# Reinforcement Learning based Trading Bot using Alpha Vantage & Ethereum Smart Contracts

## Overview
This project leverages the Alpha Vantage API for stock market data and integrates an Ethereum smart contract for decentralized trading operations. The system allows users to fetch stock data, predict trends using ML models, and execute trades securely via blockchain.

## Features
- Fetch real-time and historical stock data using Alpha Vantage
- Predict stock trends using machine learning
- Execute trades via an Ethereum smart contract
- Secure and decentralized trading mechanism

## Tech Stack
- **Backend**: Flask, Python, Alpha Vantage API
- **Frontend**: React.js
- **Blockchain**: Solidity, Ethereum, Web3.js
- **Database**: PostgreSQL
- **ML Model**: Scikit-Learn, TensorFlow

## Setup Instructions

### 1. Clone the Repository
```sh
git clone https://github.com/your-repo-url.git
cd your-project-folder
```

### 2. Install Dependencies
```sh
pip install -r requirements.txt
cd frontend && npm install
```

### 3. Setup Alpha Vantage API
- Register on [Alpha Vantage](https://www.alphavantage.co/)
- Get your API key
- Add it to `.env`:
  ```sh
  ALPHA_VANTAGE_API_KEY=your_api_key_here
  ```

### 4. Ethereum Smart Contract Deployment

#### Prerequisites
- Install [Node.js](https://nodejs.org/)
- Install [Hardhat](https://hardhat.org/)

```sh
npm install --save-dev hardhat
```
- Install Web3.js
```sh
npm install web3
```

#### Deploy Contract
1. Navigate to the Solidity contract directory:
   ```sh
   cd blockchain
   ```
2. Compile the contract:
   ```sh
   npx hardhat compile
   ```
3. Deploy the contract to a test network (Goerli, Sepolia, or Local Hardhat):
   ```sh
   npx hardhat run scripts/deploy.js --network goerli
   ```
4. Save the deployed contract address and update it in the frontend `.env` file:
   ```sh
   REACT_APP_CONTRACT_ADDRESS=your_deployed_contract_address
   ```

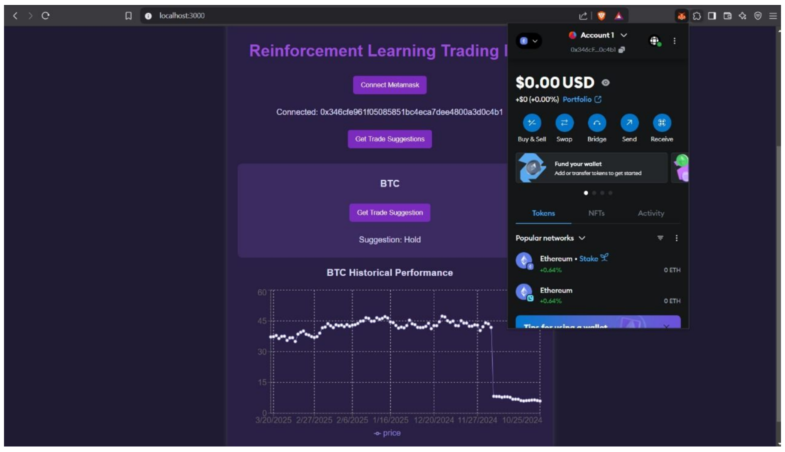
### 5. Run the Backend Server
```sh
flask run
```

### 6. Run the Frontend
```sh
cd frontend
npm start
```

## Smart Contract Overview

### **Contract: StockTrading.sol**
```solidity
pragma solidity ^0.8.0;
contract StockTrading {
    mapping(address => uint256) public balances;
    function deposit() public payable {
        balances[msg.sender] += msg.value;
    }
    function trade(address receiver, uint256 amount) public {
        require(balances[msg.sender] >= amount, "Insufficient funds");
```

~~~
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
    }
}
```

### Interacting with the Contract
```js
const contract = new web3.eth.Contract(ABI, CONTRACT_ADDRESS);
await contract.methods.trade(receiverAddress, amount).send({ from: senderAddress });
```

---
## Deployment
### 1. Deploy Backend to Heroku
```sh
git push heroku main
```
### 2. Deploy Frontend to Vercel
```sh
vercel --prod
```
### 3. Verify Ethereum Contract Deployment
Use [Etherscan](https://etherscan.io/) to verify and interact with the contract.

---
## Future Improvements
- Implement AI-driven trading strategies
- Support for additional blockchain networks
- Multi-user trading pool integration

---
## Contributors
- Your Name (@yourGitHub)


## screenshot
~~~

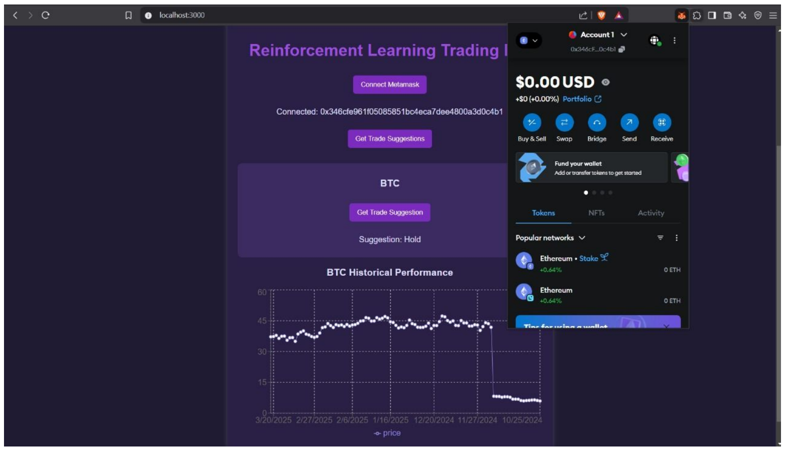


Connected: 0x346cfe961f05085851bc4eca7dee4800a3d0c4b1

Suggestion: Hold

$0.00 USD
+$0 (+0.00%)

Fund your wallet
Add or transfer tokens to get started



Reinforcement Learning Trading

Connected: 0x346cfe961f05085851bc4eca7dee4800a3d0c4b1

BTC

Suggestion: Hold

0x346cF...0c4b1

$0.00 USD

+$0 (+0.00%) Portfolio

Fund your wallet
Add or transfer tokens to get started

Ethereum • Stake
+0.64%
0 ETH

Ethereum
+0.64%
0 ETH