S.C.A.M

Secure Cryptocurrency Asset Management Platform

PROJECT REPORT

Of

IE 6700: Data Management for Analytics

BY

GROUP NUMBER 18:

ARCHIT SINGH (002813253)

AMIT KANOJIA (002208440)

617-749-8646

857-313-0630

singh.arc@northeastern.edu

kanojia.a@northeastern.edu

Percentage of Effort Contributed by Student 1: 50

Percentage of Effort Contributed by Student2: 50

Signature of Student 1: Archit Singh Digitally signed by Archit Singh Date: 2023.12.10 00:06:54 -05'00'

Signature of Student 2: _____

Submission Date: 12/10/2023

USE CASE STUDY REPORT

GROUP NUMBER: 18

STUDENTS NAME: ARCHIT SINGH AND AMIT KANOJIA

Executive Summary:

As the popularity of cryptocurrencies continues to grow, our project aims to provide a reliable, industry-ready relational database to effectively manage risk. While traditional forms of investment offer predictable risks, low volatility, and stable returns, the potential returns from cryptocurrency investments far exceed those of stocks, bonds, and real estate. However, it is important not to overlook the risks associated with cryptocurrencies, such as high volatility, technical complexity, and a general lack of understanding of the financial markets. Our project's goal is to significantly reduce the risks involved in cryptocurrency trading and increase overall safety for our users. Moreover, the project provides users with a proactive approach to every cryptocurrency investment made across different platforms.

To achieve this, we designed the database considering the requirements of a user along with various centralized and decentralised viewpoints. Based on the considered data fields, we created the conceptual models for both EER and UML, followed by the mapping of the conceptual model to the relational model, and primary and foreign keys were properly defined. In addition, the logical model was then normalised to reduce data redundancy, improve data integrity, and enhance query performance.

Furthermore, the database was implemented in MySQL for its reliability, scalability, open-source nature, and widespread support for various data-driven applications, as well as in a NoSQL implementation.

Lastly, connecting the database to Python was a great success. We were able to deliver immense data analysis for each user in terms of Risk Analysis, Tax Analysis, Diversification Analysis, and much more. The next stage of improvement would be setting up the backend code and the frontend to create a fully functional application that can be released into the real world.

1. INTRODUCTION

The rise of cryptocurrencies in today's shifting financial markets has captivated investors with the promise of substantial returns and unique investment opportunities. Unlike traditional investment options with calculable risks and stable returns, cryptocurrencies have introduced an unprecedented level of volatility and complexity. The prospect of significant returns, on the other hand, is offset by the inherent risks associated with factors such as high volatility, technical complexities, and a lack of fundamental understanding in the financial market.
Recognizing the challenges and opportunities in the cryptocurrency world, our project sets out on a mission to address one of the most pressing issues in this domain: risk management in cryptocurrency trading. The goal is to capitalize on the potential of cryptocurrencies while minimizing the risks associated with them. This project is especially important because of the growing interest in diversifying investment portfolios with digital assets, which necessitates a strategic and secure approach to cryptocurrency asset management.

Background Information:

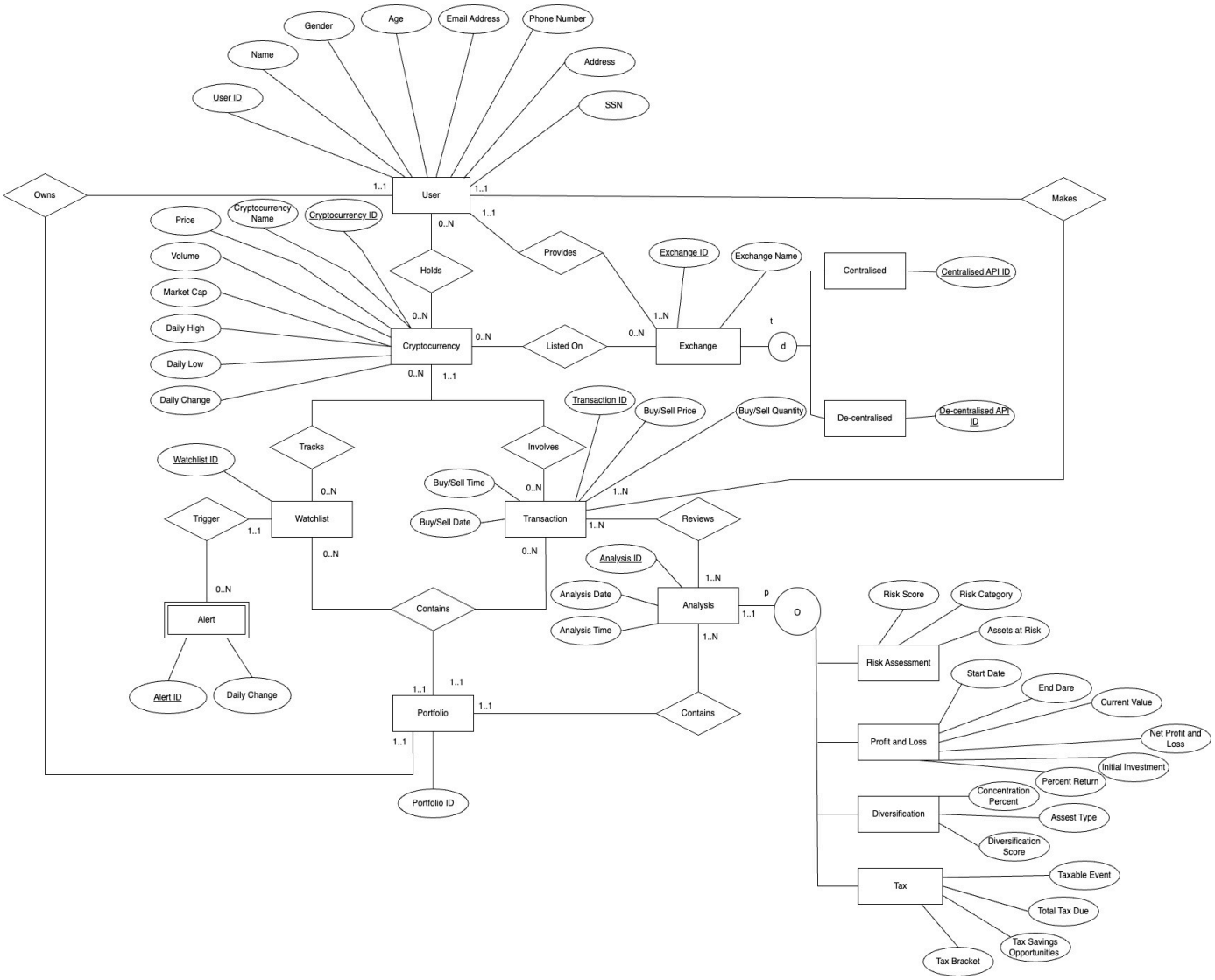
The Problem:
Cryptocurrency trading is fraught with risks such as market volatility, technical complexities, and a lack of comprehensive tools for portfolio management. Investors face the challenge of making informed decisions across various platforms, often leading to suboptimal outcomes and exposure to potential scams.

The Goal of the Study:
Risks associated with cryptocurrency trading include market volatility, technical complexities, and a lack of comprehensive portfolio management tools. Investors face the challenge of making informed decisions across multiple platforms, which often ends in suboptimal results and exposure to potential scams.

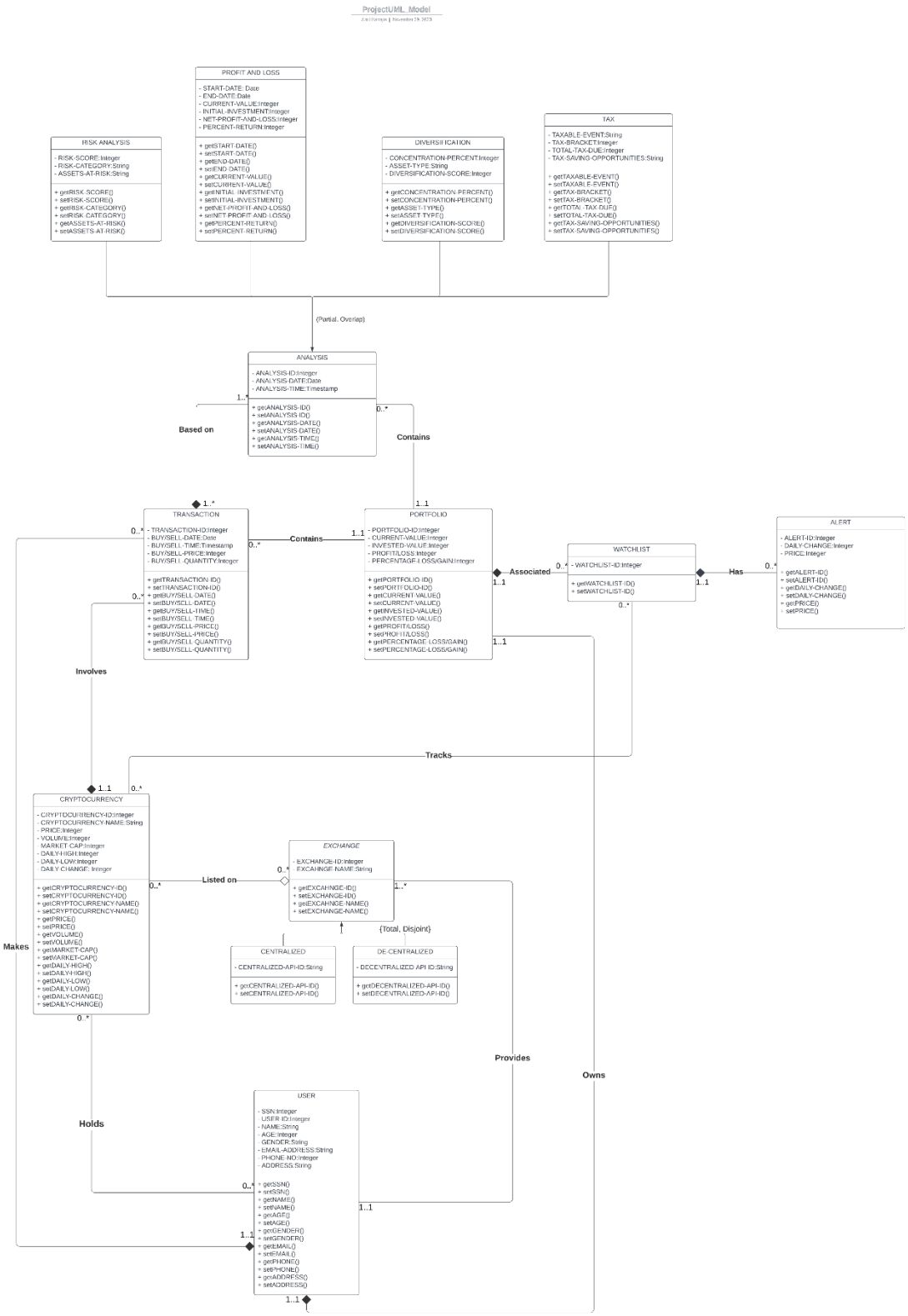
The Requirements:
To achieve our goal, we propose the development of a comprehensive platform that seamlessly integrates with both centralized and decentralized cryptocurrency exchange wallets. The platform will require user authentication through user ID and store essential details such as name, age, gender, email address, and contact information. Additionally, users will be able to connect their exchange accounts through API integration, allowing real-time updates of crucial cryptocurrency price data and transaction details.

II. CONCEPTUAL DATA MODEL

1. EER Diagram



2. UML Diagram



III. MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL

Primary Keys – PK, Foreign Keys - FK

User(User ID(PK),Name, Gender, Age, Email Address, Phone Number, Address, SSN(Unique))
- User ID is primary key, NOT NULL
- SSN is unique for user

Holds(User ID(FK),Cryptocurrency ID(FK))
- User ID and Cryptocurrency ID form a composite primary key where User ID is a foreign key referring User ID in User table and Cryptocurrency ID is a foreign key referring Cryptocurrency ID in Cryptocurrency table

Cryptocurrency(Cryptocurrency ID(PK),Cryptocurrency Name, Price, Volume, Market Cap, Daily High, Daily Low, Daily Change)
- Cryptocurrency ID is primary key, NOT NULL

Provides(User ID(FK),Exchange ID(FK))
- User ID and Exchange ID form a composite primary key where User ID is a foreign key referring User ID in User table and Exchange ID is a foreign key referring Exchange ID in Exchange table

Tracks(Cryptocurrency ID(FK), Watchlist ID(FK))
- Cryptocurrency ID and Watchlist ID form a composite primary key where Cryptocurrency ID is a foreign key referring Cryptocurrency ID in Cryptocurrency table and Watchlist ID is a foreign key referring Watchlist ID in Watchlist table

Listed_On(Cryptocurrency ID(FK),Exchange ID(FK))
- Cryptocurrency ID and Exchange ID form a composite primary key where Cryptocurrency ID is a foreign key referring Cryptocurrency ID in Cryptocurrency table and Exchange ID is a foreign key referring Exchange ID in Exchange table

Exchange(Exchange ID(PK),Exchange Name, User ID(FK))
- Exchange ID is primary key, NOT NULL
- User ID is a foreign key referring to User ID in User table, NOT NULL

Centralized_Exchange(Exchange ID(PK), Centralized API ID)
- Exchange ID acts as a foreign key referring to Exchange ID from Exchange Table

De-centralized_Exchange(Exchange ID(PK, De-centralized API ID))
- Exchange ID acts as a foreign key referring to Exchange ID from Exchange Table

Watchlist(Watchlist ID(PK),Portfolio ID(FK))
- Portfolio ID acts as a foreign key referring Portfolio ID in Portfolio table, NOT NULL

Transaction(Transaction ID(PK), Buy/Sell Price, Buy/Sell Quantity, Buy/Sell Date, Buy/Sell Time, User ID(FK), Portfolio ID(FK), Cryptocurrency ID(FK))
- Portfolio ID and Cryptocurrency ID act as foreign key referring to Portfolio and Cryptocurrency tables, both columns are NOT NULL

Reviews(Transaction ID(FK),Analysis ID(FK))
- Transaction ID and Analysis ID form a composite primary key where Transaction ID is a foreign key referring Transaction ID in Transaction table and Analysis ID is a foreign key referring Analysis ID in Analysis table

Portfolio(Portfolio ID(PK),User ID(FK))
- Portfolio ID is primary key, NOT NULL
- User ID acts as a foreign key referring to User ID in User table, NOT NULL

Alert(Alert ID(PK),Watchlist ID(FK),Daily Change)

- Alert ID is primary key, NOT NULL
- Watchlist ID acts as a foreign key referring to Watchlist ID in Watchlist table, NOT NULL

Analysis(Analysis ID(PK),Analysis Date, Analysis Time, Portfolio ID(FK))
- Analysis ID is a primary key, NOT NULL
- Portfolio ID acts as a foreign key referring to Portfolio ID in Portfolio table, NOT NULL

Risk Assessment(Risk Score, Risk Category, Assets at Risk, Analysis ID(FK))
- Analysis ID acts as a foreign key referring Analysis ID in Analysis table

Profit and Loss(Start Date, End Date, Current Value, Net Profit and Loss, Initial Investment, Percent Return, Analysis ID(FK))
- Analysis ID acts as a foreign key referring Analysis ID in Analysis table

Diversification(Concentration Percent, Asset Type, Diversification Score, Analysis ID(FK))
- Analysis ID acts as a foreign key referring Analysis ID in Analysis table

Tax(Taxable Event, Total Tax Due, Tax Savings Opportunities, Tax Bracket, Analysis ID(FK))
- Analysis ID acts as a foreign key referring Analysis ID in Analysis table

IV. IMPLEMENTATION OF RELATION MODEL VIA MYSQL AND NOSQL

A . MySQL Implementation:

The database was created in MySQL and the following queries were performed:

QUERY 1: Simple Query

To find all the users who all are from New Mexico

```
SELECT * FROM USER
WHERE Address = 'New Mexico';
```

UserID	Name	Gender	Age	EmailAddress	PhoneNumber	Address	SSN
3	User_3	Other	80	user3@email.com	185-213-7640	New Mexico	574-21-5969
13	User_13	Other	62	user13@email.com	295-336-1286	New Mexico	100-33-8657
15	User_15	Female	18	user15@email.com	521-623-9177	New Mexico	459-98-1385

QUERY 2: Aggregate Query

To find the total quantity of each type of transaction (like buy or sell) for every cryptocurrency

```
SELECT
   U.Name,
   T.Type,
   C.CryptocurrencyName,
   SUM(T.Quantity) AS TotalQuantity
FROM
   User U
      JOIN
   Transaction T ON U.UserID = T.UserID
      JOIN
   Cryptocurrency C ON T.CryptocurrencyID = C.CryptocurrencyID
GROUP BY U.Name , T.Type , C.CryptocurrencyName;
```

Name	Type	CryptocurrencyName	TotalQuantity
User_1	Buy	LINK	2.15
User_1	Buy	LSK	2.80
User_1	Buy	LEO	4.66
User_1	Buy	EOS	2.33
User_1	Sell	SOL	6.02
User_1	Buy	DCR	2.19
User_1	Buy	FIL	6.50
User_1	Buy	BNB	5.85
User_1	Sell	TRX	0.35
User_1	Buy	RUNE	9.15
User_1	Buy	BAT	8.37
User_1	Buy	SUSHI	1.90
User_1	Buy	ATOM	2.24
User_1	Buy	DAI	8.57
User_1	Sell	BNB	1.80
User_1	Buy	ETH	9.67
User_1	Buy	NANO	2.04
User_1	Sell	BAT	7.00
User_1	Buy	OMG	8.25
User_1	Sell	EOS	4.70
User_1	Buy	SNX	7.92
User_1	Buy	XTZ	9.10
User_2	Sell	SNX	8.18
User_2	Buy	XLM	1.62
User_2	Sell	HT	0.71
User_2	Sell	XTZ	6.83
User_2	Buy	EOS	2.03
User_2	Sell	ZRX	7.04
User_2	Sell	BNB	0.85
User_2	Sell	USDT	5.96
User_2	Buy	BNB	2.23
User_2	Buy	BAT	9.33
User_2	Sell	DOGE	0.95
User_2	Buy	DASH	6.26
User_2	Buy	XTZ	12.96
User_2	Buy	ALGO	4.14
User_2	Buy	BTC	7.28
User_3	Buy	BCH	1.48
User_3	Buy	DOT	0.17
User_3	Sell	HT	2.53
User_3	Buy	BTC	11.62
User_3	Buy	XRP	8.43
User_3	Buy	XTZ	18.17
User_3	Buy	USDT	4.67

QUERY 3: JOIN Query

To display the names of exchanges and the cryptocurrency 'EOS' is listed.

```
SELECT
   E.ExchangeName, C.CryptocurrencyName
FROM
   Exchange E
      JOIN
   Listed_On L ON E.ExchangeID = L.ExchangeID
      JOIN
   Cryptocurrency C ON L.CryptocurrencyID = C.CryptocurrencyID
WHERE C.CryptocurrencyName = 'EOS';
```

ExchangeName	CryptocurrencyName
Poloniex	EOS

QUERY 4: Nested Query

To find users who own cryptocurrencies priced higher than the average price of all cryptocurrencies

```
SELECT
   Name
FROM
   User
WHERE
   UserID IN (SELECT
         UserID
      FROM
         Cryptocurrency
      WHERE
         Price > (SELECT
               AVG(Price)
            FROM
               Cryptocurrency));
```

Name
User_1
User_2
User_3
User_4
User_5
User_6
User_7
User_8
User_9
User_10
User_11
User_12
User_13
User_14
User_15
User_16
User_17
User_18
User_19
User_20

QUERY 5: Correlated Query

To find users who are older than the average age of users sharing the same gender

```
SELECT
   U1.Name, U1.Age
FROM
   User U1
WHERE
   U1.Age > (SELECT
         AVG(U2.Age)
      FROM
         User U2
      WHERE
         U1.Gender = U2.Gender);
```

Name	Age
User_1	47
User_3	80
User_4	64
User_6	54
User_8	71
User_9	60
User_10	74
User_11	68
User_13	62
User_14	38
User_16	77
User_17	80
User_20	40

QUERY 6: >=ALL Query

To find all users whose age is greater than or equal to the average daily change of all cryptocurrencies

```
SELECT
    *
FROM
    User
WHERE
    Age >= ALL (SELECT
            AVG(DailyChange)
        FROM
            Cryptocurrency);
```

UserID	Name	Gender	Age	EmailAddress	PhoneNumber	Address	SSN
1	User_1	Female	47	user1@email.com	740-424-2595	Washington	204-82-4703
2	User_2	Male	57	user2@email.com	596-879-6107	Arizona	597-52-6924
3	User_3	Other	80	user3@email.com	185-213-7640	New Mexico	574-21-5969
4	User_4	Male	64	user4@email.com	293-989-5703	Wisconsin	129-69-4919
5	User_5	Other	36	user5@email.com	875-144-2061	Maine	704-31-6810
6	User_6	Female	54	user6@email.com	293-723-6118	Georgia	505-40-5276
7	User_7	Other	28	user7@email.com	896-682-5454	Michigan	613-46-5585
8	User_8	Other	71	user8@email.com	921-798-6415	Wisconsin	520-50-1934
9	User_9	Male	60	user9@email.com	198-823-2853	Hawaii	351-12-9017
10	User_10	Male	74	user10@email.com	514-383-5728	Iowa	699-73-5359
11	User_11	Other	68	user11@email.com	109-827-1014	Nevada	543-91-8038
12	User_12	Other	45	user12@email.com	830-403-6272	Delaware	384-11-9705
13	User_13	Other	62	user13@email.com	295-336-1286	New Mexico	100-33-8657
14	User_14	Female	38	user14@email.com	601-341-8761	South Dakota	378-65-7660
15	User_15	Female	18	user15@email.com	521-623-9177	New Mexico	459-98-1385
16	User_16	Male	77	user16@email.com	107-772-9701	Georgia	711-94-9133
17	User_17	Other	80	user17@email.com	179-379-7762	Massachusetts	973-91-8529
18	User_18	Male	18	user18@email.com	390-343-2306	Florida	842-41-5377
19	User_19	Female	29	user19@email.com	385-838-5755	Nebraska	316-50-1246
20	User_20	Female	40	user20@email.com	828-701-6286	Oregon	867-79-5910

QUERY 7: SET Operation (Union) Query:

To find a combined list of unique cryptocurrency names and exchange names, set operations

```
(SELECT
   CryptocurrencyName AS Name
FROM
   Cryptocurrency) UNION (SELECT
   ExchangeName AS Name
FROM
   Exchange);
```

Name
EOS
DAI
ATOM
QTUM
OMG
ALGO
BAT
AAVE
ZEC
XMR
SOL
SNX
NEO
DOT
LSK
BNB
NANO
UNI
RUNE
ETH
XRP
LINK
COMP
TRX
FIL
DASH
BTC
LTC
ADA
MIOTA
ZIL
XTZ
HT
WAVES
XLM
BCH
THETA
YFI

Query 8: Subqueries in SELECT

To find the count of cryptocurrencies for each user

```
SELECT
   UserID,
   (SELECT
         COUNT(*)
      FROM
         Cryptocurrency
      WHERE
         UserID = User.UserID) AS CryptoCount
FROM
   User;
```

UserID	CryptoCount
13	49
4	49
1	49
19	49
9	49
14	49
12	49
15	49
6	49
8	49
11	49
3	49
2	49
7	49
10	49
5	49
16	49
18	49
20	49
17	49

QUERY 9: Subqueries in FROM

To find the average price of cryptocurrencies listed on each exchange

```
SELECT
   E.ExchangeName, AVG(L.Price) AS AvgPrice
```

```
FROM
   Exchange E
      JOIN
   (SELECT
      L.ExchangeID, C.Price
   FROM
      Listed_On L
   JOIN Cryptocurrency C ON L.CryptocurrencyID = C.CryptocurrencyID) AS L ON
E.ExchangeID = L.ExchangeID
GROUP BY E.ExchangeName;
```

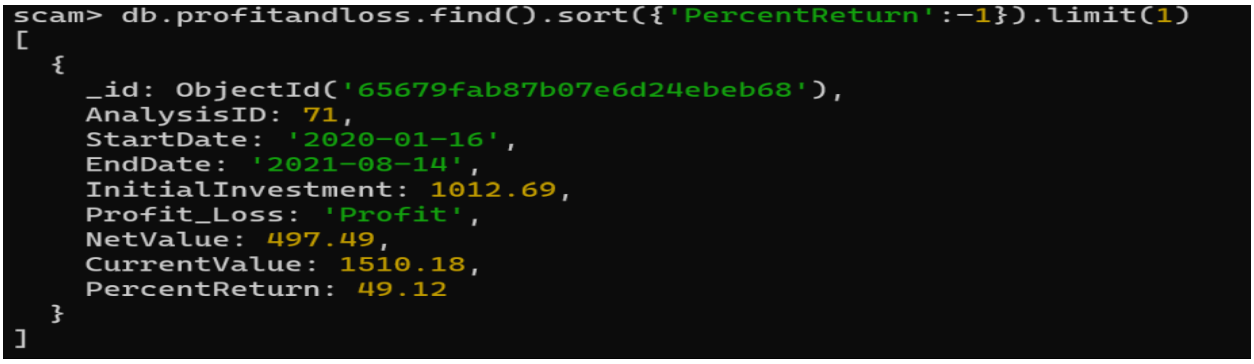
ExchangeName	AvgPrice
Poloniex	24244.264118
Binance	15286.912500
Huobi	35279.000000
Bitstamp	16622.545000
Bittrex	27037.050000
Bitfinex	23778.385000

B. NoSQL Implementation:

Query 1: Simple Query

Find the highest percent return from the ProfitandLoss table

db.profitandloss.find().sort({'PercentReturn':-1}).limit(1);



Query 2: Complex Query

Find the user and cryptocurrency details for the cryptocurrency Bitcoin ‘BTC’

db.holds.aggregate([{$lookup:{from:"user", localField:"UserID", foreignField:"UserID", as:"holdings"}},{$lookup:{from:"cryptocurrency", localField:"CryptocurrencyID", foreignField:"CryptocurrencyID",pipeline:[{$match:{CryptocurrencyName:'BTC'}}], as:"Crypto"}},{$match:{"Crypto":{$ne:[]}}}]);

Query 3: Aggregate Query

Find the number of transactions done by each user where total transactions greater than or equal to 25, order result by userid

db.transaction.aggregate([{$group:{_id:'$UserID',Total:{$sum:1}}},{$match:{Total:{$gte:25}}},{$sort:{_id:1}}]);

```
scam> db.transaction.aggregate([{$group:{_id:'$UserID',Total:{$sum:1}}},{$match:{Total:{$gte:25}}},{$sort:{_id:1}}])
[
  { _id: 5, Total: 29 },
  { _id: 7, Total: 26 },
  { _id: 8, Total: 26 },
  { _id: 9, Total: 31 },
  { _id: 10, Total: 31 },
  { _id: 13, Total: 30 },
  { _id: 14, Total: 30 },
  { _id: 16, Total: 26 },
  { _id: 18, Total: 35 },
  { _id: 20, Total: 29 }
]
```

Query 4: MapReduce Query

Count the number of transactions by each user using mapReduce function

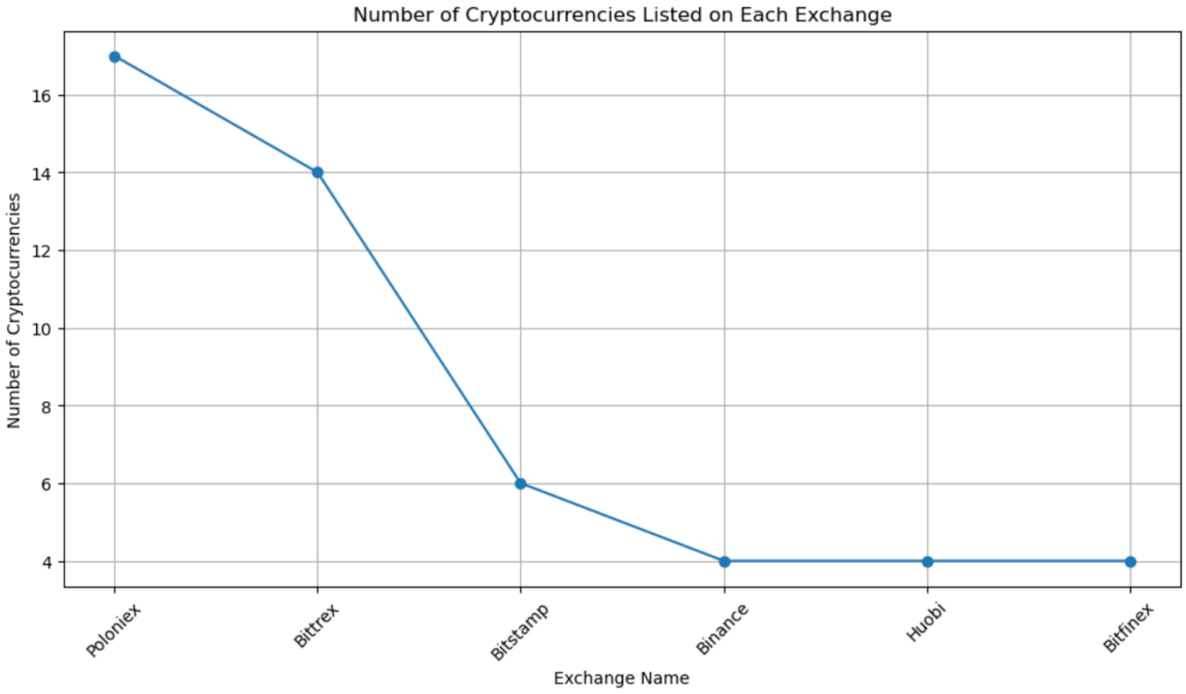
db.transaction.mapReduce(function(){emit(this.UserID, 1);},function(key,values){return Array.sum(values)},{out:{'inline':1}});

```
scam> db.transaction.mapReduce(function(){emit(this.UserID, 1);},function(key,values){return Array.sum(values)},{out:{'inline':1}}
);
{
  results: [
    { _id: 11, value: 22 }, { _id: 12, value: 23 },
    { _id: 3, value: 18 },  { _id: 6, value: 18 },
    { _id: 2, value: 20 },  { _id: 20, value: 29 },
    { _id: 18, value: 35 }, { _id: 1, value: 23 },
    { _id: 4, value: 20 },  { _id: 19, value: 24 },
    { _id: 14, value: 30 }, { _id: 17, value: 24 },
    { _id: 15, value: 15 }, { _id: 7, value: 26 },
    { _id: 8, value: 26 },  { _id: 10, value: 31 },
    { _id: 13, value: 30 }, { _id: 9, value: 31 },
    { _id: 16, value: 26 }, { _id: 5, value: 29 }
  ],
  ok: 1
}
```

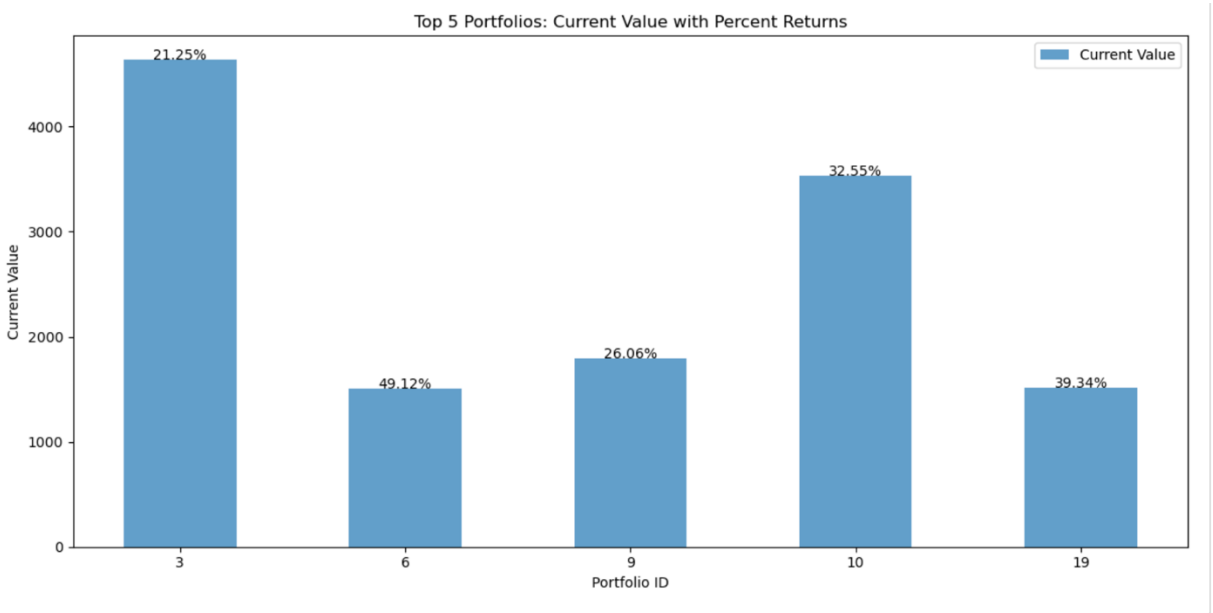
V. DATABASE ACCESS VIA PYTHON

The function fetch_data is crafted for data extraction from a database. This function initiates a connection to a MySQL database using a secure method, ensuring proper credentials are used for access. It efficiently handles SQL queries by utilizing a specialized command within the pandas library, which seamlessly converts the query outcomes into a structured data format, known as a DataFrame. In situations where the data retrieval encounters any issues, the function is equipped to identify these exceptions, providing a notification of the error, and returning a null response to indicate the unsuccessful operation.

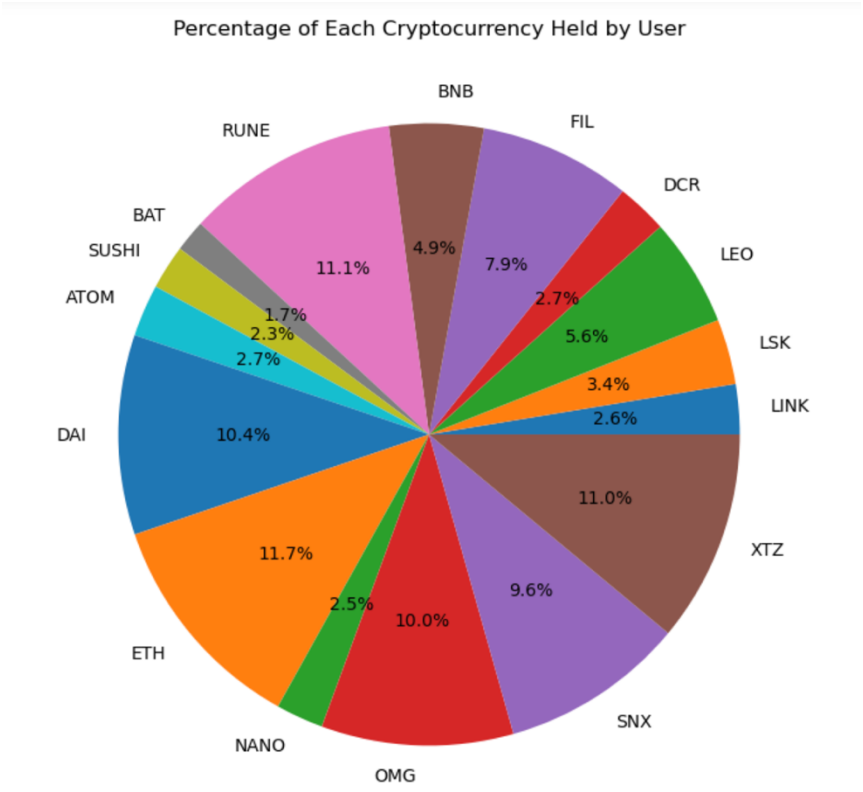
VISUALIZATION 1: To show how many cryptocurrencies are listed on Each Exchange



VISUALIZATION 2: To show the top 5 Portfolios that have made the most per cent returns



VISUALIZATION 3: To show the percentage of each cryptocurrency held by the UserID '1'



VI. SUMMARY AND RECOMMENDATION

Our project addresses the challenges and opportunities presented by the ever-evolving cryptocurrency trading landscape. As cryptocurrencies gain popularity, so does the need for effective risk management in this volatile market. Traditional investments provide security, but cryptocurrencies offer substantial returns at a higher risk. Recognizing these dynamics, our project aims to reduce the risks of cryptocurrency trading while also improving user safety. We designed a comprehensive database that caters to both centralized and decentralized viewpoints, ensuring a user-centric approach. The relational model, successfully captures the intricacies of cryptocurrency trading, providing a foundation for secure and efficient data management. The NoSQL implementation further enhances flexibility and scalability. Through extensive SQL and NoSQL queries, we demonstrated the database's capabilities in handling complex operations, offering valuable insights into user transactions, portfolio performance, and market trends. The integration of Python allows for advanced data analysis, facilitating risk assessments, tax analyses, and diversification evaluations.

Advantages:

- Comprehensive Risk Management: The project successfully addresses the high volatility and technical complexities associated with cryptocurrency trading, offering users a proactive approach to risk management.
- Database Efficiency: The relational model ensures data integrity, reduces redundancy, and enhances query performance. The integration of MySQL and NoSQL provides a versatile and scalable solution.
- User-Centric Approach: The inclusion of various analyses, real-time alerts, and portfolio insights caters to the diverse needs of users, empowering them to make informed investment decisions.
- Data Visualization: The project incorporates data visualization, making complex information more accessible and facilitating a better understanding of market trends and user-specific analyses.

Shortcomings:

- Limited Frontend Development: While the backend has been successfully implemented, the project lacks a fully functional frontend. Developing a user-friendly interface is crucial for a seamless user experience.
- Real-World Deployment: The project is currently in a developmental stage, and transitioning to real-world deployment requires additional steps, including setting up a front end and ensuring security measures are robust.

Recommendations:

- Complete Application Development: Prioritize the development of the frontend to create a user-friendly application. This will enhance accessibility and usability, making the platform more appealing to a wider audience.
- Security Measures: As the project progresses towards real-world deployment, ensure stringent security measures are in place to safeguard user data and transactions.

- User Testing: Conduct thorough user testing to identify any usability issues or areas of improvement. User feedback is invaluable in refining the platform for optimal performance and user satisfaction.
- Continuous Updates: Stay abreast of developments in the cryptocurrency market and technology to ensure the platform remains current and effective in addressing evolving challenges.

In conclusion, our project represents a significant step toward enhancing the safety and efficiency of cryptocurrency trading. It has the potential to be a valuable tool for investors navigating the complexities of the cryptocurrency landscape with further development and attention to user experience and security.