

# **Crypto Transaction Tracker**

## **Developers Studio**

Developed by

Bilal Arif

Domain: Blockchain

Team Lead: Hamza Rasheed Chaudhary

## Table of contents:

Introduction: .....	3
Scope of project: .....	3
Functionalities: .....	3
Error handling: .....	4
Technology used: .....	4
Workflow diagram: .....	7
User:.....	5
Department: .....	6

## Introduction:

The crypto transaction tracker is a comprehensive tool that is used to monitor the transaction of addresses that is added to watchlist. This is an automated tool that keeps looking for upcoming transactions of wallet addresses, to be up to date and be informed about transactions. It helps to keep up to date on any transactions performed by the watchlist address. The application also notifies you via sending messages in the message's app “Slack”

## Scope of project:

The crypto Transaction main purpose is to track the upcoming new transactions performed by the address that has been added to watchlist. The watchlist consists of those addresses that will be tracked by system. But the transactions that are only tracked are Ethereum based on ERC-20 Token based transactions. Internal transactions are not being tracked. Also, the transactions that were performed when the system application was offline will also not be tracked and hence only those transactions will be saved when the system is online and running. Here are the functionalities of the application.

## Functionalities:

### 1. Add address to watchlist

To track a wallet address, the system requires a list of addresses from a database which will be tracked. This is a **POST** method **API endpoint** that will be executed to add address to watchlist in MongoDB database. Here are the parameters required to run this API

- **Url:** <http://localhost:5000/address/addAddress>

- **Body:** address: "address"
- **Method:** POST

## 2. Get Transactions of address

To track a transaction of a specific address, the user can access transactions tracked by this system of address via this API. This is a **GET** method **API endpoint** that will be executed to fetch the transactions of the given address. Here are the parameters required to run this API

- **Url:** http://localhost:5000/address/viewAddressTransactions/:address
- **Method:** GET

## 3. Transaction Watcher

This is the main transaction of the application. The transaction watcher uses these procedures step by step to track addresses.

- Event emitted by JSON rpc provider that occurs whenever the block is mined, and latest block is refreshed. This keeps executing on latest block change.
- Getting the transactions of the latest block.
- Fetching watchlist addresses from the database and comparing them with the transactions to see if the transaction list includes the address or not.
- Checking if the type of transaction performed by the wallet address via getting the transaction receipt of transactions and identifying their logs in the receipt. The logs will identify if the transaction is **ETH** normal transaction or **ERC-20** transaction.
- The transaction will be pushed to watchlist address's list of transactions.
- The notifications will be triggered on slack via sending the message to channel, which displays that the address has performed a transaction.

### Error handling:

- Address input errors if the address length is not 42. All Ethereum based crypto wallet addresses consist of 42 characters.
- Errors related to the database if specific entry is not found such as address or transaction.
- Errors related to API calls will be printed to terminal of the program.

### Technology used:

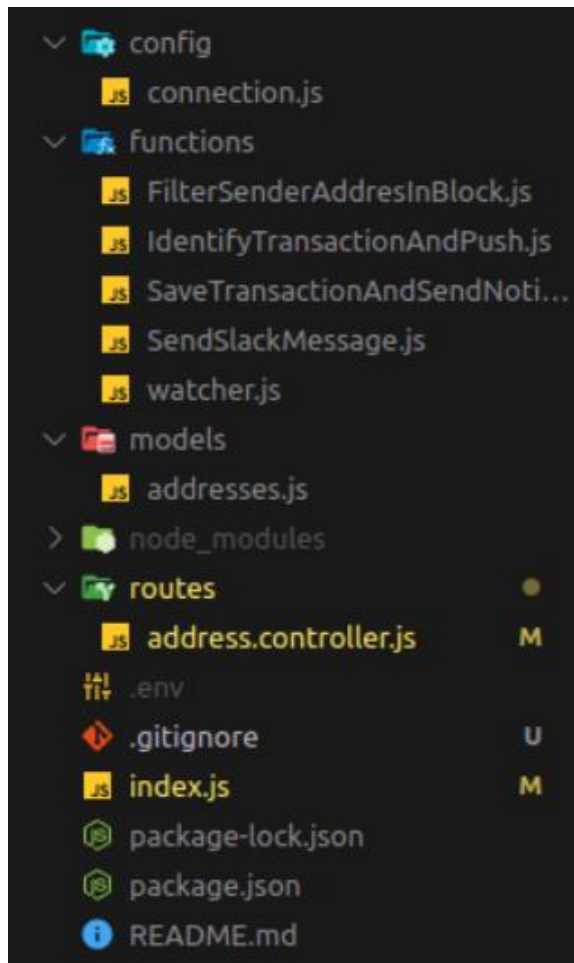
- Node.js
- Express.js

- MongoDB

#### Libraries used:

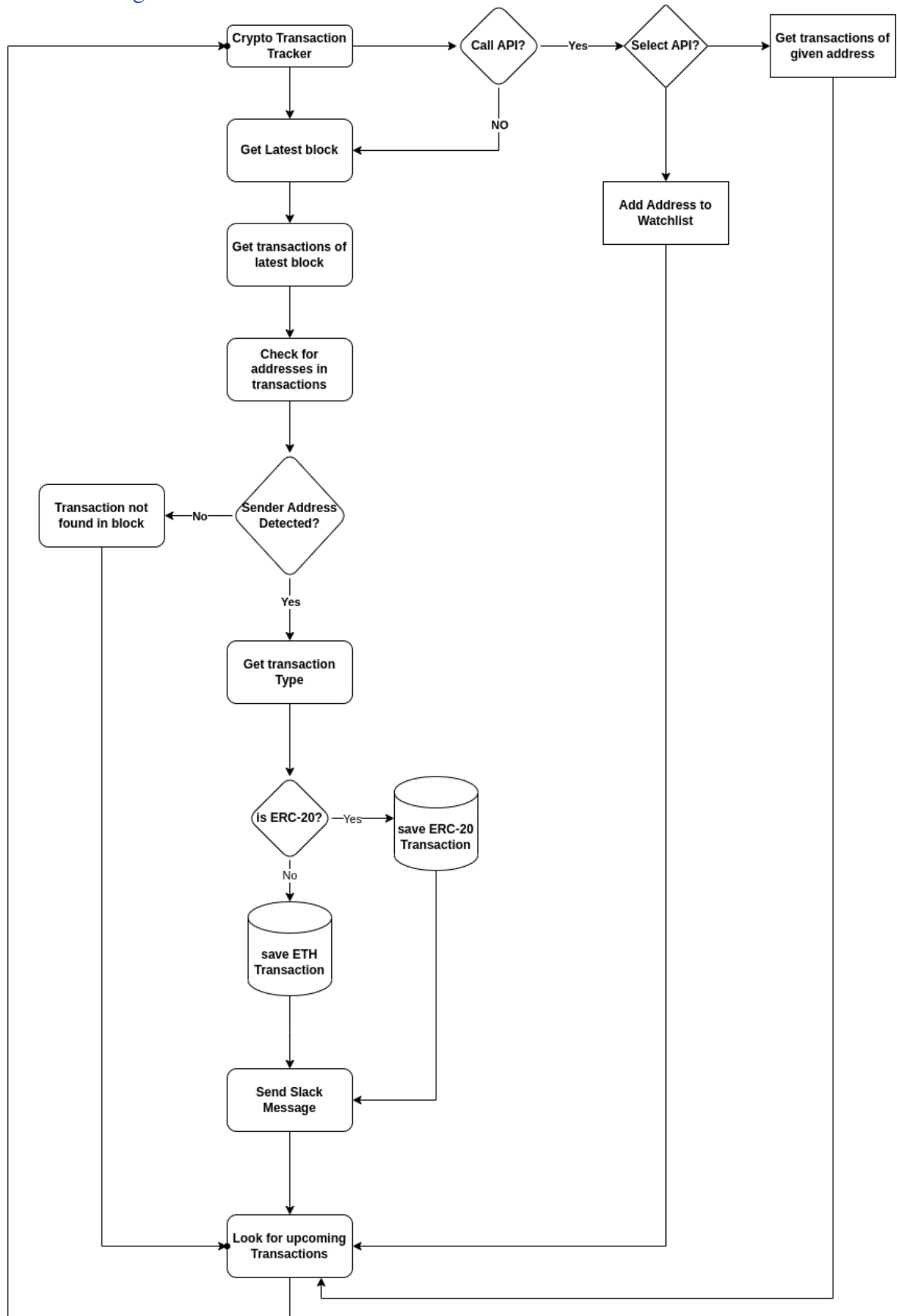
- Ethers.js
- Axios
- Dot env
- Nodemon

#### Folder structure:

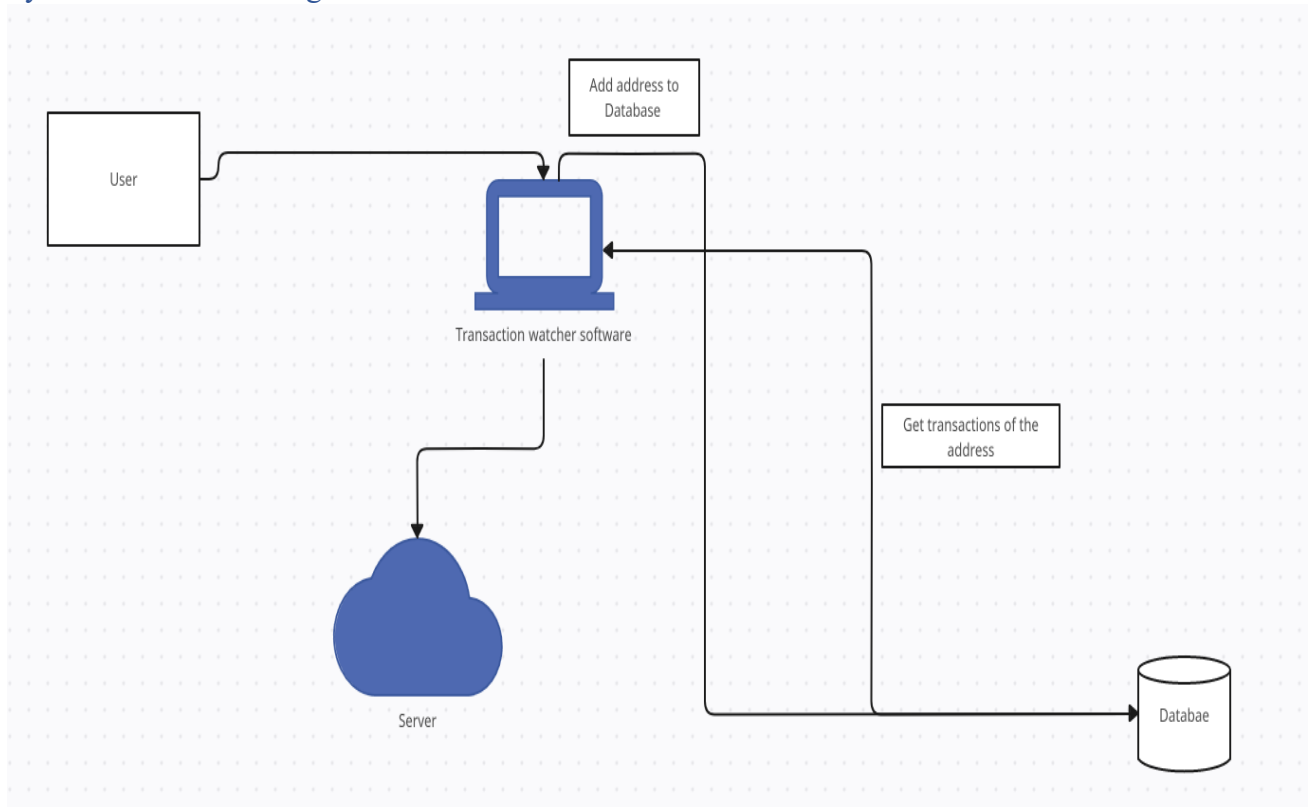




Workflow diagram:



### System architecture diagram:





Database diagram:

