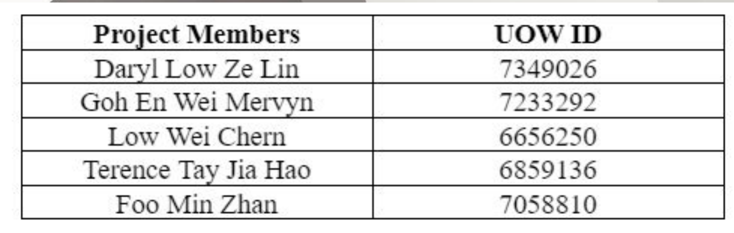
Final Presentation

FYP-22-S3-11



Project Members	UOW ID
Daryl Low Ze Lin	7349026
Goh En Wei Mervyn	7233292
Low Wei Chern	6656250
Terence Tay Jia Hao	6859136
Foo Min Zhan	7058810

PRODUCT TRAILER

Table of Contents

Existing Competitors

Development Methodology

Final Product

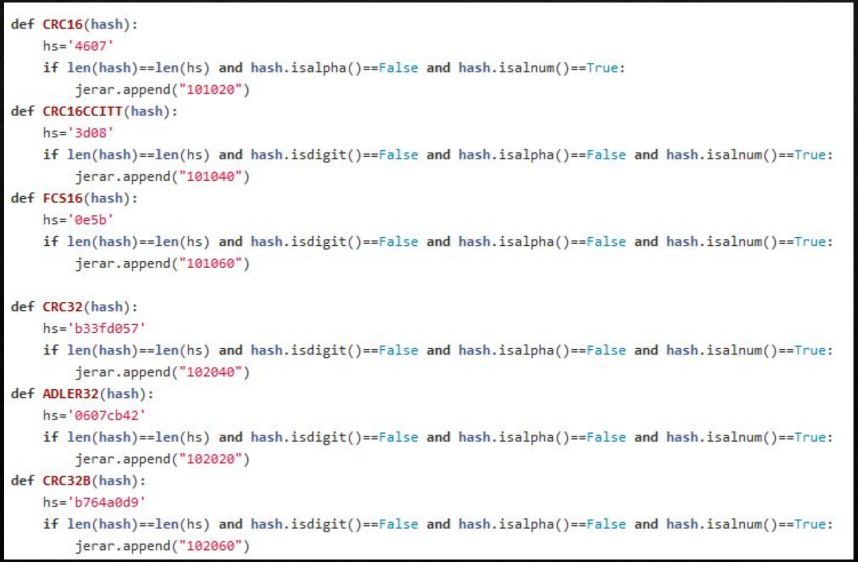
Final Product Composition

Product Demo

Existing Competitors

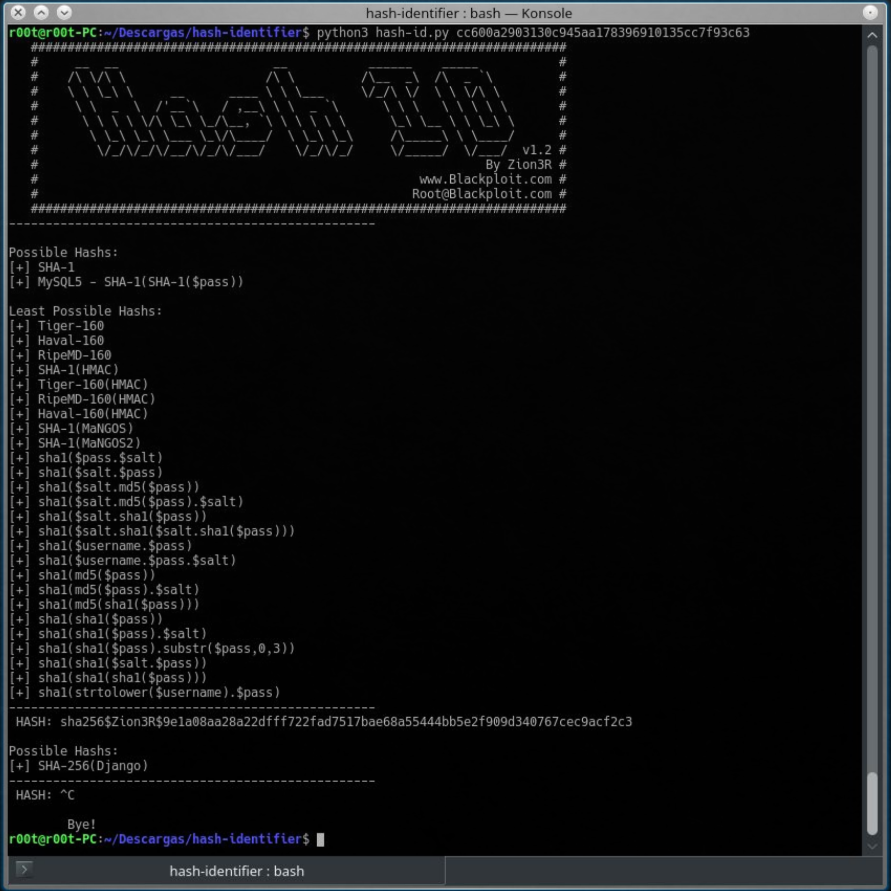
Identifies Hash based on Input

Cross References based on in-built patterns



```python
def CRC16(hash):
    hs='4607'
    if len(hash)==len(hs) and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("101020")
def CRC16CCITT(hash):
    hs='3d08'
    if len(hash)==len(hs) and hash.isdigit()==False and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("101040")
def FCS16(hash):
    hs='0e5b'
    if len(hash)==len(hs) and hash.isdigit()==False and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("101060")

def CRC32(hash):
    hs='b33fd057'
    if len(hash)==len(hs) and hash.isdigit()==False and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("102040")
def ADLER32(hash):
    hs='0607cb42'
    if len(hash)==len(hs) and hash.isdigit()==False and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("102020")
def CRC32B(hash):
    hs='b764a0d9'
    if len(hash)==len(hs) and hash.isdigit()==False and hash.isalpha()==False and hash.isalnum()==True:
        jerar.append("102060")
```

KALI Hash Identifier



```
r00t@r00t-PC:~/Descargas/hash-identifier$ python3 hash-id.py cc600a2903130c945aa178396910135cc7f93c63
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------

Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))

Least Possible Hashs:
[+] Tiger-160
[+] Haval-160
[+] RipeMD-160
[+] SHA-1(HMAC)
[+] Tiger-160(HMAC)
[+] RipeMD-160(HMAC)
[+] Haval-160(HMAC)
[+] SHA-1(MaNGOS)
[+] SHA-1(MaNGOS2)
[+] sha1($pass.$salt)
[+] sha1($salt.$pass)
[+] sha1($salt.md5($pass))
[+] sha1($salt.md5($pass).$salt)
[+] sha1($salt.sha1($pass))
[+] sha1($salt.sha1($salt.sha1($pass)))
[+] sha1($username.$pass)
[+] sha1($username.$pass.$salt)
[+] sha1(md5($pass))
[+] sha1(md5($pass).$salt)
[+] sha1(md5(sha1($pass)))
[+] sha1(sha1($pass))
[+] sha1(sha1($pass).$salt)
[+] sha1(sha1($pass).substr($pass,0,3))
[+] sha1(sha1($salt.$pass))
[+] sha1(sha1(sha1($pass)))
[+] sha1(strtolower($username).$pass)
--------------------------------------------------
 HASH: sha256$Zion3R$9e1a08aa28a22dfff722fad7517bae68a55444bb5e2f909d340767cec9acf2c3

Possible Hashs:
[+] SHA-256(Django)
--------------------------------------------------
 HASH: ^C

        Bye!
r00t@r00t-PC:~/Descargas/hash-identifier$ 
```

Existing Competitors

USP of our Product :

Due to the handling of:

- Block Hashes
- Blockchain Source Files

The intended target user is already very specific and narrow

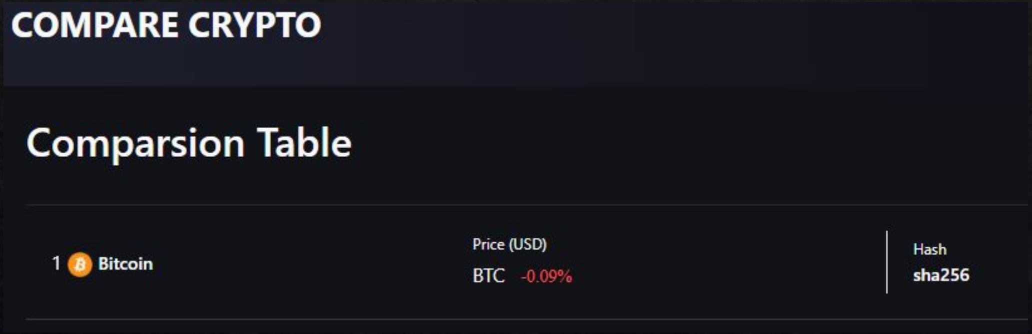
Our product is intended to this niche group of users

Our product offers analysis on the Blockchain Source files

Final Product

2 Parts

WebApp



COMPARE CRYPTO

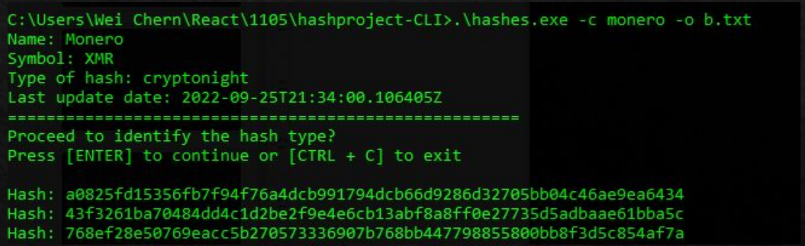
Comparsion Table

1 Bitcoin	Price (USD) BTC -0.09%	Hash sha256

View Blockchains

Blockchain Analysis

Command Line Interface (CLI) Program



```
C:\Users\Wei Chern\React\1105\hashproject-CLI>.\hashes.exe -c monero -o b.txt
Name: Monero
Symbol: XMR
Type of hash: cryptonight
Last update date: 2022-09-25T21:34:00.106405Z
=====================================================
Proceed to identify the hash type?
Press [ENTER] to continue or [CTRL + C] to exit

Hash: a0825fd15356fb7f94f76a4dcb991794dcb66d9286d32705bb04c46ae9ea6434
Hash: 43f3261ba70484dd4c1d2be2f9e4e6cb13abf8a8ff0e27735d5adbaae61bba5c
Hash: 768ef28e50769eacc5b270573336907b768bb447798855800bb8f3d5c854af7a
```

View Information Quickly

Identifying Hash Function from a Hash Block

Development Methodology

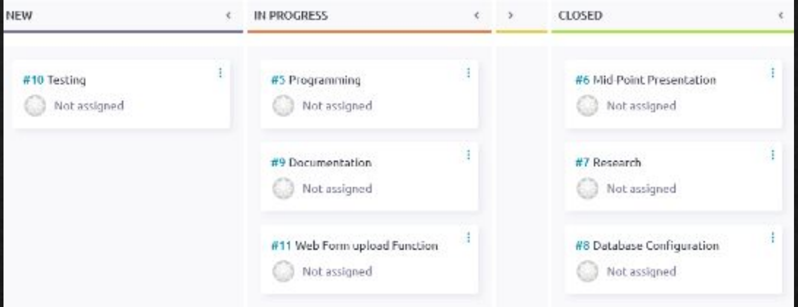
SCRUM was chosen

Familiarity, after taking 314

Time Constraints

2 Month Long Sprints

Bi-weekly Review



NEW	IN PROGRESS	CLOSED
#10 Testing Not assigned	#5 Programming Not assigned	#6 Mid Point Presentation Not assigned
	#9 Documentation Not assigned	#7 Research Not assigned
	#11 Web Form upload Function Not assigned	#8 Database Configuration Not assigned

Why 2 X Long Sprints Instead of Short Sprints?

Progress was Slow

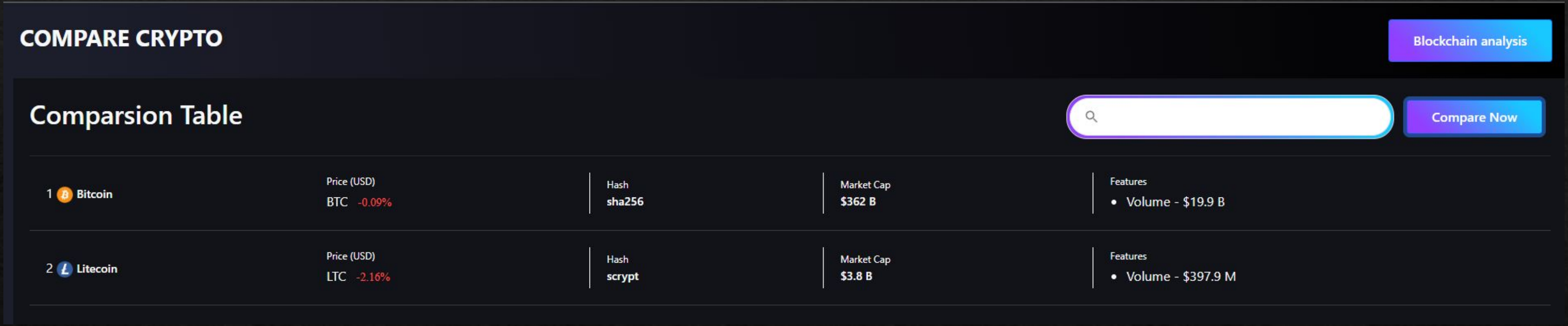
Research was slow

Progress was Erratic and Abrupt

Astonishing Progress Could have been made in a week

While bugs, errors, took an unexpected amount of time to rectify

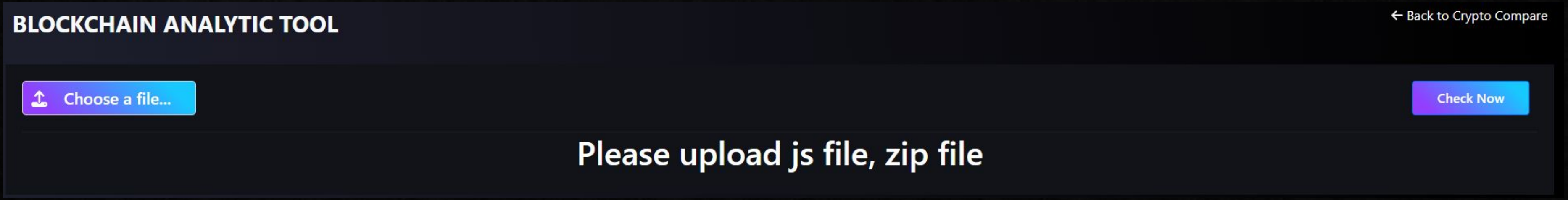
Webapp Final Product



COMPARE CRYPTO

Comparsion Table

1 Bitcoin	Price (USD) BTC -0.09%	Hash sha256	Market Cap $362 B	Features • Volume - $19.9 B
2 Litecoin	Price (USD) LTC -2.16%	Hash scrypt	Market Cap $3.8 B	Features • Volume - $397.9 M

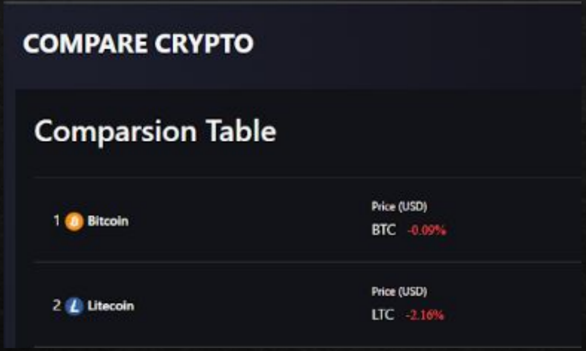


BLOCKCHAIN ANALYTIC TOOL

Please upload js file, zip file

Webapp Final Product

Comprises of 2 components:



COMPARE CRYPTO

Comparsion Table

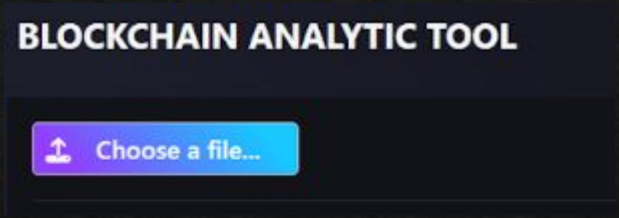
1 Bitcoin Price (USD) BTC -0.09%

2 Litecoin Price (USD) LTC -2.16%

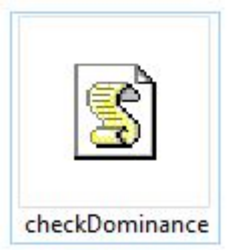
home

View hashes on Blockchain Cryptocurrencies

Web scrapping to display information



BLOCKCHAIN ANALYTIC TOOL



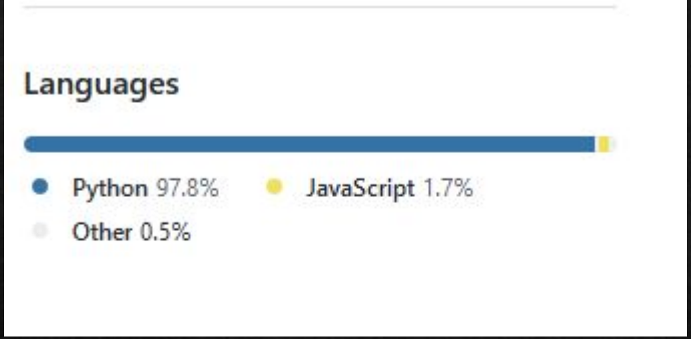
checkDominance

Accepts Cryptocurrency source files

Performs Analysis

Displays information to users

Webapp Composition

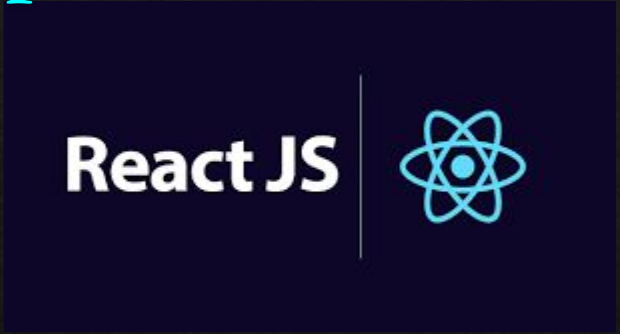


Languages

- ● Python 97.8%
- ● JavaScript 1.7%
- ● Other 0.5%

Front End:

home



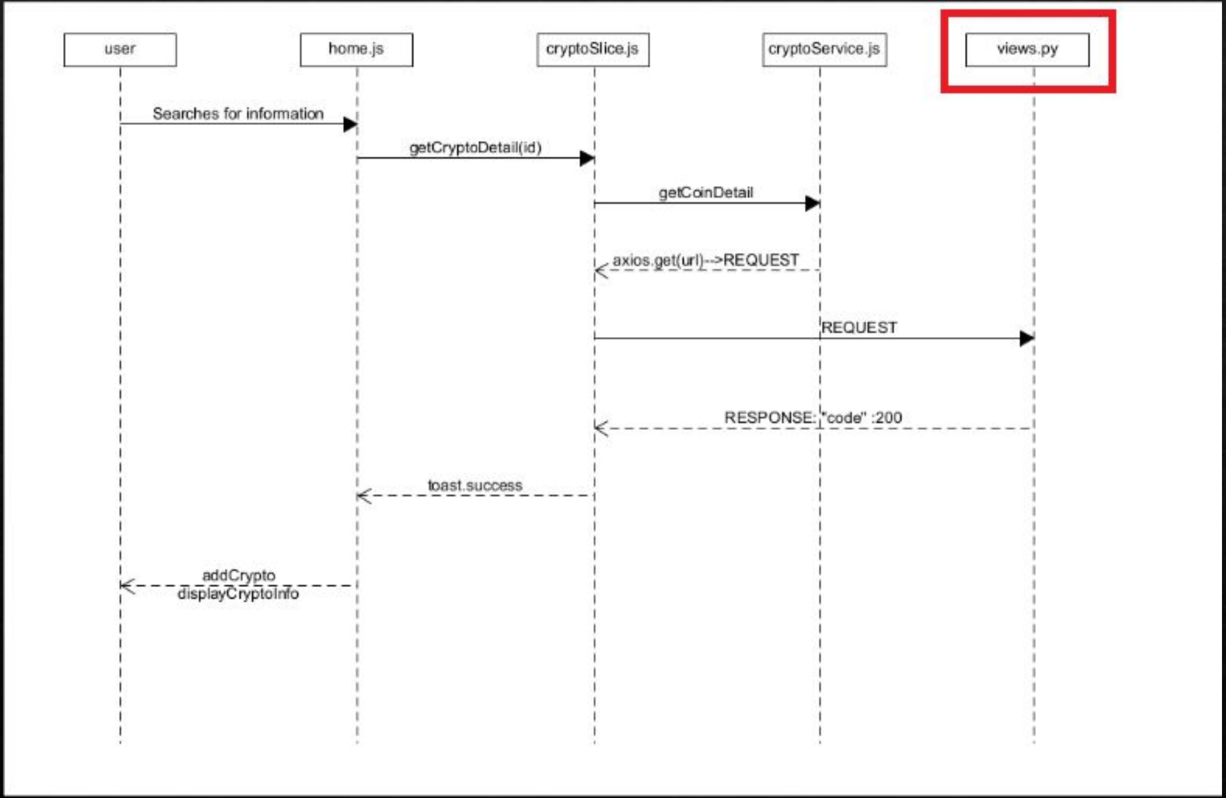
Backend :

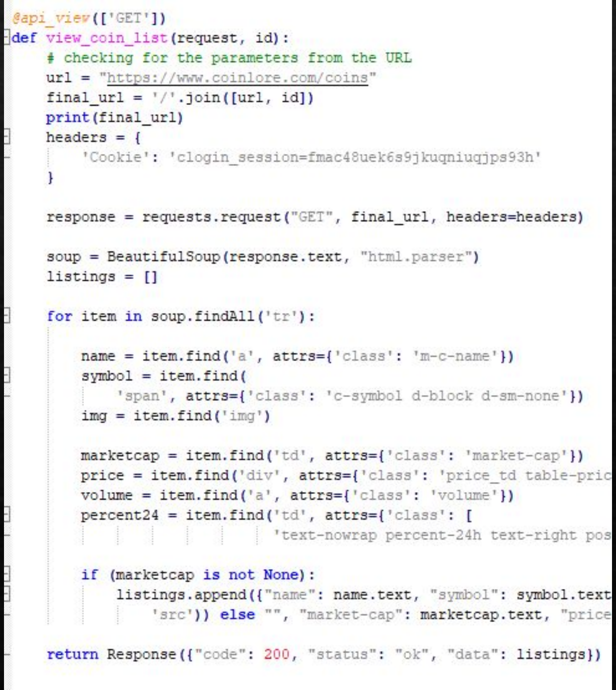
Django Models
Django Crypto

Web scrapping for information

Task Processing

Webapp Composition

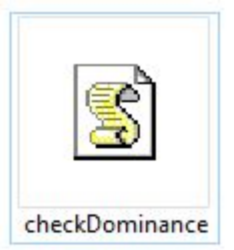




```
@api_view(['GET'])
def view_coin_list(request, id):
    # checking for the parameters from the URL
    url = "https://www.coinlore.com/coins"
    final_url = '/'.join([url, id])
    print(final_url)
    headers = {
        'Cookie': 'clogin_session=fmac48uek6s9jkuqniuqjps93h'
    }

    response = requests.request("GET", final_url, headers=headers)

    soup = BeautifulSoup(response.text, "html.parser")
    listings = []

    for item in soup.findAll('tr'):

        name = item.find('a', attrs={'class': 'm-c-name'})
        symbol = item.find(
            'span', attrs={'class': 'c-symbol d-block d-sm-none'})
        img = item.find('img')

        marketcap = item.find('td', attrs={'class': 'market-cap'})
        price = item.find('div', attrs={'class': 'price_td table-pric
        volume = item.find('a', attrs={'class': 'volume'})
        percent24 = item.find('td', attrs={'class': [
                                                'text-nowrap percent-24h text-right pos

        if (marketcap is not None):
            listings.append({"name": name.text, "symbol": symbol.text
                'src')) else "", "market-cap": marketcap.text, "price

    return Response({"code": 200, "status": "ok", "data": listings})
```

Webapp Composition

Front End:



React JS

Backend :

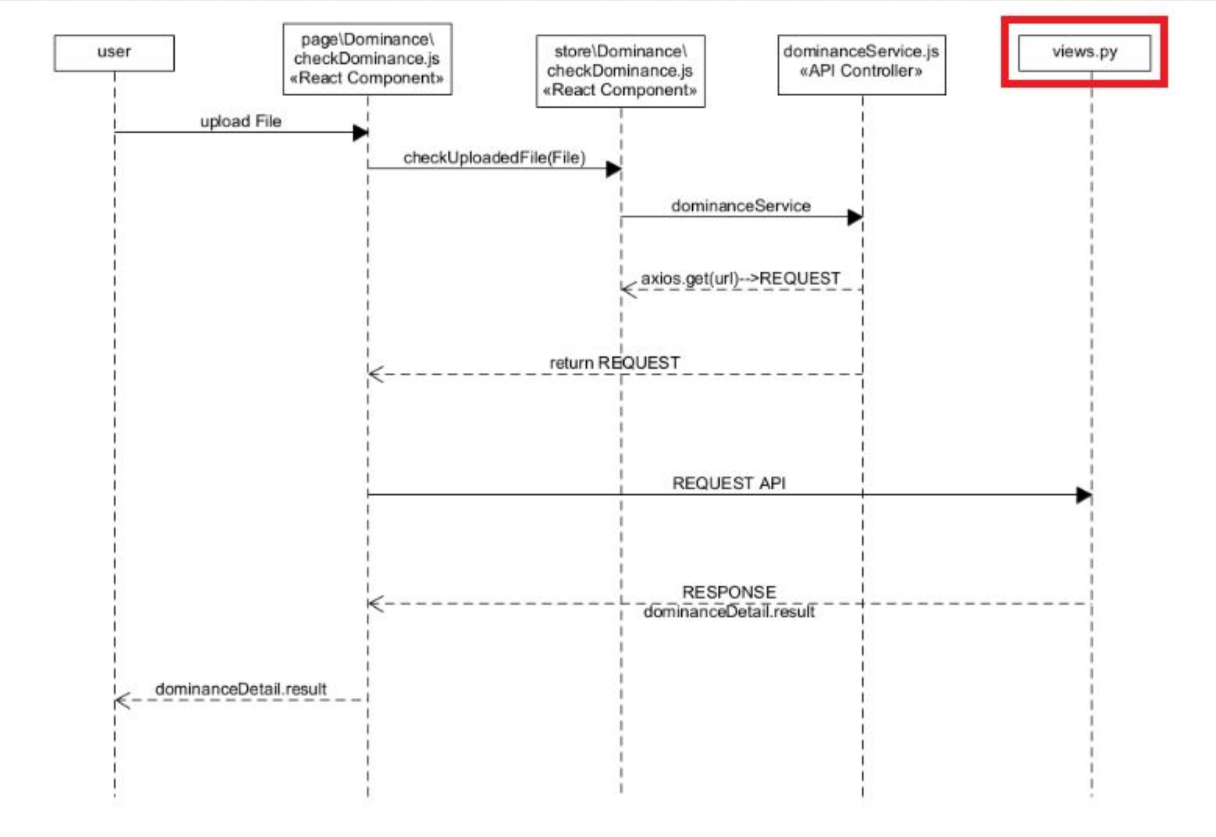
& django

Django Models
Django Crypto

Celery

Task Processing

Webapp Composition



user

page\Dominance\
checkDominance.js
«React Component»

store\Dominance\
checkDominance.js
«React Component»

dominanceService.js
«API Controller»

views.py

upload File

checkUploadedFile(File)

dominanceService

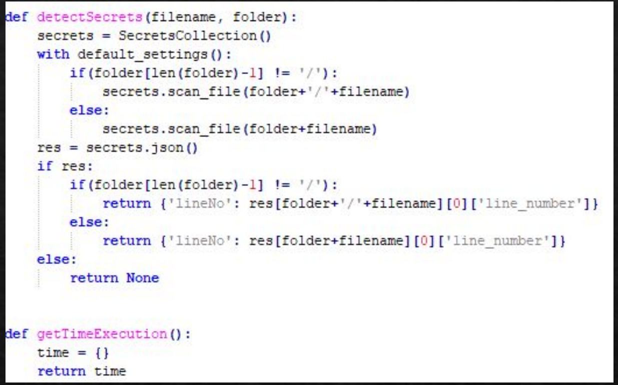
axios.get(url)-->REQUEST

return REQUEST

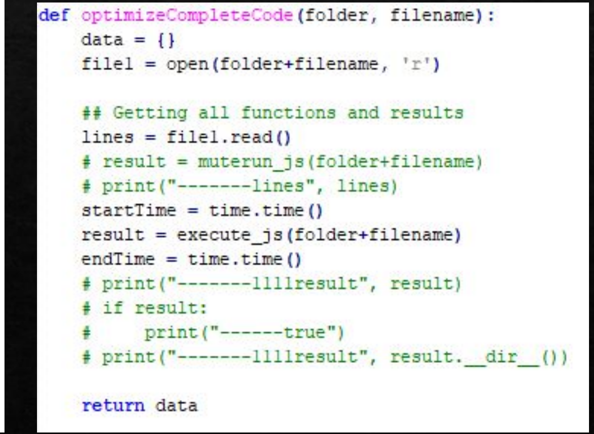
REQUEST API

RESPONSE
dominanceDetail.result

dominanceDetail.result



```
def detectSecrets(filename, folder):
    secrets = SecretsCollection()
    with default_settings():
        if(folder[len(folder)-1] != '/'):
            secrets.scan_file(folder+'/'+filename)
        else:
            secrets.scan_file(folder+filename)
    res = secrets.json()
    if res:
        if(folder[len(folder)-1] != '/'):
            return {'lineNo': res[folder+'/'+filename][0]['line_number']}
        else:
            return {'lineNo': res[folder+filename][0]['line_number']}
    else:
        return None


def getTimeExecution():
    time = {}
    return time
```



```
def optimizeCompleteCode(folder, filename):
    data = {}
    file1 = open(folder+filename, 'r')

    ## Getting all functions and results
    lines = file1.read()
    # result = muterun_js(folder+filename)
    # print("-------lines", lines)
    startTime = time.time()
    result = execute_js(folder+filename)
    endTime = time.time()
    # print("-------1111result", result)
    # if result:
    #     print("------true")
    # print("-------1111result", result.__dir__())

    return data
```

Command Line Interface (CLI) Program

Fast, Offline Hash Identification

Backup in the event the Web site is down / Unavailable

Identifying Hashes Input by User

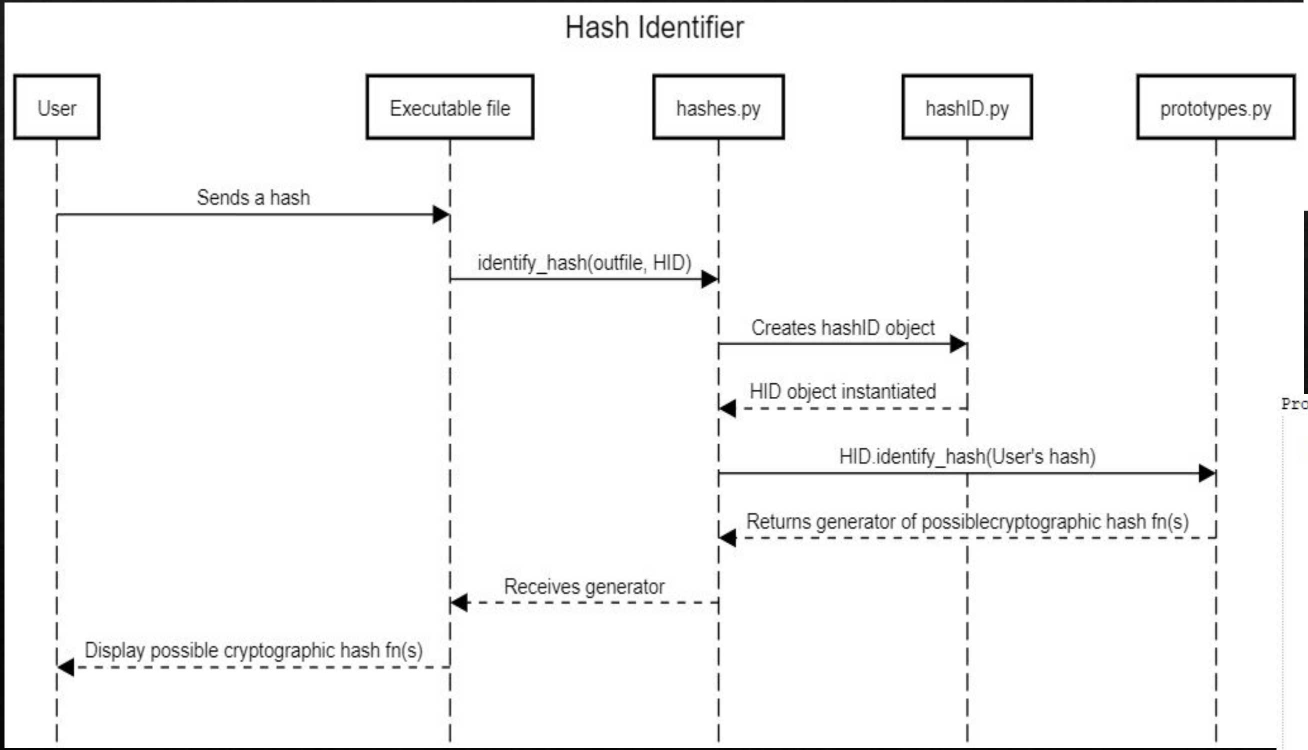
Identifies Possible Hash Functions

Identification via Length and Patterns, defined by the team

Retrieve Hash Information on Blockchains via Blockchain Name Input

Retrieve Cryptocurrency Information Based on input name

CLI Composition



Hash Identifier

User

Executable file

hashes.py

hashID.py

prototypes.py

Sends a hash

identify_hash(outfile, HID)

Creates hashID object

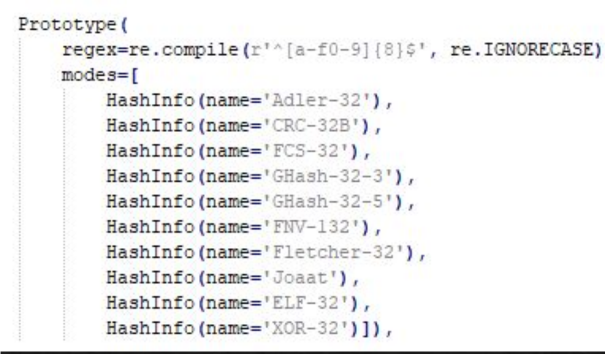
HID object instantiated

HID.identify_hash(User's hash)

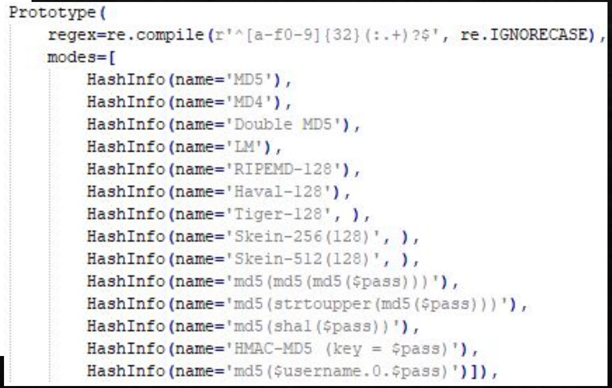
Returns generator of possiblecryptographic hash fn(s)

Receives generator

Display possible cryptographic hash fn(s)

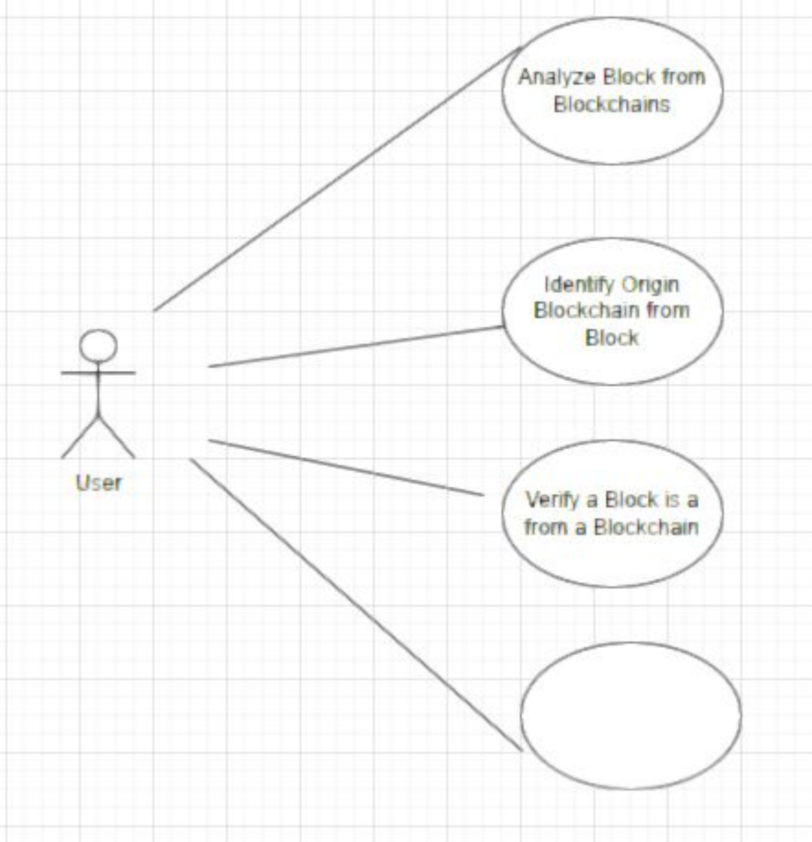


```
Prototype(
    regex=re.compile(r'^[a-f0-9]{8}$', re.IGNORECASE)
    modes=[
        HashInfo(name='Adler-32'),
        HashInfo(name='CRC-32B'),
        HashInfo(name='FCS-32'),
        HashInfo(name='GHash-32-3'),
        HashInfo(name='GHash-32-5'),
        HashInfo(name='FNV-132'),
        HashInfo(name='Fletcher-32'),
        HashInfo(name='Joaat'),
        HashInfo(name='ELF-32'),
        HashInfo(name='XOR-32')]),
```



```
Prototype(
    regex=re.compile(r'^[a-f0-9]{32}(:.+)?$', re.IGNORECASE),
    modes=[
        HashInfo(name='MD5'),
        HashInfo(name='MD4'),
        HashInfo(name='Double MD5'),
        HashInfo(name='LM'),
        HashInfo(name='RIPEMD-128'),
        HashInfo(name='Haval-128'),
        HashInfo(name='Tiger-128', ),
        HashInfo(name='Skein-256(128)', ),
        HashInfo(name='Skein-512(128)', ),
        HashInfo(name='md5(md5(md5($pass)))'),
        HashInfo(name='md5(strtoupper(md5($pass)))'),
        HashInfo(name='md5(sha1($pass))'),
        HashInfo(name='HMAC-MD5 (key = $pass)'),
        HashInfo(name='md5($username.0.$pass)')]),
```

User Stories


Analyze Block from Blockchains
Identify Origin Blockchain from Block
Verify a Block is a from a Blockchain
User

User Stories presented in our first Requirement Specification

Initial User Stories Promised too little and also too Much

WRONG Terms used / Promised

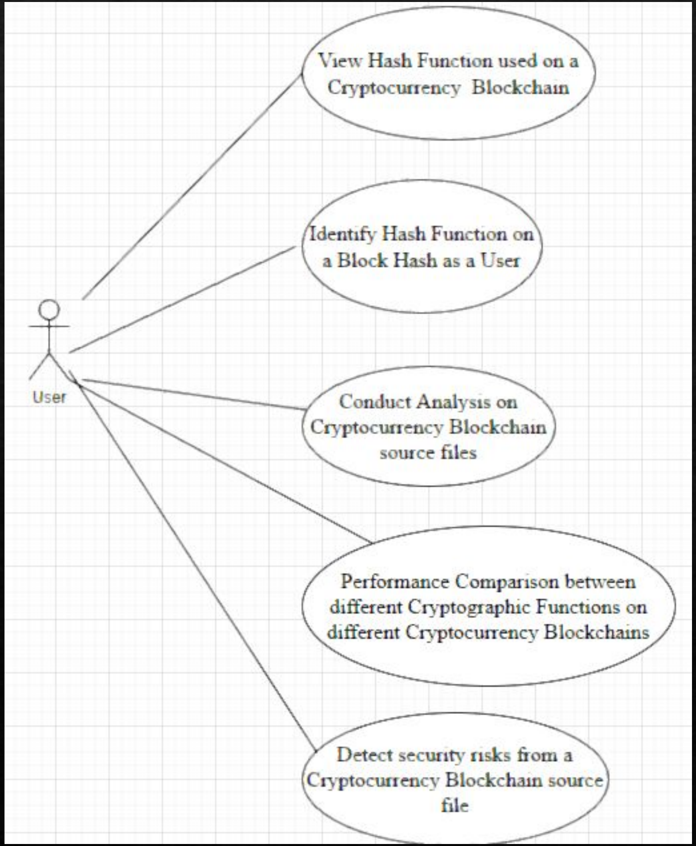
After the Mid-Point presentation

User Stories had to be Changed

Our requirements were better defined

Direction of project was clearer

Updated User Stories


View Hash Function used on a Cryptocurrency Blockchain
Identify Hash Function on a Block Hash as a User
User
Conduct Analysis on Cryptocurrency Blockchain source files
Performance Comparison between different Cryptographic Functions on different Cryptocurrency Blockchains
Detect security risks from a Cryptocurrency Blockchain source file

Misuse of the proper Blockchain terms distorted the group's intended proposals

The Requirement Specification had to be redrafted by our team

User Story Demo

User Story: View Hash Function used on a Cryptocurrency Blockchain

User Story: Conduct Analysis on Cryptocurrency Blockchain source file

User Story: Detect Security Risks on a Cryptocurrency Blockchain source file

Webapp Final Product

User Story Demo

User Story: View Hash Function used on a Cryptocurrency Blockchain

User Story: Identify Hash Function from a given Block Hash

Command Line Interface (CLI) Program