TABLE OF CONTENTS

1.0 Introduction

Cryptocurrencies are a digital asset treated as a **non-fiat currency**, i.e, not backed by any central authority or bank. Transactions are stored on a decentralised, distributed, "ledger". This technology is termed "the blockchain". Our project revolves around these concepts and technologies to build a **future-proof cryptocurrency** that is every bit as useful as the existing cryptocurrencies but also complies with security standards in ways that existing cryptocurrencies cannot match up to.

In brief, the software requirements of this project would be to build a blockchain from scratch along with all the supporting software to use the blockchain for transacting in cryptocurrencies, the most important of which would be a library of functions enabling the transactions on the blockchain to be **cryptographically signed** with **quantum-resistant signatures**.

Performance characteristics of our project would simply be the enabling of users to **transact on our blockchain** using our cryptocurrency. The beauty of building a sophisticated blockchain-based cryptocurrency and yet, a simplistic, intuitive user interface for interacting and transacting with it makes our product usable by nearly anyone capable of operating the average personal computer with a moderately fast internet connection.

Our users would include, in all essence, the **general public** in its entirety since cryptocurrencies are, after all, currencies in and of themselves. In addition to the general public, our user base would include **digital currency exchanges**, similar to stock exchanges, but meant for cryptocurrencies. Coinbase is a major such exchange operating out of the United States. The general public and these exchanges would require our product to support the reading and writing of transactions on the blockchain and also individual wallets for each user.

Transactions involving money implicitly require some form of **verification**. With digital currencies, such as our brownie points, we support this verification using **digital signatures**. However, unlike existing standard cryptocurrencies, our signatures are quantum resistant, i.e, they cannot be broken even by **quantum computers**, let alone regular computers. This quantum resistance is imbued into our product using the "**rainbow scheme**", an algorithm for the more general Unbalanced Oil and Vinegar scheme that uses multivariate cryptography. Regulatory bodies, such as the SEBI that regulates stock trades and exchanges in India, would be able to explore our product and it's features and data on a superficial level, maintaining sufficient anonymity such that an individual's identity is not easily compromised, but at the same time, their identity is unique enough that transactions can be carried out as effectively as transferring money from one bank account to another and also verified with equal impeccability.

1.1 Scope

The aim of this project is to implement **blockchain technology** for the creation and use of a cryptocurrency which we have termed "Brownie Points". Cryptocurrencies built on top of blockchain technology enable trustful yet pseudo-anonymous transactions wherein transactions are verified and trusted and defacing existing data is an insurmountable challenge. It transfers control from central authorities, such as banks and governments, to anyone and everyone. This is what gives such cryptocurrencies an innate quality of decentralized authority and secure transactions.

The novelty in this project involves the use of **quantum resistant signatures** for securing transactions and the use of secure wallets, which is a greatly underestimated quality who's importance only grows with development in the fields of quantum computing being increasingly progressive and the forever diminishing costs of all technology in general.

2.0 Product Perspective

Brownie Points are a **digital asset**, or, a cryptocurrency which can be used for making transactions without physical entities or a central governmental or bank-line authority to verify, validate, and store the currency and the transaction. This product has several parts that can be broadly described as a **web-based system** implementing a standard client-server architectural model for the wallet subsystem while implementing the primary ledger using blockchain technology.

The **ledger** of transactions is made available to its entire user-base via a separate web service. Blockchain inherently gives our product its long list of features which we further enhance by implementing functionality not typically supported by existing cryptocurrencies and their underlying technology.

2.1 User characteristics

While the general, and largest, part of our user base is the general public, our system is primarily considered with transaction handling and given this context, we have identified three primary, distinct, users.

- **Sender:** A user who wants to transact, i.e, send some brownie points from his account to another account. They should have a valid account termed "a wallet".
- **Receiver:** This user receives the brownie points sent from the sender's wallet. They should also have a valid account, also termed "a wallet".
- **Miner:** Miners, in brief, validate transactions. They solve a cryptographic puzzle to verify transactions and commit it to the blockchain. In return they receive a small transaction fee

It is assumed that the user would have basic knowledge of opening, operating, and interacting with a website on the the internet. The miners are expected to be familiar with the

working of a blockchain and should be able to solve the cryptographic puzzle (using the "nonce" parameter), given an API to interact with the blockchain. Also miners are expected to have decent computing capacity in their systems.

2.2 General Constraints, Assumptions and Dependencies

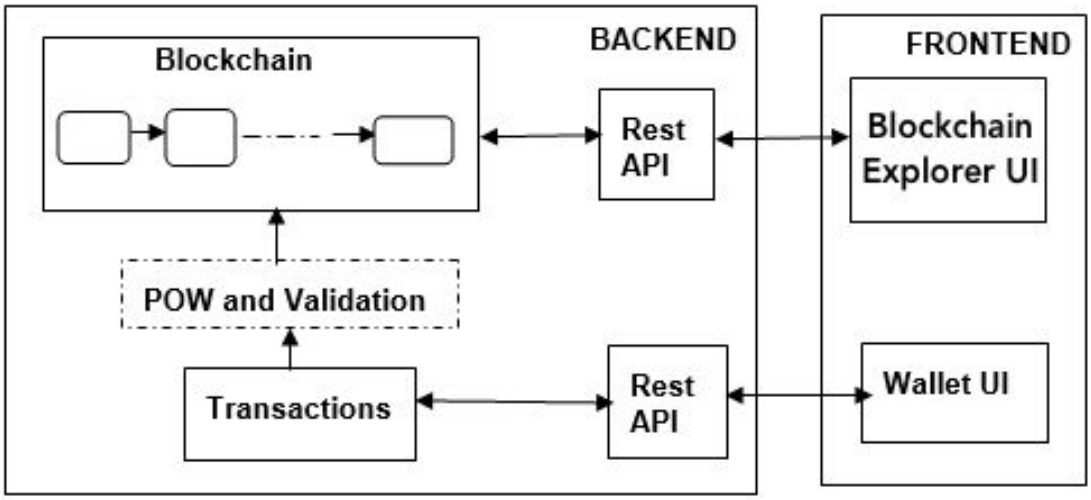
1. This product ships with web-based user interfaces only, requiring every user to have a web browser to access it.
2. The user's browser should be HTML5 compatible for a satisfactory user experience.
3. Every transaction takes approximately 4-5 minutes to get committed to the blockchain ledger.
4. Since the transaction pool is publicly available, the primary constraint is to make individual transactions inherently anonymous. Wallet addresses are not directly indicative of identity, however, given a wallet, it is easy to identify all the transactions committed to/from that wallet and hence, by that user.

2.3 Risks

- **51% vulnerability :** Blockchain relies on a distributed consensus mechanism to establish mutual trust. However, this mechanism reliance on a distributed consensus means that if an attacker were to take control of over 50% of the nodes on the blockchain, they could prevent new transactions from completing or even reverse existing transactions that occur while they were in control of the network. Generating new currency, however, would be almost impossible. Despite that, it is important to note that controlling over 51% of a sufficiently large network would be an arduous task, at best.

- **Private key security:** When using cryptocurrencies built on top of blockchain technology, the user's private key is deemed their identity and primary security credential, which is generated and maintained by the user making it their responsibility to store and maintain it, and any of its backups, securely.

3.0 System Architecture



Explanation:

- The customer is given a **wallet UI** where they can view their balance. The wallet also contains the cryptographic private and public key of the user with which he can make transactions.
- The **blockchain explorer** is a public tool that allows the user to view blockchain statistics and other information about individual blocks, addresses, transactions, and the transaction pool on the blockchain.
- **Rest API** – It is a type of application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.
- **Transactions** consists of a
    - Transaction Id: represented by taking a hash of the contents of the transaction.
    - Transaction input: contains the address from where the transaction is coming. It contains the digital signature of the sender which is signed using the public key.
    - Transaction output contains the address (public key) of the sender and the amount of coins sent. This can be unlocked by the receiver using private key.
- These transactions are public and broadcast to the entire **P2P network**, but only the person with the appropriate **private key** can obtain/access them. The P2P network of nodes will now validate the transaction using a set of known algorithms with the criteria that the format of these transactions should match and the transaction inputs, outputs and signatures should match. On validation of the transaction, it will be added to the existing blockchain.
- The **blockchain** consists of a chain of nodes. Each node has an index, timestamp, cryptographic hash of the contents of the block, previous hash value of the previous block and a nonce which is a counter used in the proof of work algorithm.

4.0 Requirements specifications

4.1 Frontend/UI

Reqmt #	Requirement
1	Easy to use UI.
2	User should be able to view his balance with the click of a button.
3	Simple interface to make transactions.
4	User should be able to explore the blockchain on the click of a button.

4.2 Wallet

Reqmt #	Requirement
1	Securing private key in wallet.
2	Notifications for security updates.

4.3 Transactions

Reqmt #	Requirement
1	Secure cryptographic algorithms for signing the transactions.
2	Public key cryptographic methods in which the public key can be derived from private key but not vice versa

4.4 Minimal Working Blockchain

Reqmt #	Requirement
1	Secure hash algorithms to hash contents of the block.
2	To add a block into the blockchain, there should be a use of a proof-of-work puzzle that is difficult to solve, but easy to verify.
3	Use of a distributed database to maintain blockchain.

5.0 External Interface Requirements

5.1 Hardware Requirements

- For running a node, the minimum requirements are:
  - A desktop/laptop with an OS.
  - 145 GB of disk space
  - 2GB RAM
  - Internet connection with upload speeds of at least 50 Kbps
- For mining, the hardware requirements are:
  - GPU or CPU with minimum of 3 GHz speed for better mining.
  - Power supply - minimum 150 Watt supply
  - Motherboard - with minimum of 1 PCI - E slot for the Graphics card.
- Customer perspective:
  - A desktop/laptop with an internet connection and with an OS that supports a web browser.

5.2 Software Requirements

- **Python** will be used as the programming language. Some advantages of using python over others include:
  - Availability of many built in packages
  - Simplicity of language
  - Good documentation
  - Reliable language
  - Inherently cross-platform

- **Django** will be used as the web framework. Benefits of using Django for this project include:
    - Security
    - The code base for the system is Python and Django offers good interaction with this code base.
    - Scalable, when there is heavy traffic demand.

- **Vue.js** will be used for building user interfaces. It is a progressive framework that provides highly optimized and responsive architecture. It's features such as flexibility, easy to use, detailed documentation help in developing a better web interface.
- **Multivariate Cryptosystems (Unbalanced oil and vinegar scheme)** - It is a secure public key cryptosystem in which the public key is a set of multivariate quadratic forms over a finite field. It has advantage of being faster and more secure when compared to RSA.

5.3 Communication Interfaces

- **REST API** is a web service communication protocol interface and is used as a communication interface between the client's UI and the server. REST API uses HTTP requests to GET, POST , PUT and DELETE data. The advantage of using REST API over other API's such as SOAP include:
    - REST allows a greater variety of data formats, whereas SOAP only allows XML.
    - REST offers better support for browser's clients.
    - REST provides superior performance, particularly through caching for information that's not altered and not dynamic.
    - REST is generally faster and uses less bandwidth.

6.0 User Interfaces

The user interface includes two major parts:

- **Wallet UI :** This provides the means to send and receive coins, view balance and check the transaction status.
- **Blockchain Explorer :** This is used to visualise the state of the blockchain. Information about the transactions added to the blockchain is available here, as well as a transaction pool with all the unconfirmed transactions. It is public and can be viewed by everyone, providing high transparency.

  The interface ensures easy and flexible interaction between the user and the system. The design provides affordances to improve user experience.

7.0 Performance Requirements

- The size limit of each block is 1MB.
- There is no upper limit for the length of the blockchain.
- A block is added to the blockchain every 10 minutes.
- Any number of miners can be involved to solve the proof of work problem.
- Only one miner will effectively add the block of a transaction to the blockchain.
- Any number of users can transact brownie points at the same time.
- The end software product should be able to work on cross platforms.
- Provide appropriate interface for all its users (Receiver, Sender and Miner)
- The end software product should be able to function reliably across networks (LAN, WAN etc).
- Nodes on the blockchain

8.0 Special Characteristics:

In order to make the blockchain more robust the following novel features will be incorporated. These features make our blockchain more secure compared to the existing ones.

- **Quantum resistant signatures** : With quantum computers becoming more popular, accessible and powerful, existing cryptographic techniques will quickly become expendable. Thankfully, quantum cryptography gives us algorithms that are resistant to both quantum computers and standard computers. Whenever a transaction is made, the sender must and should attach a digital signature created using his/her private key. The same is verified by the receiver using the sender's public key. We aim to provide for a tool that generates the key-pair required to produce a quantum resistant digital signature using Multivariate cryptography and in particular the Unbalanced Oil and Vinegar Scheme (UOV), also commonly known as the rainbow scheme. This fulfils the requirement of cryptographically secure signatures.
- **Secure way of storing the private key** : Most of the blockchain technologies store the private key in the wallets. Some cloud servers, web apps, desktop apps and mobile wallet software's offer to securely store the private key. But, here we entrust someone to secure our private keys. This increases the risk of our private key to be hacked. To overcome this, in our application we introduce the concept of not storing

the private key in the wallet but instead it will be the user's responsibility of storing the keys securely offline.

9.0 Help

For the detailed documentation, code-base, issues tracked, and any queries, please refer to the link: bit.ly/brownie-points

10.0 Other Requirements

10.1 Site Adaptation Requirements

Brownie Points will be supported on any platform that supports the python programming language and by extension, the python interpreter. It will ship as a software application; as an end to end solution for making transactions and mining currency on the blockchain.

Mining Brownie Points requires, at the minimum, an average personal computer with networking capabilities and a majority of the CPU load. Running a Brownie Points compatible wallet requires, at the minimum, a basic, entry-level personal computing device and a web browser to connect with it's privately hosted wallet backend.

10.2 Safety Requirements

The user should safely and securely store his private key (preferably offline) so that any funds associated with their account cannot be compromised by a security breach on a remote server. In addition to this, it is the user's responsibility to keep any backups of their key, if required, and ensure that the backup is also in a safe and secure location.

11.0 Packaging

The software will ship as two independant applications, one for enabling the machine it runs on to be a node on the underlying blockchain and one, intended for the end users, to act as a wallet applications to store and transact Brownie Points.

The software application will support all the encryption and safety mechanisms implemented by Brownie Points under the hood, i.e, quantum resistant signing and verification of transactions through proof-of-work.

12.0 Traceability Matrix

URS Reference Section No. and Name	CRS Reference Section No. and Name