import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks, butter, filtfilt

def load\_ecg\_signal(file\_path):

# Simulated function to load ECG signal from a file

# For demonstration purposes, let's generate a simple sinusoidal ECG signal

fs = 1000 # Sampling frequency (Hz)

t = np.arange(0, 10, 1/fs) # Time vector (10 seconds)

ecg\_signal = np.sin(2 \* np.pi \* 1 \* t) + 0.5 \* np.sin(2 \* np.pi \* 20 \* t) # Simulated ECG signal

return t, ecg\_signal

def filter\_ecg\_signal(ecg\_signal, fs):

# Filter the ECG signal to remove noise using a bandpass filter

# For demonstration purposes, let's use a simple Butterworth bandpass filter

lowcut = 0.5 # Lower cutoff frequency (Hz)

highcut = 40 # Higher cutoff frequency (Hz)

order = 4 # Filter order

nyquist = 0.5 \* fs

low = lowcut / nyquist

high = highcut / nyquist

b, a = butter(order, [low, high], btype='band')

filtered\_ecg\_signal = filtfilt(b, a, ecg\_signal)

return filtered\_ecg\_signal

def detect\_r\_peaks(ecg\_signal):

# Detect R-peaks in the ECG signal using peak detection algorithm

peaks, \_ = find\_peaks(ecg\_signal, distance=0.2\*fs) # Adjust distance as per expected heart rate

return peaks

def calculate\_heart\_rate(peaks, fs):

# Calculate heart rate from R-peaks

heart\_rate = 60 \* len(peaks) / (len(ecg\_signal) / fs)

return heart\_rate

def plot\_ecg\_with\_r\_peaks(t, ecg\_signal, peaks):

# Plot ECG signal with detected R-peaks

plt.figure(figsize=(10, 6))

plt.plot(t, ecg\_signal, label='ECG Signal')

plt.plot(t[peaks], ecg\_signal[peaks], 'ro', markersize=5, label='R-Peaks')

plt.xlabel('Time (s)')

plt.ylabel('Amplitude')

plt.title('ECG Signal with Detected R-Peaks')

plt.legend()

plt.grid(True)

plt.show()

# Example usage:

file\_path = "ecg\_signal.txt" # Path to the ECG signal file

t, ecg\_signal = load\_ecg\_signal(file\_path)

fs = 1000 # Sampling frequency (Hz)

# Filter the ECG signal to remove noise

filtered\_ecg\_signal = filter\_ecg\_signal(ecg\_signal, fs)

# Detect R-peaks in the filtered ECG signal

peaks = detect\_r\_peaks(filtered\_ecg\_signal)

# Calculate heart rate from R-peaks

heart\_rate = calculate\_heart\_rate(peaks, fs)

# Plot ECG signal with detected R-peaks

plot\_ecg\_with\_r\_peaks(t, filtered\_ecg\_signal, peaks)

print("Heart Rate:", heart\_rate, "bpm")