Change Email Address?**

* Update EMAIL\_USER secret in Settings → Secrets

## **WHAT YOU GET DAILY**

Email Subject: Trading Signals - 2025-10-11

Email Body:

text

Daily Signals - 2025-10-11

SPY - WAIT

AMZN - BUY

MSFT - WAIT

NVDA - SELL

TSLA - BUY

GOOGL - WAIT

## **COST: $0 - COMPLETELY FREE!**

✅ GitHub Actions: Free (2,000 minutes/month)  
✅ Gmail: Free  
✅ No credit card needed  
✅ Runs in the cloud (your computer can be off)

You're all set! Check your email every day at 11 AM UTC for your signals!