AWM Sniping Bot: Backend (Flask)

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
from web3 import Web3
import hashlib

app = Flask(__name__)

# Configurations for database (PostgreSQL)
app.config['SQLALCHEMY_DATABASE_URI'] =
'postgresql://username:password@localhost/awm_sniping_bot_db'
db = SQLAlchemy(app)

# Model for Admin with hashed password
class Admin(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), nullable=False)
    password = db.Column(db.String(120), nullable=False)

# Model for storing user wallet information
class Wallet(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, nullable=False)
    wallet_address = db.Column(db.String(120), nullable=False)
    network = db.Column(db.String(10), nullable=False)  # 'ETH' or 'SOL'

# Initialize the database
db.create_all()

# Admin login endpoint
@app.route('/admin_login', methods=['POST'])
def admin_login():
    data = request.json
    username = data['username']
    password = hashlib.sha256(data['password'].encode()).hexdigest()
    admin = Admin.query.filter_by(username=username, password=password).first()
    if admin:
        return jsonify({"message": "Login successful"}), 200
    return jsonify({"message": "Invalid credentials"}), 401

# Admin password is initially set to 'Admin'
if not Admin.query.filter_by(username='Admin').first():
    admin_password = hashlib.sha256("Admin".encode()).hexdigest()
    admin = Admin(username='Admin', password=admin_password)
    db.session.add(admin)
    db.session.commit()

# Update user wallet settings
@app.route('/update_wallet', methods=['POST'])
def update_wallet():
    data = request.json
    user_id = data['user_id']
    wallet_address = data['wallet_address']
    network = data['network']  # ETH or SOL

    wallet = Wallet.query.filter_by(user_id=user_id).first()
    if wallet:
        wallet.wallet_address = wallet_address
        wallet.network = network
    else:
        new_wallet = Wallet(user_id=user_id, wallet_address=wallet_address, network=network)
        db.session.add(new_wallet)
    db.session.commit()
    return jsonify({"message": "Wallet updated successfully"}), 200

# Placeholder for trading bot logic
@app.route('/start_bot', methods=['POST'])
def start_bot():
    # Execute the trading logic based on user configuration (MACD, RSI, etc.)
    return jsonify({"message": "AWM Sniping Bot started"}), 200

if __name__ == '__main__':
    app.run(debug=True)
```

AWM Sniping Bot: Frontend (React.js)

```
import React, { useState } from 'react';

function AdminLogin() {
    const [username, setUsername] = useState('');
    const [password, setPassword] = useState('');

    const handleLogin = () => {
        fetch('/admin_login', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ username, password }),
        })
            .then(response => response.json())
            .then(data => {
                alert(data.message);
            });
    };

    return (
        <div>
            <h2>Admin Login</h2>
            <input type="text" placeholder="Username" value={username} onChange={(e) =>
setUsername(e.target.value)} />
```

```
            <input type="password" placeholder="Password" value={password} onChange={(e)
=> setPassword(e.target.value)} />
            <button onClick={handleLogin}>Login</button>
        </div>
    );
}

export default AdminLogin;
```