A Transparent Marking System over a Distributed Ledger

SIT792: Minor Thesis

Edwin John Nadarajan

218599279

Team 2

Supervisor: Dr. Rolando Trujillo

Content

Abstract

This paper starts with the importance of examinations for education and knowledge gain and further proceeds into how online quizzing can benefit the efficiency of education. Then it describes 4 online quiz systems: 2 centralized and 2 decentralized, discussing their strengths and weaknesses. It goes on to describe systems and concepts relevant to improving the existing quiz systems like document integration into blockchain, reputation system and automated proctoring. The paper mentions 2 of the key questions/challenges that are being addressed and explained. How do the existing online quiz examination systems work and where do they all lack? How does the proposed system tackle the problems present in the existing system? And lastly, what will the proposed system look like and how will it work? The mid sections of the paper elaborate the methodologies and methods used in research; centering around quantitative and applied research. Blockchain based systems can resolve problems of non-transparency and tampering of results in the existing systems. However, blockchain isn't enough due to its own limitations. A community-based decentralized online quizzing system which boosts its experts can resolve all our issues. This paper proposes such a system called QuizzEth. It discusses the design and implementation of the system along with the technologies used in QuizzEth. Finally, the paper concludes with the key findings, screenshots of QuizzEth, link to its repository and steps to run it. Also, future work which can't be accomplished in the little amount of given time and needs further research is mentioned.

Introduction

Examinations are an integral part of any education system. It's very a common practice for schools, universities and institutes to assess students through examination. From the perspective of students, they can be very stressful and daunting at times, however it can be considered as a necessary evil. Examinations a more of a feedback of a student's capabilities and knowledge. They also bring improvement to a student's knowledge by informing them of their shortcomings.

Traditionally, exams are conducted offline using pen and paper or OMR based. [20] This system has been followed since a long time and has certain advantages like time management, lack of technical issues, hard to cheat, etc. But with the dawn of digitization, education systems have started adapting to the use of technologies for various purposes including examinations. Online examinations have rapidly started replacing the traditional pen and paper examinations. The benefits thereof lie not only in easier management but also reduces the cost that comes with pen and paper on both, money and environment. Studies show [5] that over 3 billion trees are cut down every year for the manufacture of paper! This indicates a massive blow to the environment. While it is true that people plant new trees and trees also multiply due to activities in the environment, the balance is slowly tipping, and we can already see the effects in the form of global warming.

Online examination systems, despite having many advantages, still have many flaws including the requirement to trust the professors or teachers who prepare the exam as the final say in marking goes to them. This is where blockchain comes in. Blockchain has unique features such as security, decentralization, reliability, immutability and data integrity which could possibly help improve the existing online examination systems of maybe even develop a new system.

Background

There can be different types of online exams. The most popular ones are online quizzes, which will be the focus of this paper. Online quizzing is shown to improve the learning rates and knowledge gain of students. Lorenzo, Antonio and Laura conducted a research [2] to analyze how online quizzes can be used as a teaching and assessment tool. They tested the effectiveness of online quizzes on the overall performance of students in 2 ways: when used solely as an assessment tool, and when integrated into the course as a combined strategy. The results were favorable and constructive. Quizzes strengthened the involvement of students in other activities (like group tasks, discussions, assessments, peer reviews etc). Quizzes have the added advantage of guiding students through the learning process with moderate and continual effort and have a proven positive influence on students' academic performance. A study conducted by John L. Dobson [4] determined that the use of quizzes can be associated with ameliorating summative exam scores. A retrospective analysis was performed on the course scores from three different groups of Exercise Physiology students. Students in group 1 completed the original version of the course, those in group 2 completed an updated version of the course that included more rigorous exam questions, and those in group 3 completed the same updated version of the course but with the addition of 10 required online quizzes. Results showed that the overall mean summative exam score from group 3 was significantly higher than that from group 2.

With the advancements in software and hardware technologies, many new systems were developed to achieve fair and flexible online quizzing for both, the instructors and the students. Given below are the summarized works of few people who have developed such systems.

Kazuaki Kajitori, Kunimasa Aoki and Sohei Ito, in their paper “Developing a Compact and Practical Online Quiz System” [7], developed a compact and practical online quiz system called QDB (Quiz DataBase) to satisfy needs in their classes. QDB has various features which enables it to make problems and quizzes and register them, auto-evaluation of some problems, printing of quiz, link generation to quiz, etc for the teachers and instructors. It also provides various capabilities for the students to do the online quizzes and exercises.

Frances Griffin and Ross Moore, in their paper “MacQTEX Randomised Quiz System” [6], developed a quiz system which provides online quizzes in PDF format for mathematics, having randomised question parameters which allow students to practice many examples of the same quiz. The quizzes are automatically marked and contain fully worked solutions which aid students in understanding the mathematical concepts involved. The staff interface contains a suite of tools which allows lecturers to monitor student progress and to create new quizzes.

Elena-Carmen Tentea and Valeriu Manuel Ionescu, in their paper “Online quiz implementation using blockchain technology for result tampering prevention” [12] showcased the custom build of a system that allows single sign in, creation of custom quizzes, solving quizzes and result tampering prevention.

Haojian Shen and Youan Xiao, in their paper “Research on Online Quiz Scheme Based on Double-layer Consortium Blockchain” [11] proposed an online quiz scheme based on Double-Layer Consortium Blockchain which helps to solve the problems associated with non-transparent scoring process.

Motivation

After a literature review on the existing systems for online quizzing, it was found that while there are systems based on blockchain which successfully solve some of the limitations of centralized online quizzing systems; making the answer checking and results tamperproof, they still suffer from the limitations of solely using

blockchain. In a blockchain, whenever a new block is added, all the nodes verify the new node and it is added only when more than 50 percentage of the participants have reached consensus. However, if a chain doesn't have enough participants, how will we ensure that the majority haven't been bribed? Therefore, the fewer the participants, the higher the chances of data compromise. Schools, universities and educational institutes, by themselves can have only 1000s of participants at most by themselves. The system must be standardized across all educational institutes to have a large chain.
For many people including students and teachers across all educational institutes to come together, what better way is there than forming an online community? When we look into platforms such as StackOverflow and Quora, we can see the flourishment of an online community to boost each other's knowledge. What if we bring online quizzing into such a community? Also, adding blockchain for tamper proofing results would further perfect the system. That's is where this thesis originates from. In this paper, such a system will be introduced. The scope for future research in this area still remains as there could be many improvements made when heading in this direction.

Problem Statement

Transparency and Decentralization of online quizzing is the main concern of this research project. The primary questions that need to be addressed are:

- How do the existing online quiz systems work and where do they all lack?
- How does the proposed system tackle the problems present in the existing system?
- What will the proposed system look like and how will it work?

To know about the existing online quiz systems, many papers were read through and a literature review was done. The gap discovered upon literature review was the lack of a mechanism by which large number of students and instructors across the world can form a community to share and test their knowledge online without worrying about the tampering of results, in other words, maintaining the transparency of the system.
A system called QuizzEth was proposed which blends in the features of community-based platforms such as StackOverflow and Quora with decentralized mechanism of blockchain into quizzing. This system was designed and then implemented in suitable technologies.

Related Work

Research in the field of online quizzing started even before the beginning of this millennia. In 1999, Macquarie University started the development of MacQTEX Randomised Quiz System [6]. MacQTEX was designed to provide mathematics online quizzes in PDF format with randomization of parameters; allowing the students to practice multiple exams of the same quiz. It was designed quite uniquely as in it doesn't need to communicate with the server until the completion of the quiz. All the interactions and score evaluation take place inside the document itself. A MacQTEX quiz requires a large download initially but the subsequent data transfers on quizzing are small. The quizzes heavily use the form functionality of PDF and enable interactivity using Javascript. Quizzes are automatically generated when provided with necessary information: questions, heading text, colour scheme and the file name. Despite the good number of merits of this system, there are a few limitations in it. This system can be used only for maths questions.

Kazuaki, Kunimasa and Sohei, in 2014, came up with a web based online quiz system called QDB [7] (Quiz DataBase). Just like MacQTEX, QDB is also for mathematics and statistics except that it uses the web browser as its interface and an apache web server enables communications. It uses MySQL database system to keep a record of all the problems, quizzes and results of conducted quizzes. It uses LaTex, Maxima and Perl for the

internal working and scoring of quizzes. In the client side, Javascript is used for some GUI interfaces. Teachers can register different types of quizzes consisting of problems or questionnaires into the system whereas the students usually take the online quiz or answer a questionnaire.

The use of these 2 systems are limited to mathematics. Also, in QDB, just like MacQTEX, once the server is hacked or a teacher chooses to be biased/unethical, the results can be tampered with. QDB is better than MacQTEX in the aspect of data consumption but it still has its own limitations.

So far, we've seen 2 quiz systems. Both suffer from the common limitation of result tampering due to centralized server. A decentralized approach can counter these limitations. Blockchain [10] is a decentralized immutable ledger a platform for people where they are able to completely trust the system, carry out the operations and not be worried about any tampering, their choice and their decision remains unaltered and thus cannot be manipulated in any way. Blockchain has a huge potential in education and many researches have been conducted to explore this and how it can be used to solve existing problems in education systems [8]. A system called "learning is earning" introduced by Sharples and Domingue allows students to get digital currency based on their learning outcomes which can be used for their tuition, learning resources and so on. Smart contract (an executable code stored in blockchain) can be used to enable trusted exchanges between multiple parties. In 2019, a systematic review was conducted on the applications of blockchain in education [9] according to which blockchain can bring many benefits into the education system such as: security, better control, enhanced accountability and transparency, enhanced trust, lower overall costs, authentication, data exchange efficiency, enhanced interactivity and so on.

In 2018, a research was conducted by Haojian Shen and Youan Xiao to solve some of the limitations of previous online quizzing systems. They proposed an online quiz scheme [11] based on double-layer consortium blockchain to make the scoring process transparent, just, trustworthy and tamperproof. The two layers consist of sub-chain which has many shards to store data and prime-chain which indexes the sub-chain; further reducing the load due to storage. This scheme achieves public verification of students' answer and the answer records are recorded in the consortium blockchain, which is public, unchangeable, and traceable. Group signature system is used for privacy and traceability of students.

In 2019, another quiz system based on blockchain [12] was proposed which allows SSO (single sign in), custom quiz creation, quiz solving and tamperproof results. The frontend of this system was implemented using Angular 6.0 framework involving Typescript, CSS and HTML. Backend used Firebase (Backend as a Service) and to make the results tamperproof, blockchain was used. Whenever a quiz is submitted, the quiz grade is added to the database and a new block is created to be joined into the blockchain which holds all the quiz related data, the hash of prior block, hash of the quiz data and previous hash, and the nounce.

These systems successfully solve the limitations of the prior systems; making the results tamperproof and disabling any injustice secretly done. However, they are still not enough due to both: limitations of blockchain as well as lack of other features. A blockchain network can only be successful if there are large number of participants in it. The fewer the participants, higher are the chances of data compromise. Universities, schools and institutes, by themselves form only a small network unless the system is standardized over everyone. Having an online quizzing community with a good growth mechanism can help fix this issue.

Though our problem seems to be limited to online quizzing system, it is much bigger than that. To get a more holistic view, let's look other ways of using blockchain in education as well as systems which can help improve the existing online quizzing systems.

Blockchain technology can be used as a practical solution [14] for validating, issuing and sharing of digital certificates. The digital certificates that we currently have can still be tampered with given enough computing capability. But if we use blockchain for this, the certificate, once uploaded, cannot be tampered with as blocks in blockchain are immutable by nature. Also, it stays there for a very long time and can always be located.

Another application of blockchain in education is dAppER [15], a decentralized application for examination review. Review and auditing of paper is a process wherein the examination papers must not be tampered with. There are multiple steps in Examination Review involving multiple groups of people. The system allows transparency that is ideal for External Examiners and Auditors to view. Since, examination paper must not be available to the public, dAppER uses permissioned blockchain called Hyperledger Fabric. Permissioned blockchains allow the user to control the nodes, only permissioned nodes can join and therefore the application becomes private. As for document integration, they are first converted into base64 format and then added into the blockchain as strings. In case of larger documents, hybrid approach like uploading document to the cloud and adding its digital fingerprint and location into the blockchain would work.

As seen from the above two applications, our quiz system could even accommodate certification for a course which involves quiz-based exams. And documents can be added into blockchain by using base64 conversion or a hybrid method involving cloud.

Earlier we discussed about our network being too small, in which case, the chances of data compromise becomes high due to the possibility of mass bribery. One method would be to expand the network. The main aim in having online quizzes or any other examination is for knowledge, and therefore, having a community for it is very helpful. This can be seen in platforms such as Quora and StackOverflow.

Stock Overflow is a question and answer platform for professional and aspiring programmers. This website acts as a platform for users to ask and answer questions, and with membership and active participation, vote questions and answers up or down and edit questions and answers in a style similar to Wiki or Reddit. An empirical study on developer interactions in StackOverflow [1] showed the importance of community in Q&A based platforms. In the same study, it was found that majority of developers only ask questions and not answer them (83.6%). For this cause, StackOverflow developed a reputation-based system. It tries to promote and credit the contributions of experts [3]. Reputation scheme of StackOverflow considers factors like answer is voted up, question is voted up, answer is accepted, question is voted down, answer is voted down, experienced Stack Exchange user onetime, accepted answer to bounty and offer bounty on question. If our online quiz system can accommodate this, the instructors will be more enthusiastic in making questions for quizzes. By using reputation scheme like StackOverflow and promoting more experts to join the network, we could the increase both the quality and the quantity of questions [17] and answers [16].

Another important factor we need to consider when building an online quizzing system is cheating. While it's true that the system can disable the examiners or anyone else from tampering the results, it cannot determine if the results are obtained by cheating in the first place. To counter this, Mohammad A Sarrayrih and Mohammed Ilyas proposed a system [18] that provides security to improve on-line examination by utilizing technologies such as biometric authentication, internet-firewall, cryptography, network protocol and object-oriented paradigms. The proposed system is useful when examinations are given at exam centers but not when they are given in a person's own home due to the possibility of cheating. For that, Yousef, Liping, Alex, Stephen, and Xiaoming proposed an automated online exam proctoring [19] that performs continuous and automatic online exam proctoring by capturing audio-visual stream from 2 webcams (on monitor & on user) and a microphone and

applying machine learning on the input. This would reduce the chances of cheating in an online examination environment with no officials for proctoring.

Research Methodology

This paper uses quantitative research wherein the key emphasis is on the collection of quantifiable data, summarizing them and drawing of inferences from those summaries. The purpose of research can be categorized as applied research; i.e. finding a solution for an intermediate problem faced by the society in education and online quizzing. In this paper 19 carefully selected research papers were read through in detail. Out of them, 4 papers give direct existing solutions to the problem statement and 2 of them almost solve the problem. The remaining 15 papers contribute to perfecting the ideal solution; adding bits and pieces to it. By this qualitative and applied research, the paper was able to draw out a picture which is considerate to many aspects that are needed for solving our problem.

Problem Analysis, Assumptions and Contributions

To tackle the given problem we need a system in which large number of students and instructors across the world can form a community to share and test their knowledge online without worrying about the tampering of results, i.e., maintaining the transparency of the system. Following is a fishbone analysis of our problem's solution:

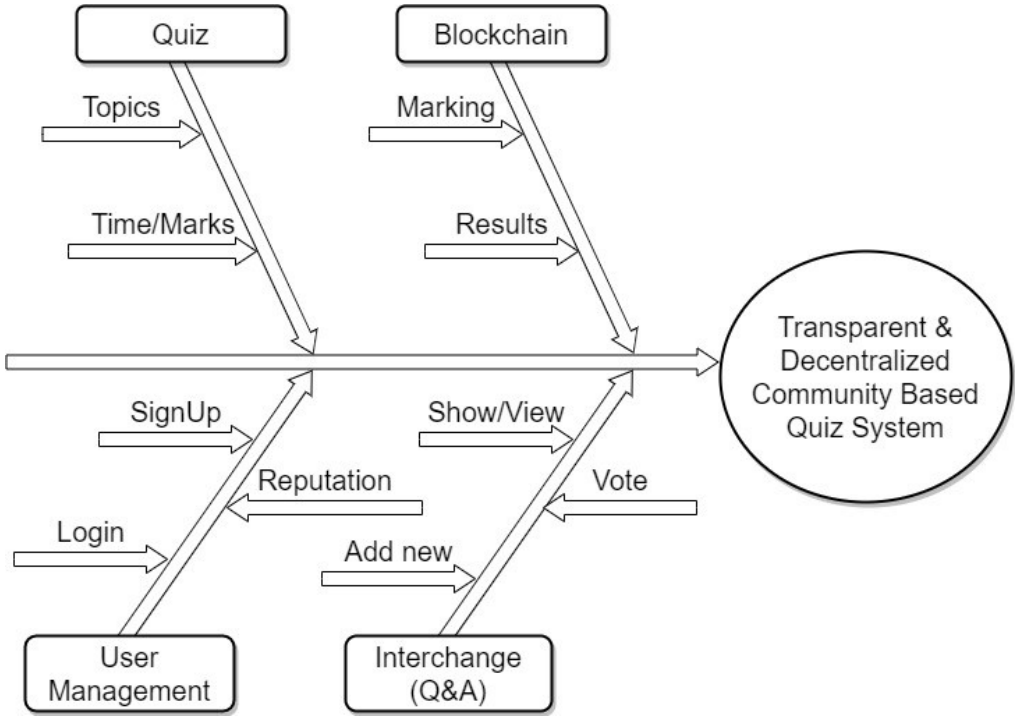

Fig: Fishbone diagram of problem/solution

For solving our problems, a system called QuizzEth is proposed, which is a solution based on the above fishbone diagram. QuizzEth is a community based decentralized online quizzing system which runs on MERN stack (Mongodb, Express, Reactjs and Nodejs) and Ethereum comprises of 3 major services:

1. User Management
2. Interchange Management
3. Quiz Management

User Management

People who use the system need a mechanism to use it as per their preferences with their identities. Most systems, user management is essential. A user needs to login to his/her account before using the system and logout after using it. If the user doesn't have an account, he/she must create one by the sign-up process. Since, our system uses blockchain, it requires the user to have a crypto-wallet and the address of the wallet is used in the registration.

Features and Stories:

1. Sign Up: If a user doesn't have an account in the system, he/she will use the signup process to register into the system. The signup will require the basic details of the user including his/her crypto-wallet address and password for the account.
2. Login: Once a user is registered, he/she may use the system for knowledge sharing and quizzing after logging into the system with his/her credentials. The login process also requires the address of the wallet. Typically, the system automatically gets it from metamask wallet.
3. Reputation: Whenever a user receives an upvote or downvote for an interchange, his/her reputation increases or decreases. A user's reputation also increases by getting more marks while doing a quiz.

Interchange Management

An interchange is basically a pair of 2 elements; in our case, a pair of question and answer. Knowledge can be represented in many ways, one of them being question and answer. To share knowledge, a user can add an interchange as a pair of question and answer while also giving options that are not correct. Once an interchange is added, it can be seen by all users of the system. Any user can view, upvote or downvote an interchange. A user may also search for certain knowledge based on tags/topics.

Features and Stories:

1. Add/Remove/Edit Interchange: A user can share his/her knowledge in the form of question and answer pair. He/she also needs to mention the incorrect options (for quizzing), tags and time difficulty (time required to solve the question). A user may edit or even remove the interchange that he/she added before.
2. View Interchange: All interchanges are public, along with their tags and time difficulties. Anyone can view, upvote or downvote an interchange. Optionally, a user may search for interchanges based on tags/topics (not implemented yet).

Quiz Management

The most important service of this system is quiz management. As seen in the literature review, quizzing positively influences a student's learning outcomes to a significant degree. Quizzing from time to time also helps in retaining information in our memory. Any user can start, submit and get results of a quiz based on topics and time required. The marking is done in blockchain so to avoid result tampering.

Features and Stories:

1. Start/Submit quiz: A user can add topics on which he/she wants a quiz, as well as the time for which he/she wants to do the quiz. Once these attributes are set, the quiz can be started. The system will select relevant questions that fit the time conditions and generate a quiz. After the user selects his/her options, the quiz can be submitted. Also, when the time runs out, the system will automatically submit the quiz after an alert.
2. Transparent marking: When a user starts a quiz and the system generates it, the quiz data is added into the blockchain. Once the user submits his/her answers, the answers aren't checked inside the system but rather sent into a smart contract in blockchain which holds the quiz data. The contract matches the user's answers with the correct answers to generate a tamper proof result and sends it to the system. The system then displays the result to the user and increases his/her reputation based on the marks.

Use Interaction Analysis

Actors:

1) User: A user can be an instructor or a student who uses the system for knowledge sharing and taking online quizzes.

Following is the use-case diagram for a user:

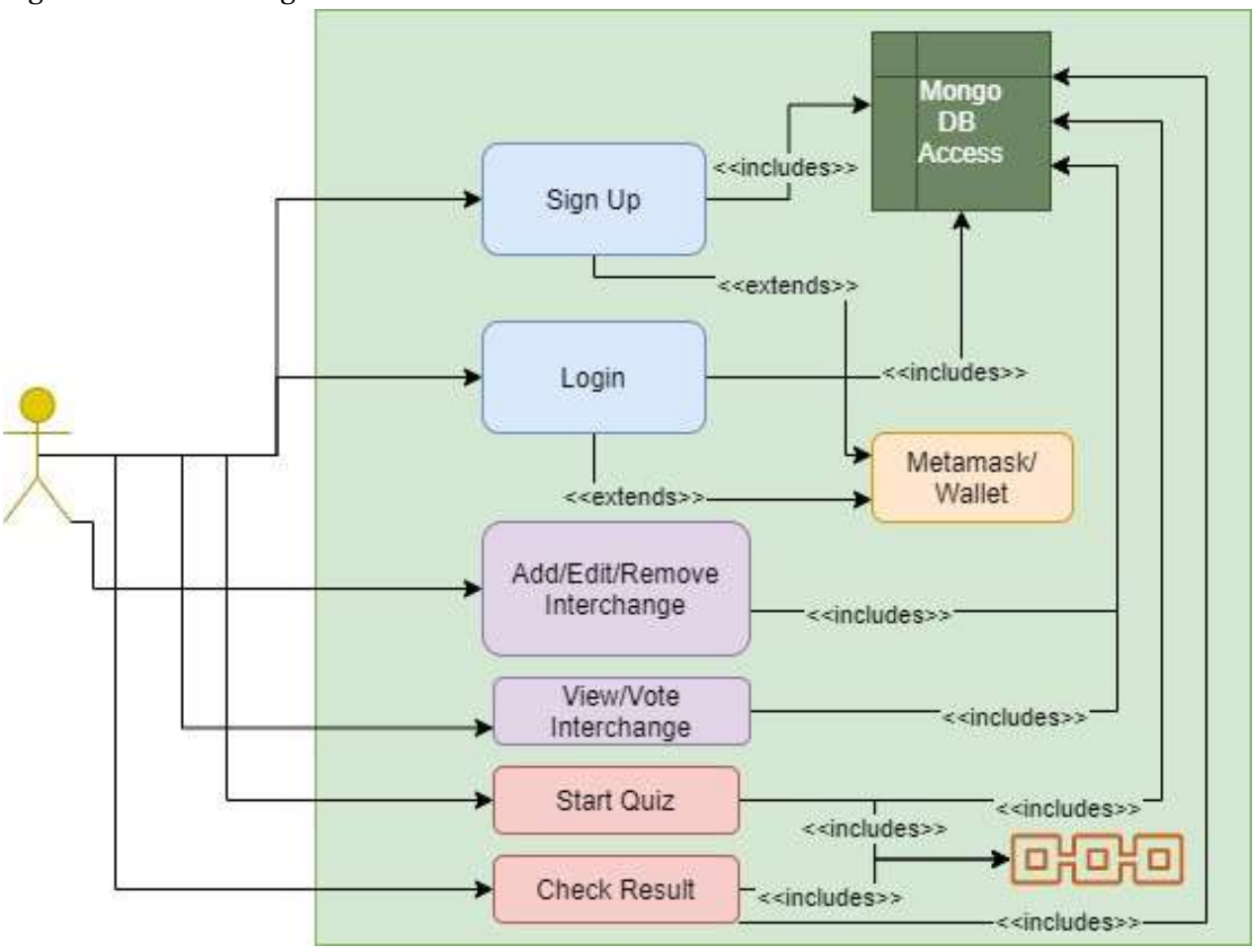

Fig 2: Use case diagram of typical user of QuizzEth

Since, QuizzEth is a DApp (Decentralized Application), users don't need to depend on system administrators or any other middleman and can directly interact with each other with the support of the blockchain.

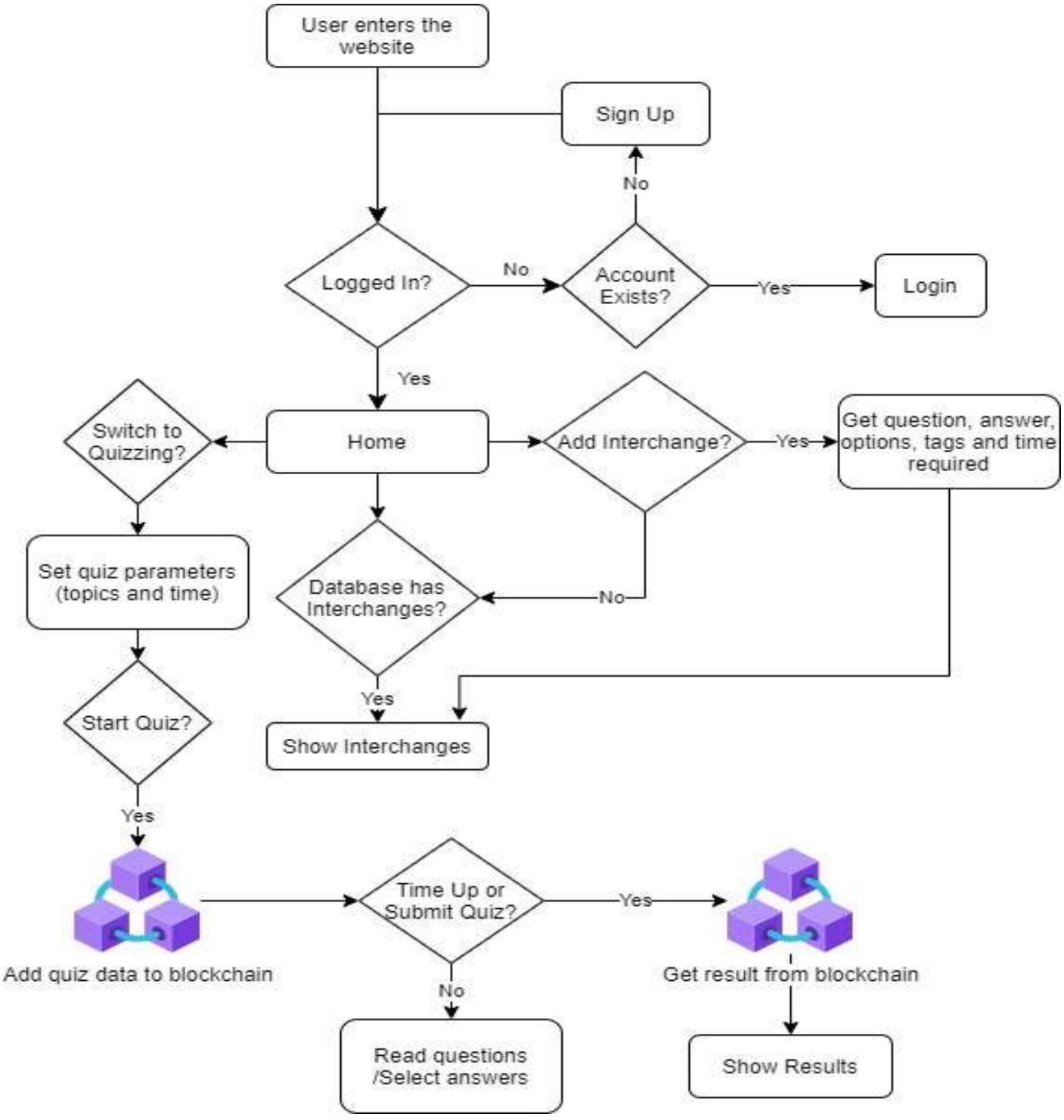

Fig 3: Data Flow Diagram of QuizzEth

Initially, if the user has already logged in, the home page is shown. Otherwise, the user has to login. If the user doesn't have an account, he/she can create one and then login. In the home page, the user can view all the

interchanges, edit or delete his/her interchange and add new interchanges. The user may also upvote/downvote any interchange. In the quiz tab, the user has to add topics of which he/she wants the quiz and set the amount of time he/she wants the quiz to be of and start the quiz. Th moment the quiz is started, questions are collected from database and sent to the blockchain's smart contract. A timer runs in the background while the user answers the questions. If the time is up or the user presses submit, the user's answers are sent to the smart contract which checks them and returns the result.

Proposed Product Backlog

Epics, Features and User stories

1) **User Management:**
    a. Login:
        i. Story: As a user, I want to be able to login using my credentials and use the website's quizzing services.
        ii. Acceptance Criteria:
            - Username, password and wallet address must be given.
            - Metamask wallet must be ready and connected.
            - Login button must be there which can start the process.
            - Option to signup in case user has no account.
        iii. Priority: 4/5
        iv. Effort: 6/10
    b. Sign Up:
        i. Story: As a user, I want to be able to make an account in the system if I don't have one already.
        ii. Acceptance Criteria:
            - Username (unique), password, repeated password, wallet address must be there.
            - Repeated password and password must match.
            - Signup button to register the user.
            - Option to go to login right after signup.
        iii. Priority: 4/5
        iv. Effort: 6/10
    c. Reputation:
        i. Story: As a user, I would want to have a reputation score which may increase or decrease based on my performance in quizzes as well as my contribution of knowledge in the community.
        ii. Acceptance Criteria:
            - A number stored internally and displayed with user information.
            - Increases with upvote on user's interchanges and quiz marks.
            - Decreases with downvotes on user's interchanges.
        iii. Priority: 4/5
        iv. Effort: 4/10
2) **Interchange Management:**
    a. Add/Edit/Remove Interchange:

i. Story: As a user, I would like to contribute knowledge to the community in the form of a Q&A interchange, edit to make corrections to my interchanges and remove my interchanges.
ii. Acceptance Criteria:
- Question and the correct answer.
- Options which would be incorrect. To be used in quizzes.
- Tags/topics for the interchange.
- Time required for a user to solve/answer the question. This also counts as the weight of the interchange.
- Add button to add the interchange into the system.
- Option: Edit or remove interchange (might change the reputation).

iii. Priority: 5/5
iv. Effort: 7/10

b. View Interchange:
i. Story: Any user would want to read through the interchanges to be able to gain knowledge.
ii. Acceptance Criteria:
- Show the question, answer, tags and time required.
- Show the number of votes.
- Buttons to upvote or downvote the interchange.
- Button to delete/edit the interchange (optional, initially).
- Search bar to search for interchanges based on keywords (optional).

iii. Priority: 5/5
iv. Effort: 7/10

3) Quiz Management:

a. Start/Submit a quiz:
i. Story: As a user, I would like to test my knowledge by taking quizzes based on certain topics. I must decide when to start and when to submit (unless time runs out) the quiz.
ii. Acceptance Criteria:
- Option to add topics based on which the quiz questions will be.
- Option to set the duration/total marks of the quiz.
- Option to start the quiz.
- Show the questions and the options of answers to choose from.
- Button to submit the answers.
- Show timer which shows the time remaining for the quiz.
- Automatically submit quiz when time runs out.

iii. Priority: 5/5
iv. Effort: 9/10

b. Transparent Marking:
i. Story: As a user, I would want the marking to be fair and transparent such that no one can tamper with the results.
ii. Acceptance Criteria:
- Smart Contract to store the quiz data (questions and correct answers).
- Smart Contract to receive user's answers, check them and submit the results.

iii. Priority: 5/5

iv. Effort: 10/10

Final Product Backlog Items (PBIs)

Product Backlog Items (PBIs)	Description of PBI	Estimate	Priority
Login	As a user, I want to be able to login using my credentials and use the website's quizzing services.	5	Low
Sign Up	As a user, I want to be able to make an account in the system if I don't have one already.	5	Low
Reputation	As a user, I would want to have a reputation score which may increase or decrease based on my performance in quizzes as well as my contribution of knowledge in the community.	3	Medium
Add/Edit/Remove Interchange	As a user, I would like to contribute knowledge to the community in the form of a Q&A interchange, edit to make corrections to my interchanges and remove my interchanges.	8	High
View Interchange	Any user would want to read through the interchanges to be able to gain knowledge.	8	High
Start/Submit Quiz	As a user, I would like to test my knowledge by taking quizzes based on certain topics. I must decide when to start and when to submit (unless time runs out) the quiz.	13	High
Transparent Marking	As a user, I would want the marking to be fair and transparent such that no one can tamper with the results.	21	High

Implementation

Using the designs shown before, I managed to implement an MVP (Minimal Viable Product) for this research product called QuizzEth. The following sections will mention the details of implementation including which technology was chosen and why, the algorithm used to select quiz questions and also the smart contract which was built for this system.

Blockchain Technology

There are many blockchain technologies available with different features. For our solution, we need blockchain technologies with support for smart contracts, which, as mentioned in the literature review, are executable code put in blockchain. The 2 most popular blockchain technologies which support smart contract are Ethereum and Hyperledger.

Hyperledger Fabric

Hyperledger Fabric is the open source enterprise-grade distributed ledger technology (DLT) platform that is mainly designed for its high-end use in the enterprise contexts and it would efficiently deliver the best key differentiating the capabilities on the distributed ledger in the high extensive manner. This platform mainly involved in offering the distributed ledger solution and its modular architecture is mainly involved with the resilient, flexible, scalable as well as confidential backend features.

Some advantages of Fabric are:

1) Modular Architecture: Modularity is encouraged in hyperledger fabric for the purpose of creating pluggable components.
2) Permission Membership: Permissioned blockchain enables data protection and privacy for health and finance aspects.
3) Trust, Performance and Scalability: Hyperledger Fabric focuses on reducing layers of trust with the varied number of verifications on the transactions. It is mainly suitable for making the transactions to easily increase its speed without any hassle.
4) Immutability: New file system mainly suitable for peer nodes that are immutable.

Some limitations of Fabric are:

1) Network fault tolerance
2) Minimum SDKs and APIs
3) Complex Architecture
4) Lack of proven use cases
5) Inadequately skilled programmers (my case)

Ethereum

Similar to bitcoin, Ethereum is an open source public blockchain network. Instead of miners mining for bitcoin, they mine for ethers, the cryptocurrency used in Ethereum. Ethereum's main feature is smart contract: a computer code on blockchain that can facilitate the exchange of money, content, property, shares, or anything of value. Because smart contracts are on blockchain, if they are well written, the possibility of corruption (defraud) is the least.

Some advantages of Ethereum are:

1) With the use of smart contracts, the involvement of 3rd party is reduced a lot.
2) It acts as a platform for other services. With time, as more developers enter and contribute, better information will be available for Ethereum.
3) There are a lot of companies which are trying to improve Ethereum beyond its foundation.

Some limitations of Ethereum are:

1) Because Ethereum is a platform, it will never be as effective as other chains that are designed specifically to be a cryptocurrency.
2) Due to major upgrades like changing from proof of work to proof of stake, the platform might not be stable, especially if the transition doesn't go smoothly.
3) Although tutorials are increasing now, it's still less as most of them are just about the basics.

In this research project, for implementing QuizzEth, Ethereum was chosen because we don't need a permissioned blockchain but instead, we want our chain to be public for increased transparency. Furthermore, there are more tutorials on Ethereum than on Fabric. The only limitation of Ethereum that might limit the implementation

would be the stability. However, for this paper, just the MVP is developed, and this won't be deployed into the mainnet.

Truffle

Truffle is a development environment for Ethereum which is very helpful for Ethereum developers. Developers can use to build, test and deploy DApps for testing purposes.

Features:

- Automated contract testing with Mocha and Chai.
- A configurable build pipeline that supports both web apps and console apps.
- Generators for creating new contracts and tests (like rails generate)
- Instant rebuilding of assets during development (truffle watch)
- Console to easily work with your compiled contracts (truffle console)
- Script runner that lets you run JS/Coffee files with your contracts included (truffle exec)
- Contract compilation and deployment using the RPC client of your choice.
- Support for JavaScript, CoffeeScript, SASS, ES6 and JSX built-in.

For QuizzEth, Truffle v5.1.48 was used.

Solidity

Solidity is a programming language developed by Christian, Alex, Liana and Yoichi for developing smart contracts for multiple blockchain technologies like Ethereum, Hyperledger, Monax etc. Solidity is compiled to bytecode (or portable code) that is executable on the Ethereum Virtual Machine (EVM), the runtime environment for smart contracts in Ethereum. It uses many concepts used by a modern-day programming language and is similar to Javascript.

Solidity version 0.6.6 was used to compile the contract written in version 0.6.0.

QuizzEth's Smart Contract

Here's the code of the smart contract used in QuizzEth. It's deployed at the start just once using truffle and doesn't need to be deployed for every quiz.

QuizMaster.js

```
pragma solidity ^0.6.0;
pragma experimental ABIEncoderV2;

contract QuizMaster{
    mapping(address=>uint) public time;
    mapping(address=>uint) public marks;
    address public system;
    mapping(address=>uint256) public quizCount;
    mapping(address=>uint) questionCount;
    mapping(address=>mapping(uint=>string)) public questions;
    mapping(address=>mapping(uint=>uint)) public weightage;
    mapping(address=>mapping(uint=>string)) answers;

    constructor() public{
```

```
        system = msg.sender;
    }
    function startQuiz(address _quizzer, uint _time, string[] memory _questions, string[] mem
ory _answers, uint[] memory _weights) public{
        require(msg.sender==system);
        time[_quizzer] = _time;
        marks[_quizzer] = 0;
        questionCount[_quizzer] = _answers.length-1;
        for(uint i=0;i<=questionCount[_quizzer];i++){
            questions[_quizzer][i] = _questions[i];
            answers[_quizzer][i] = _answers[i];
            weightage[_quizzer][i] = _weights[i];
        }
        quizCount[_quizzer]++;
    }
    function checkResult(address _quizzer, string[] memory _answers) public returns(uint, uin
t){
        require(msg.sender==system);
        marks[_quizzer] = 0;
        for(uint i=0;i<=questionCount[_quizzer];i++){
            if(keccak256(abi.encodePacked((answers[_quizzer][i]))) == keccak256(abi.encodePac
ked((_answers[i])))){
                marks[_quizzer]+=weightage[_quizzer][i];
            }
        }
        return (marks[_quizzer],time[_quizzer]);
    }
}
```

The constructor store's the deployer's address as system. The function "startQuiz" initiates a quiz for a user/quizzer with a given set of questions, answers and weightage. The function "checkResult" calculates the final score after comparing the answers of the quizzer to the actual answers.

Backend

In the backend, nodejs was used which utilized express for server and mongodb for database which are parts of the MERN stack. Also, web3js was used which is a library that helps connect the backend to the blockchain network.

Nodejs

It is an opensource, cross-platform backend environment based on javascript that executes javascript outside a browser, i.e. in the server side. It is event driven and is capable of asynchronous operations which are essential for interacting with blockchain. Nodejs features modularization and operates very well like that. I'm using Node version 12.14.1 in my development environment.

Here's the code of the main server that is run. It creates an express server, makes an active connection between the server and the mongo database, sets the routes which could be used by the frontend and finally listens at port 6000.

Server.js

```
const express = require('express')
const cors = require('cors')
const mongoose = require('mongoose')

require('dotenv').config()

const app = express()
const port = process.env.PORT || 6000

const uri = process.env.ATLAS_URI
mongoose.connect(uri,{useNewUrlParser:true, useCreateIndex: true, useUnifiedTopology: true })
.then((result)=>console.log(result)).catch((err)=>console.log)
const connection = mongoose.connection
connection.once('open',()=>{
    console.log("MongoDB connection successful!")
})

const usersRouter = require('./routes/users');
const interchangeRouter = require('./routes/interchange');
const quizRouter = require('./routes/quiz')

app.use(cors());
app.use(express.json());

app.use('/users', usersRouter);
app.use('/interchange', interchangeRouter);
app.use('/quiz', quizRouter)

app.listen(port, ()=>{
    console.log(`Server is running on port: ${port}`)
})
```

MongoDB

MongoDB is a cross-platform, document-oriented database program classified as NoSQL which uses JSON based documents and schema. It is a distributed database at its core which enhances its availability, ease of use and scaling. Version 0.3.6 of mongodb is used in this project.
A sample code from the project itself can be seen below. It's the schema of an interchange.

Interchange.modal.js

```
const mongoose = require('mongoose')
```

```
const Schema = mongoose.Schema

const interchangeSchema = new Schema({
    username: { type: String, required: true, minlength: 3, trim: true },
    question: { type: String, required: true },
    answer: { type: String, required: true },
    options: { type: [String], required: true},
    tags: {type: [String], required: false},
    votes: {type: Number, required: true},
    upvoters: {type: [String], required: false},
    downvoters: {type: [String], required: false},
    timeDifficulty: {type: Number, required: true},
    date: { type: Date, required: true }
},{
    timestamps: true
})

const Interchange = mongoose.model('Interchange', interchangeSchema)
module.exports = Interchange
```

Web3js

Digital assets such as cryptocurrencies (or programmable tokens) and smart contracts are central components of decentralized applications (DApps) that are used in blockchain. However, in order to interact with these on-chain components, transactions must be generated in blockchain. For a user or off-chain software to generate a transaction in blockchain, a node transaction must be relayed to the base peer-to-peer (P2B) network. Web3.js is a collection of libraries that allow programmers to interact with these on-chain components by facilitating connection to Ethereum nodes.
Here's the code of web3js in backend. The path '/' is for starting the quiz whereas '/result' is for checking the result of the quiz. Version 1.2.1 of Web3js is used in this project.

quiz.js

```
const router = require('express').Router();
const Web3 = require('web3');
const solc = require('solc');
const fs = require('fs');
var HDWalletProvider = require("truffle-hdwallet-provider");
var mnemonic = "under make verify bean lobster weapon jelly cost hungry evidence tiger parade
";
// For localhost
const web3 = new Web3(new HDWalletProvider(mnemonic, "https://rinkeby.infura.io/v3/cbfff34388
31426484f5b7bf33470"));
let QuizMaster;

fs.readFile('./abis/QuizMaster.json','utf-8',(err,jsonString)=>{
```

```
    QuizMaster = JSON.parse(jsonString);
 });


router.route('/').post(async (req,res)=>{
    const interchanges = req.body.interchanges;
    const address = req.body.address;
    console.log(interchanges);
    var totalTime = 0, weightage = [], questions = [], answers = [];
    interchanges.forEach(interchange => {
        totalTime+=interchange.timeDifficulty;
        weightage.push(interchange.timeDifficulty);
        questions.push(interchange.question);
        answers.push(interchange.answer);
    });
    let accounts, networkId, quizMaster, quizMasterNetwork;
    accounts = await web3.eth.getAccounts();
    networkId = await web3.eth.net.getId();
    quizMasterNetwork = QuizMaster.networks[networkId];
    if(quizMasterNetwork){
        quizMaster = new web3.eth.Contract(QuizMaster.abi, quizMasterNetwork.address, {from:
accounts[0], gas: '1000000', gasPrice: '100000000000'});
        await quizMaster.methods.startQuiz(address, totalTime, questions, answers, weightage)
        .send()
        .then(receipt=>{
            console.log(receipt);
            const startTime = new Date();
            const finishTime = new Date();
            finishTime.setMinutes(finishTime.getMinutes()+totalTime);
            console.log(startTime);
            console.log(finishTime);
            res.send({done:"Quiz started!", startTime, finishTime, totalTime});
        })
    }
    // res.send("Done")
});
router.route('/result').post(async (req,res)=>{
    var tmp = req.body.answers;
    var answers = []
    tmp.forEach(answer=>{
        if(answer)
            answers.push(answer);
        else
            answers.push("");
    })
    const address = req.body.address;
```

```
    console.log(answers);
    let accounts, networkId, quizMaster, quizMasterNetwork;
    accounts = await web3.eth.getAccounts();
    networkId = await web3.eth.net.getId();
    quizMasterNetwork = QuizMaster.networks[networkId];
    if(quizMasterNetwork){
        quizMaster = new web3.eth.Contract(QuizMaster.abi, quizMasterNetwork.address, {from:
accounts[0], gas: '1000000', gasPrice: '100000000000'});
        await quizMaster.methods.checkResult(address, answers)
        .send()
        .then(async receipt=>{
            await quizMaster.methods.marks(address).call()
            .then(async marks=>{
                console.log(marks);
                await quizMaster.methods.time(address).call()
                .then(totalMarks=>{
                    res.send({marks,totalMarks});
                })
            })
        })
    }
});

module.exports = router;
```

Frontend

The frontend uses Reactjs as its base and is equipped with web3js for connecting the user's wallet to QuizzEth. By using Reactjs, the webapp is split into multiple components which are rendered according to the design of the application.

Reactjs

React.js is an open source JavaScript library used to create user interfaces specifically for single page applications. It is used to handle the view layer for web and mobile applications. Reaction allows us to create reusable UI components. It allows developers to create large web applications that can transfer data without reloading the page. The main purpose of the reaction should be fast, scalable and simple. It only works on the user interfaces in use. It can be used with other JavaScript libraries or configurations such as Angle JS in MVC.

In QuizzEth, React was used to build many components such as Home, Login, SignUp, Quiz etc. One such component is shown below:

Home.js

```
import React from 'react';
import AddInterchange from './AddInterchange';
import ShowInterchanges from './ShowInterchanges';
import APIKit from '../APIKit'
```

```
class Home extends React.Component{
    state = {
        popup: false,
        interchanges: [],
        loaded: false,
        reputation: 0
    }
    loadData(){
        APIKit.get('/interchange')
        .then(response=>{
            console.log(response)
            this.setState({interchanges: response.data})
            this.setState({loaded: true})
        })
        .catch(err=>console.log(err));
        APIKit.patch('/users/reputation',{username:this.props.username, reputation: 0})
        .then(res=>this.setState({reputation: res.data.reputation}))
        .catch(err=>console.log(err))
    }
    componentDidMount(){
        this.loadData();
    }
    render(){
        return(
            <div>
                Welcome, {this.props.username}, rep: {this.state.reputation}
                {this.props.username?<AddInterchange refresh={()=>this.loadData()} closePopup
={()=>this.setState({popup:false})} popup={this.state.popup} username={this.props.username}/>
:null}
                <button onClick={()=>this.setState({popup: true})}>Add Interchange</button>
                {this.state.loaded?<ShowInterchanges updateRep = {(rep)=>this.setState({reput
ation:rep})} refresh={()=>this.loadData()} username={this.props.username} interchanges={this.
state.interchanges}/>:null}
            </div>
        )
    }
}

export default Home
```

Metamask

Metamask is a cryptocurrency wallet that can be used in Chrome, Firefox and bold browsers. This is a browser extension. This means that it acts as a bridge between normal browsers and the Ethereum blockchain. Ethereum

blockchain is a network where users can create their own applications (they are called dApps) and cryptocurrencies.
This research project requires the user to have metamask extension enabled in their browser. It's available in most popular browsers like Chrome, Firefox and Edge. QuizzEth uses metamask to connect the users easily to the blockchain without the need for them to download a wallet application which usually comes with a massive chain data (of blockchain).
It requires a very small piece of code for the frontend to connect to a browser's metamask extension:

```
async loadWeb3(){
  let web3;
  if(window.ethereum){
      web3 = new Web3(window.ethereum);
      await window.ethereum.enable();
  }
  else if(window.web3)
      web3 = new Web3(window.web3.currentProvider);
  this.setState({web3});
}
```

The webapp waits for the user to give it permission to access metamask. Once this is done, the user can go ahead with using QuizzEth.

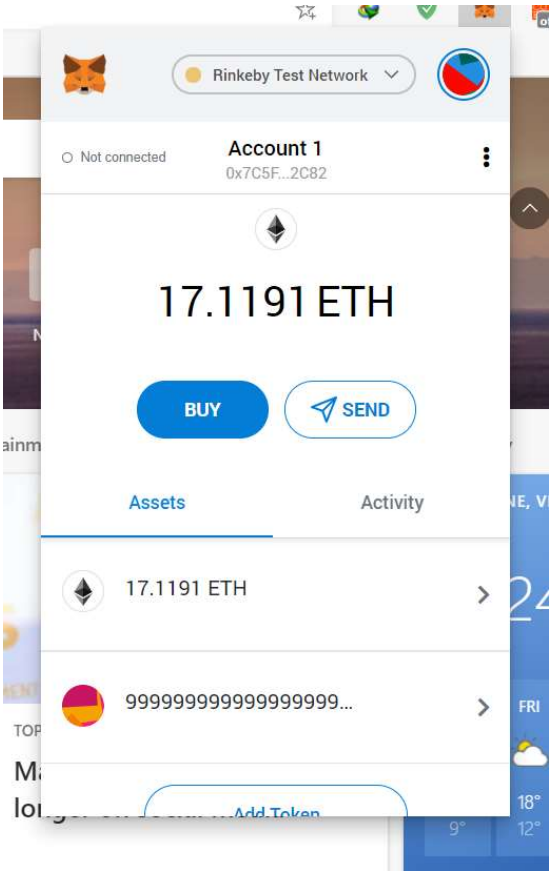

Fig: Metamask browser extension

Results and Discussion

Importance of examinations is already evident in the society. By including quizzes in the course path, a significant improvement in students' learning outcome has been observed. To incorporate the several benefits of quizzing, many online quizzing systems were developed. Most of them use centralized databases which are prone to corruption/compromise either by an instructor's bias or an external agent such as hacker. To make the marking fair, blockchain was used for preventing tampering of results and providing transparency. However, blockchain itself comes with its own limitations, mainly in terms of number of participants which needs to be high. To solve this issue, this paper came up with a community-based quizzing system based on blockchain wherein, anyone can contribute knowledge in the form of an interchange (pair of question and answer). Good knowledge will receive more upvotes; increasing the reputation of the individual who posted it whereas false knowledge will receive downvotes; decreasing the reputation of the user. This will encourage the users to contribute with integrity and to put their best efforts out there. Also, the system collects such interchanges based on a user's topic and time requirements to generate a quiz with tamper-proof results. Better result also increases the reputation of the user. This mechanism of combining reputation, community, knowledge and blockchain together seems to work out very well for students as well as instructors. Although there are still limitations such as lack of proctoring in official exams and certified storage of results, they are still problems that can be tackled in time with a bit more research. Also, at the moment, I have not included the fee for every quiz. Whenever a student gives a quiz, a small fee must be collected from him/her which could account for putting the quiz information into the blockchain and processing the result as well as incentivizing the owner of the system and those people who contributed to the knowledge of the community. Further discussions about this will be done in the future research section.

Here are a few screenshots of QuizzEth in action:

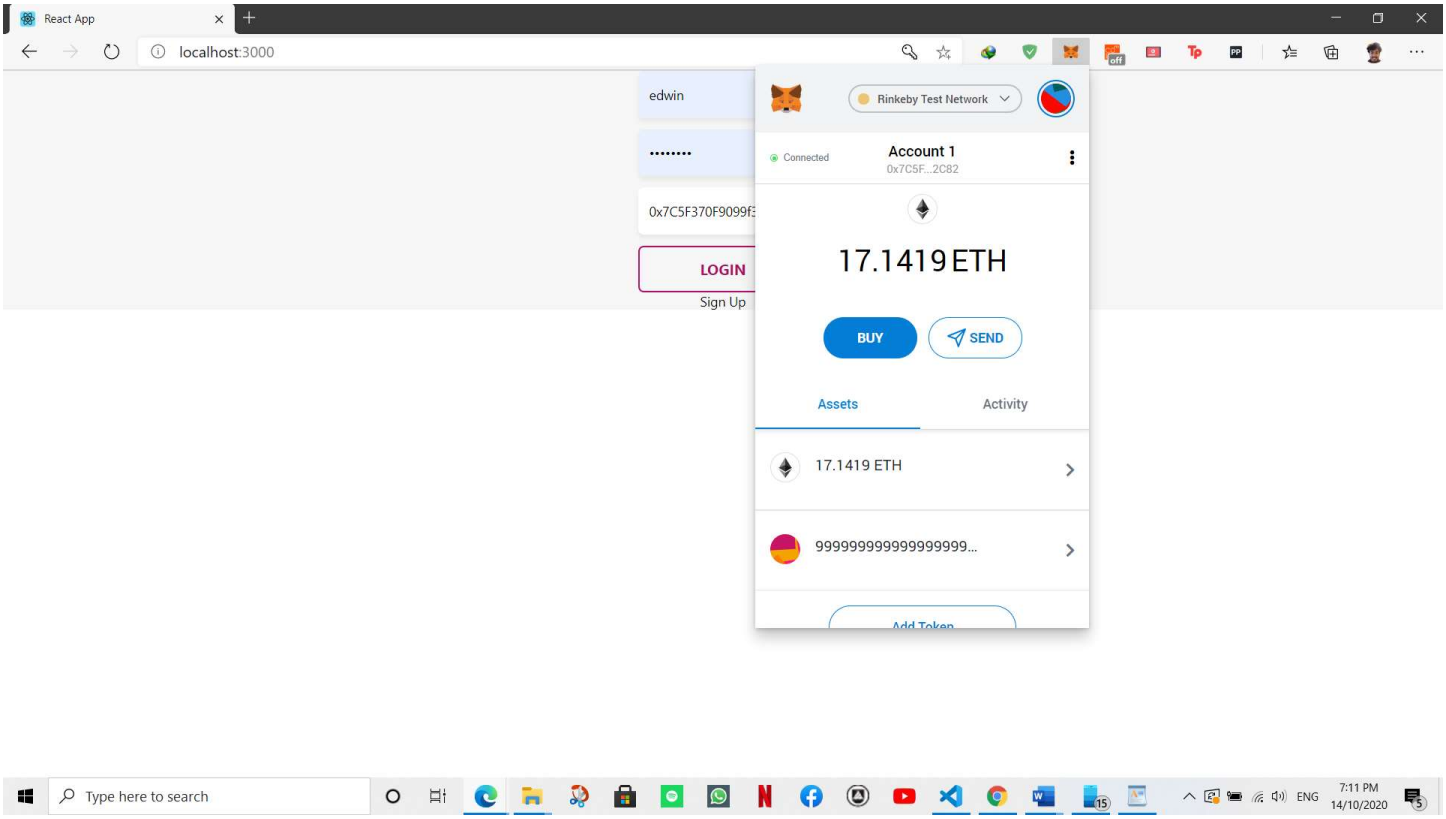

Fig 4: Login Screen and connection to metamask

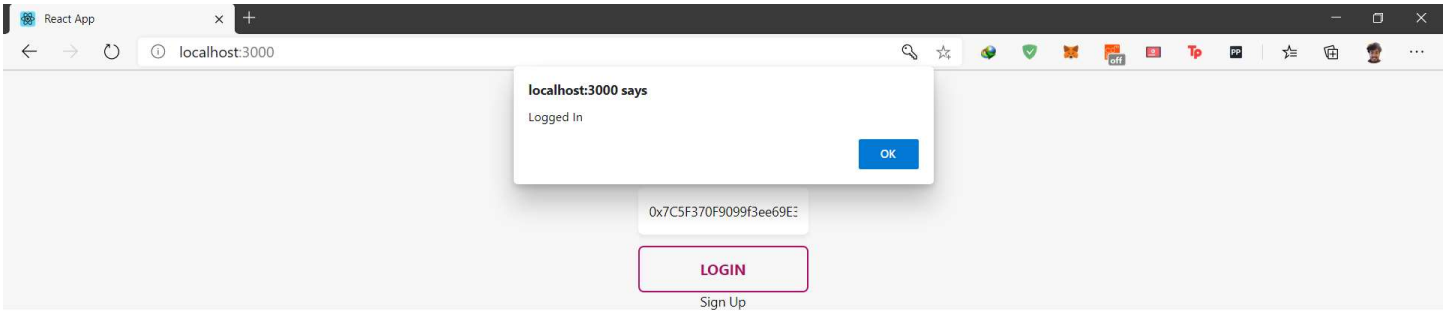

Fig 5: Successful Login

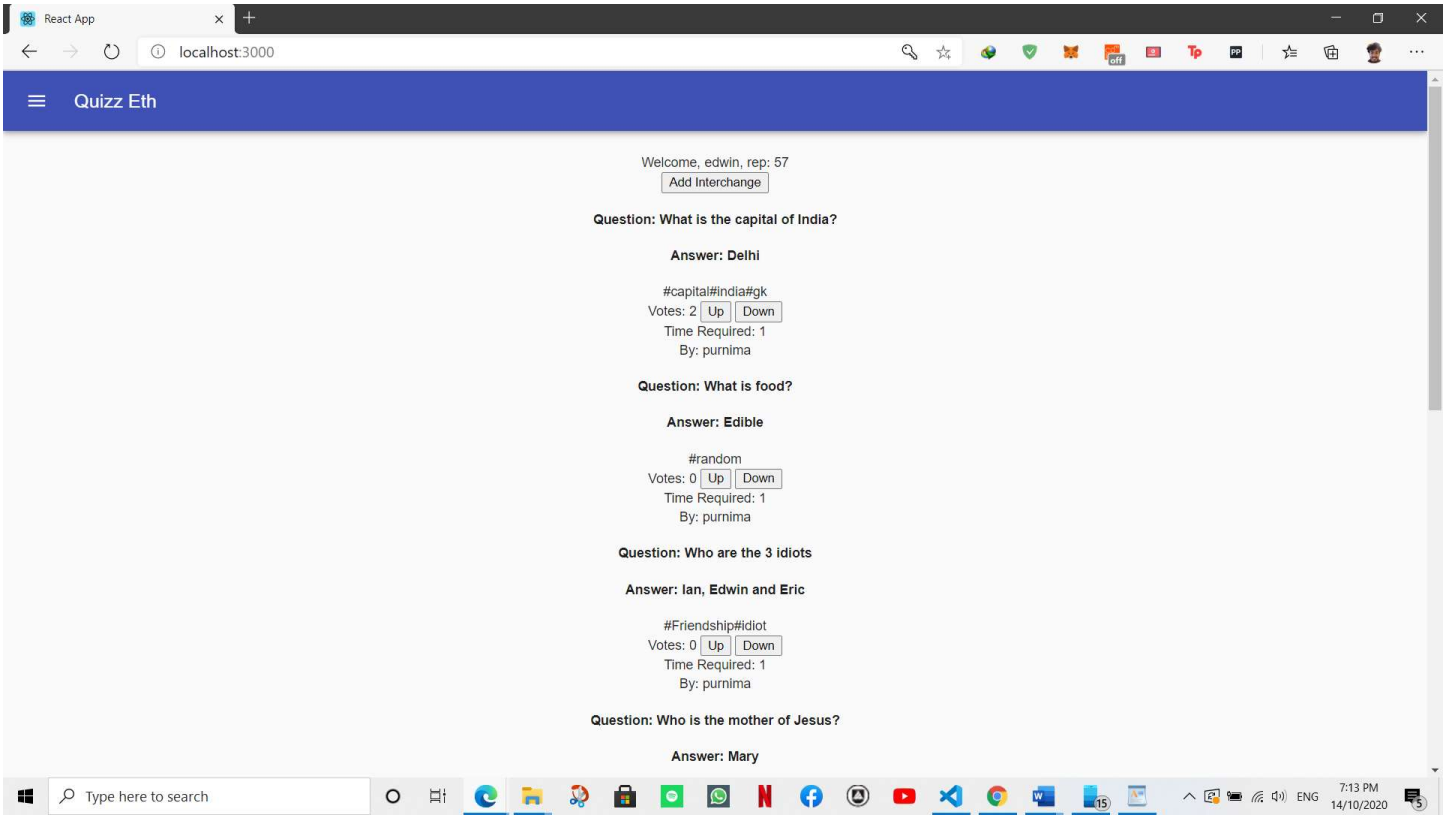

Fig 6: Home Screen with options to add interchange, upvote and downvote interchange

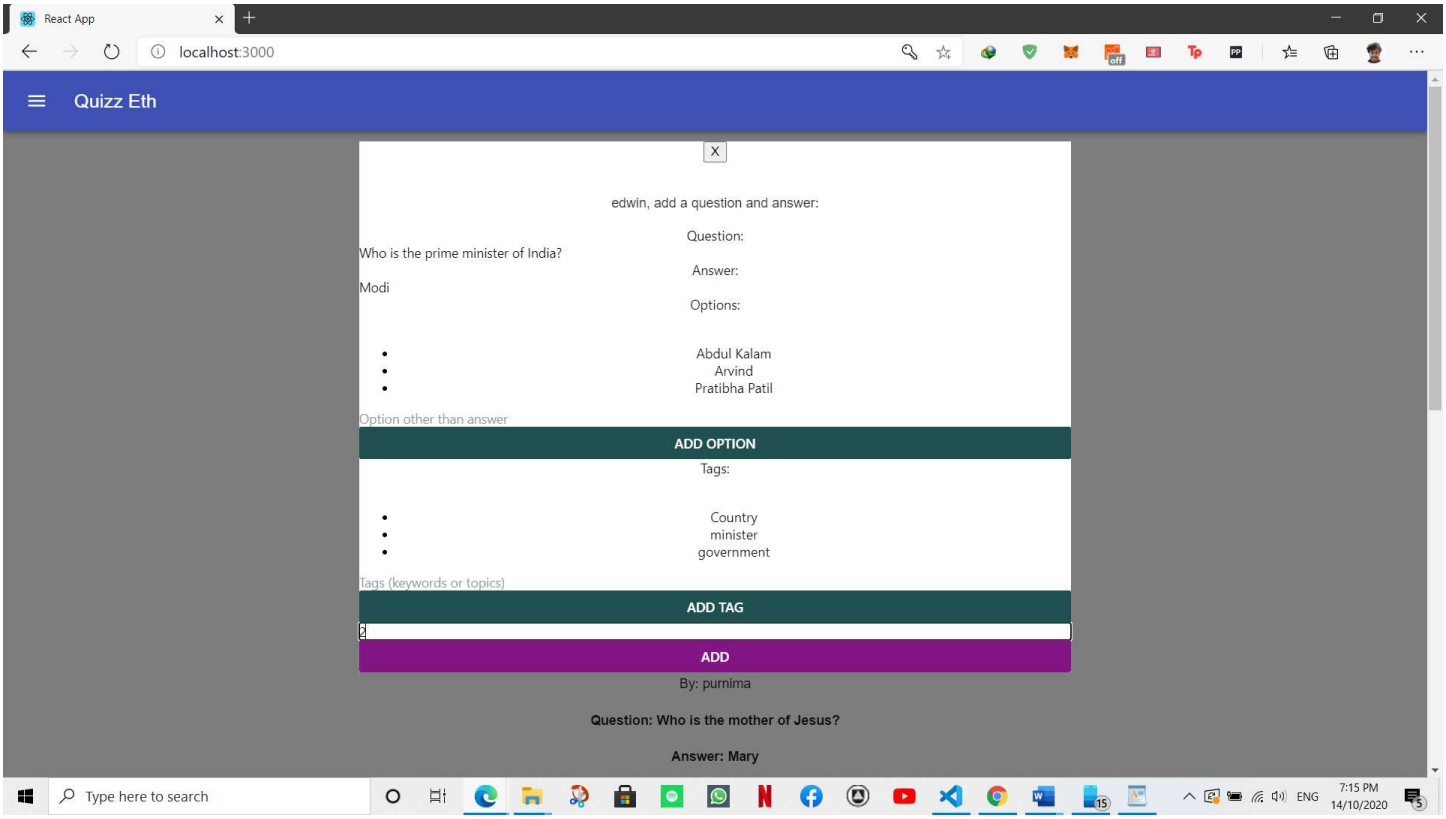

Fig 7: Adding an interchange into the system

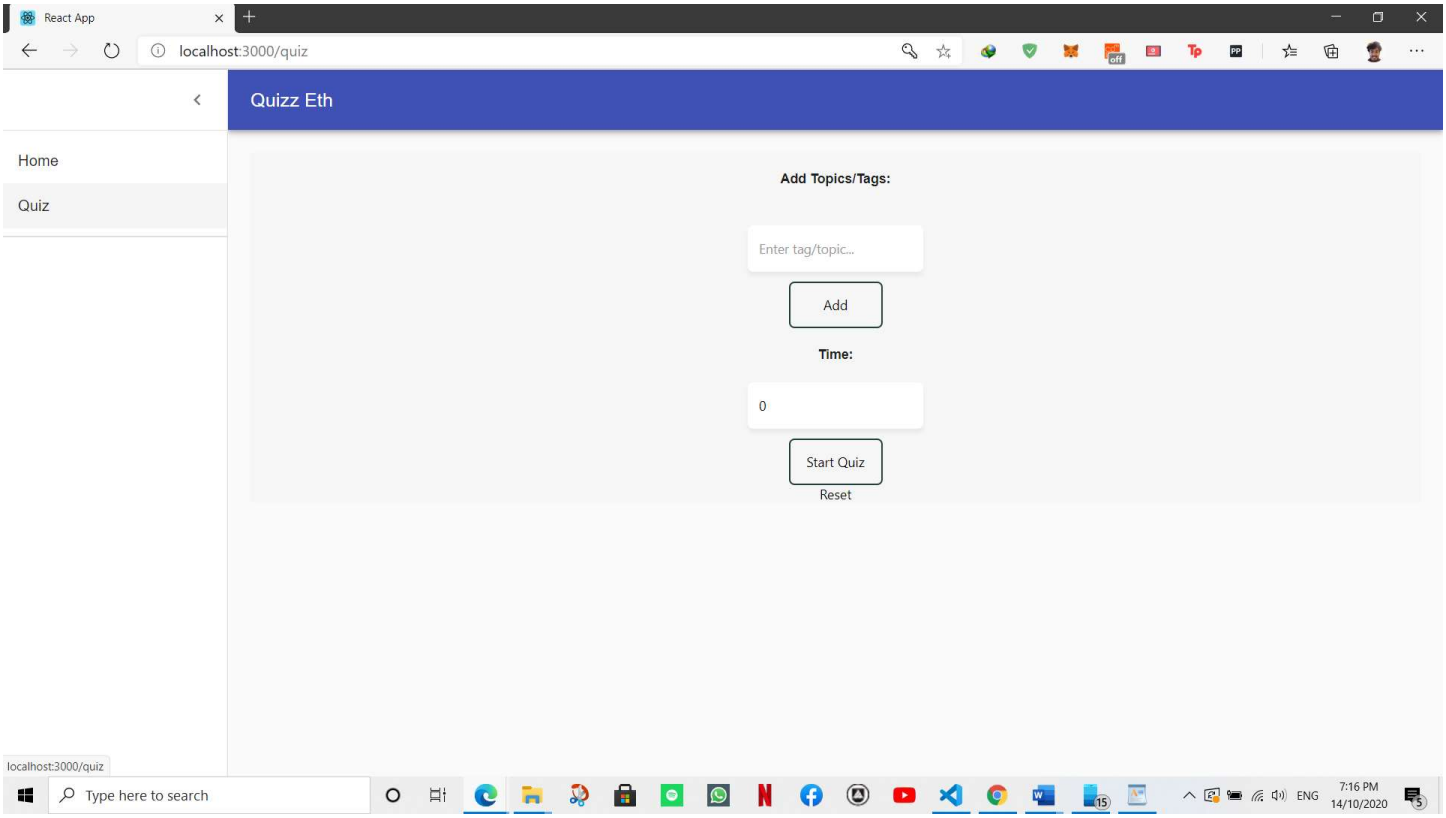

Fig 8: Switching to quiz section.

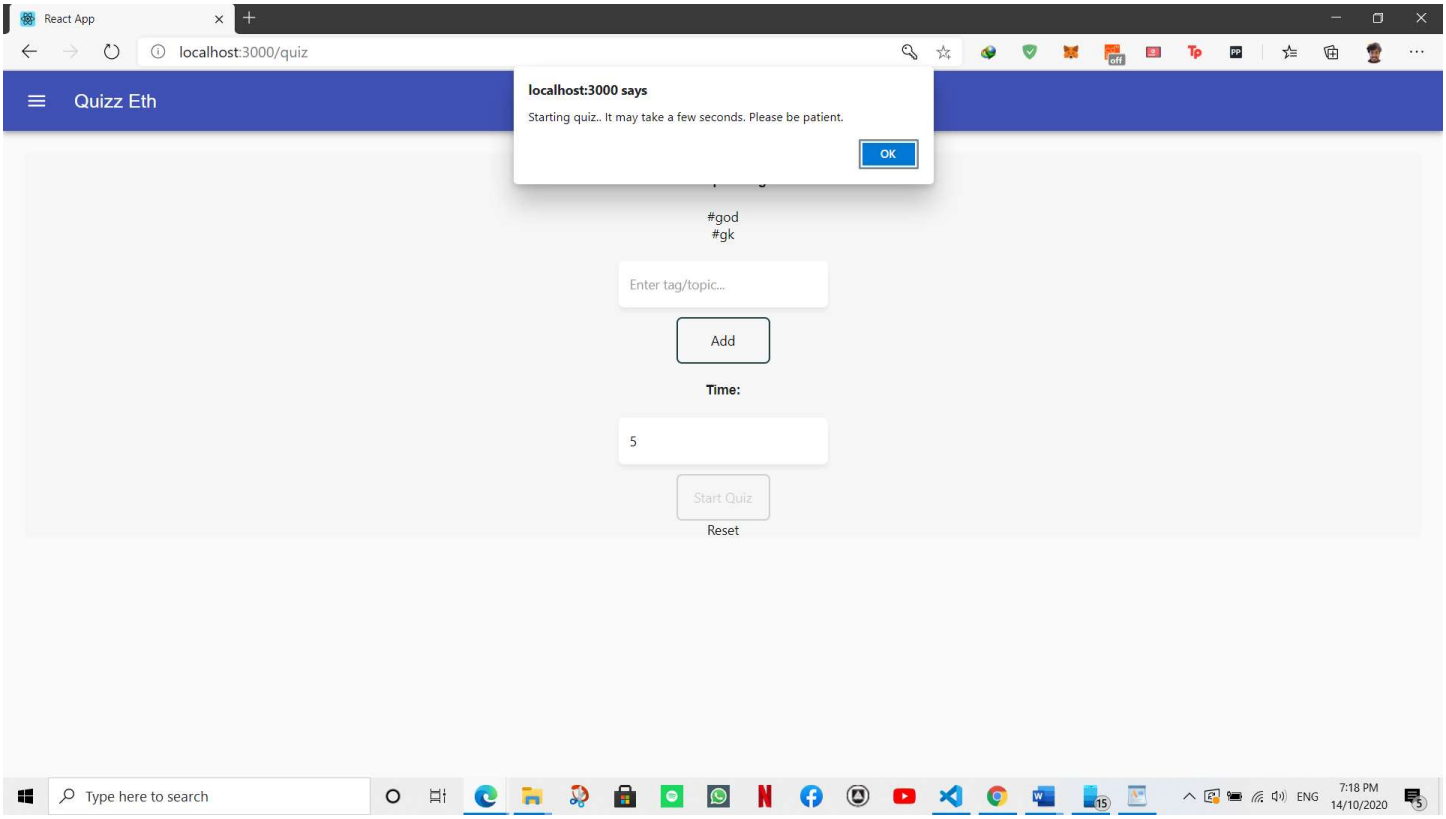

Fig 9: Starting a new quiz of 5 minutes based on 2 topics

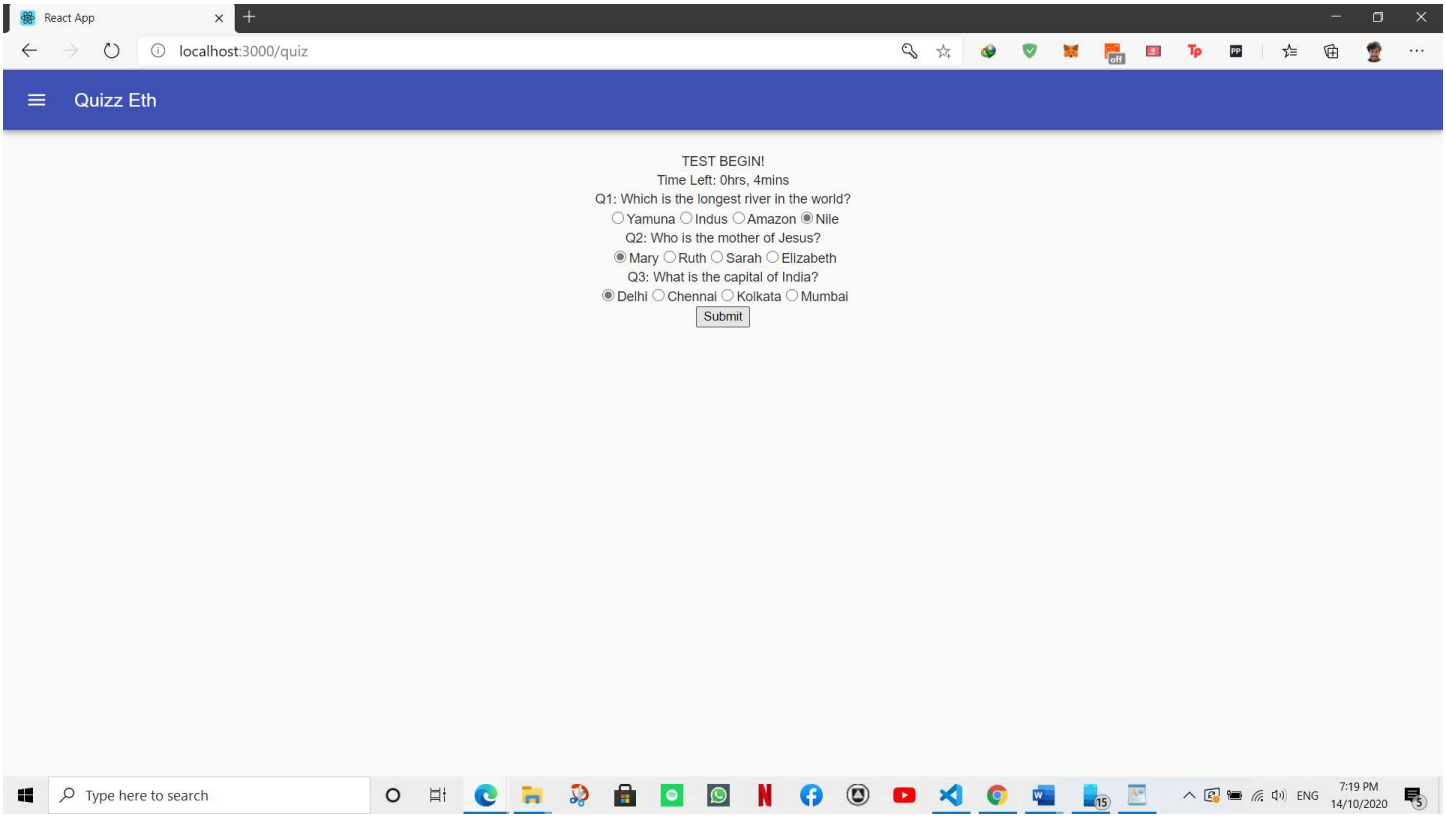

Fig 10: Quiz started, 1 min elapsed, 4 mins left.

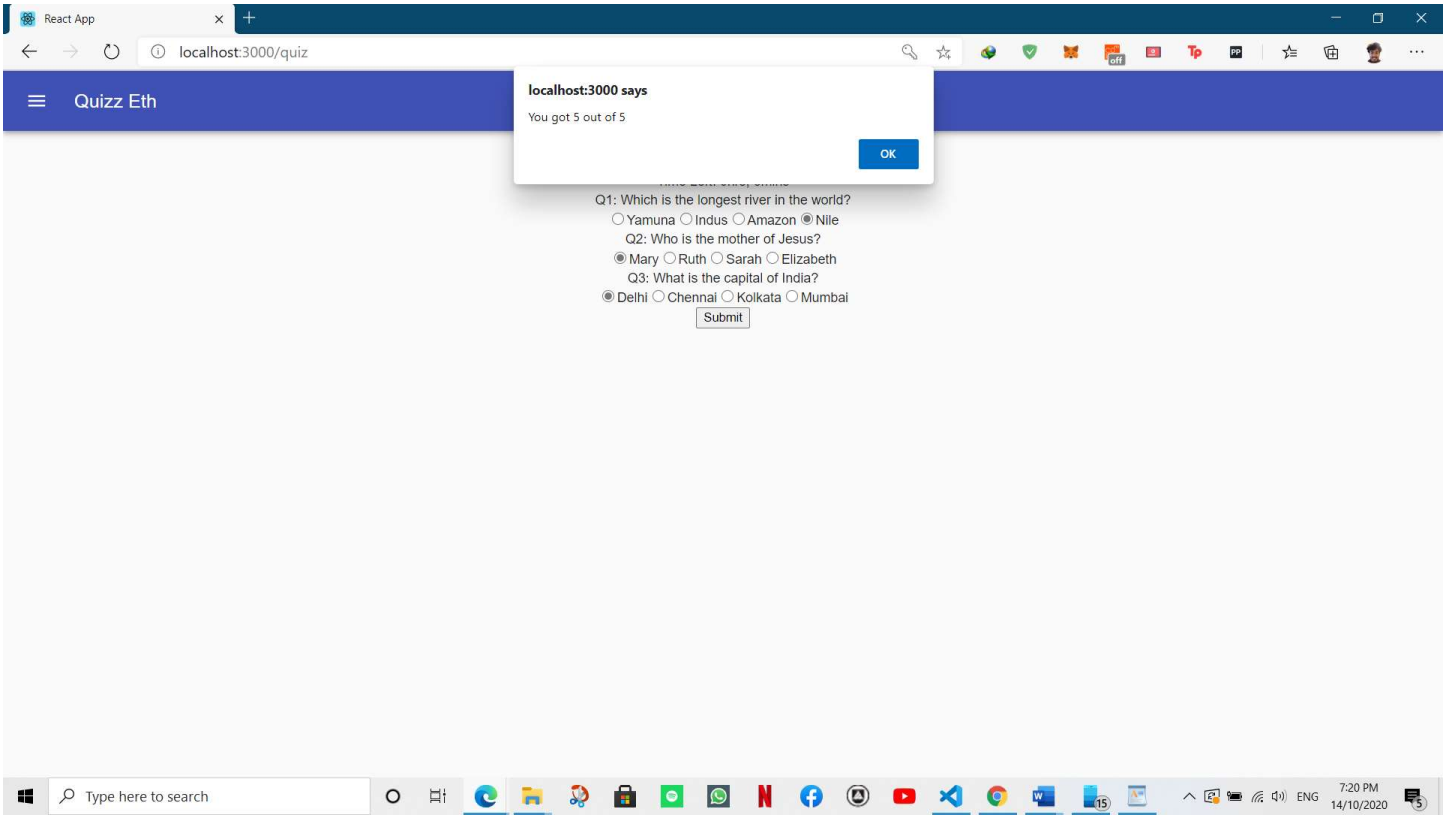

Fig 11: Quiz results coming out as an alert

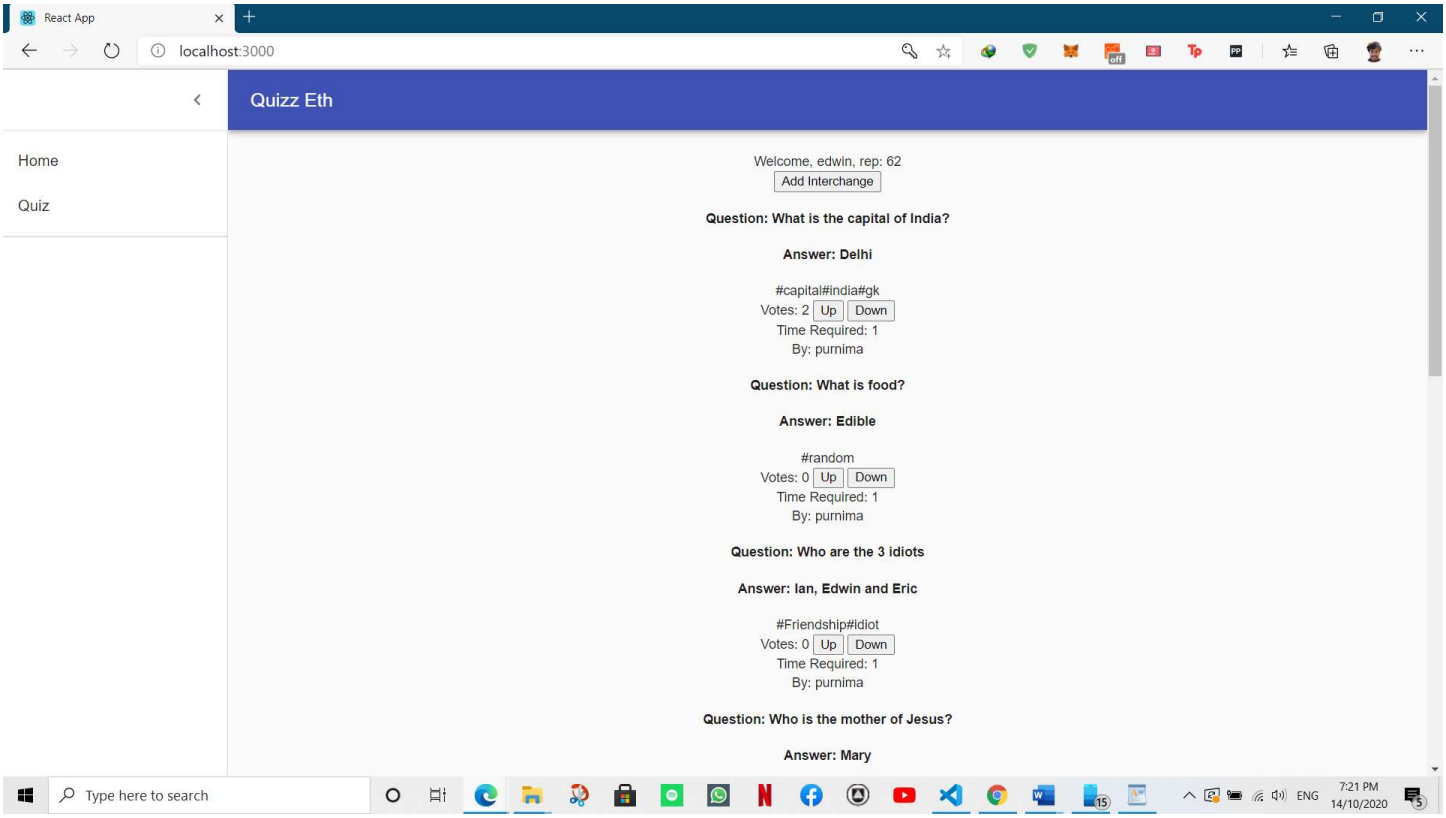

Fig 13: Increased reputation of the user due to answering questions correctly.

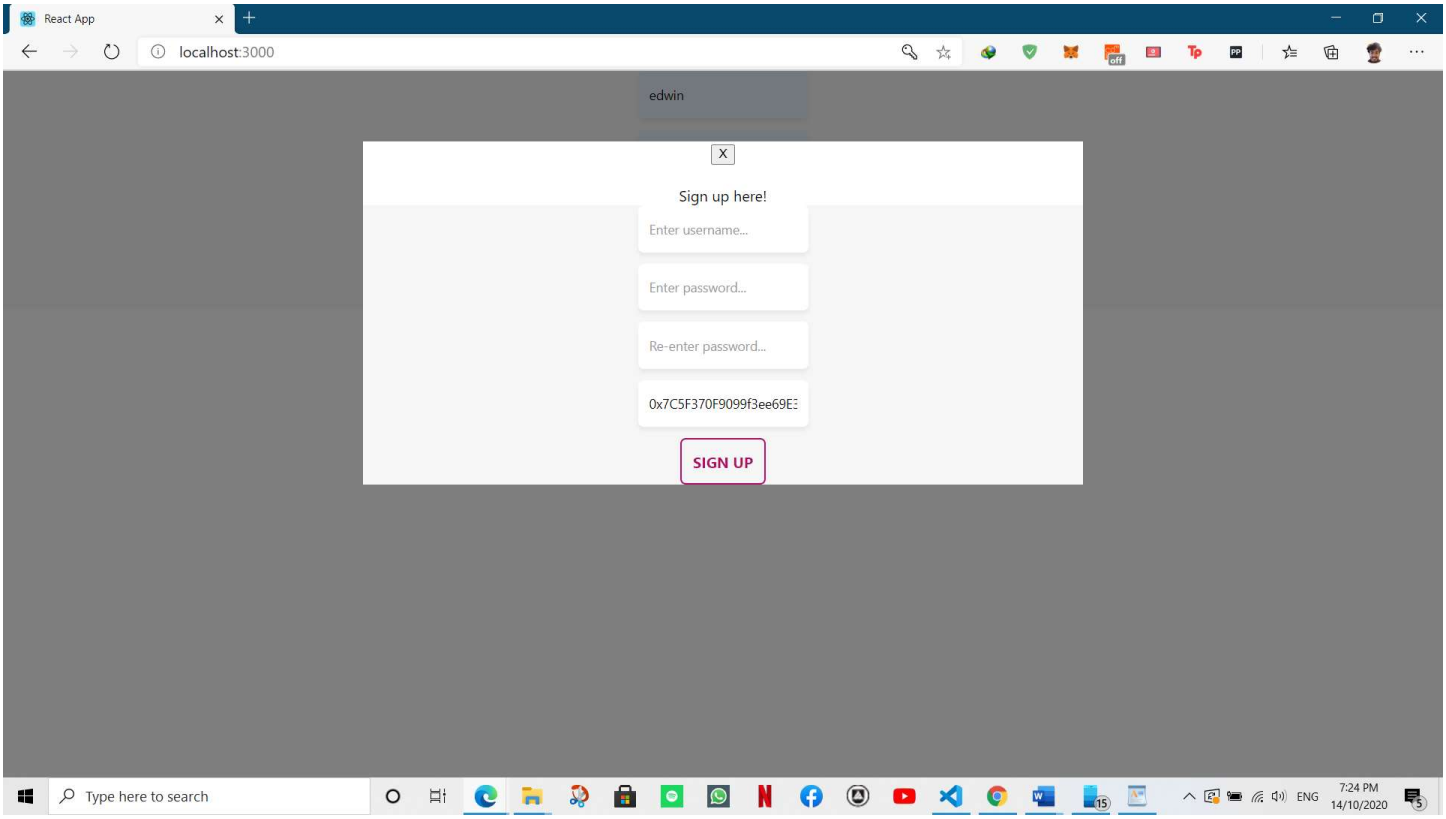

Fig 14: The sign-up page for a user to register into QuizzEth

Running QuizzEth

QuizzEth is available as a public repository on github: https://github.com/ParadoxTwo/quizz-eth . To run it, port 3000 (for frontend) and port 6000 (for backend) must of the computer system must be free. The browser should explicitly allow these ports to run. This can be done by adding --explicitly-allowed-ports=3000,6000 right after the path to browser in the shortcut. For example, for Microsoft Edge (new):
Target: "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --explicitly-allowed-ports=3000,6000
The browser must have metamask extension installed and connected to the rinkeby testnet. Once these steps are done, QuizzEth can be run.
For frontend, in the main directory of the project:
npm install
npm start

For backend, in the backend directory of the project:
npm install
node server.js

The first run will take a few minutes due to the installation of node packages and setting of the configuration.
A demo video of the running QuizzEth can be found here:
https://youtu.be/m7OMR_bnmPk

Conclusion

This paper starts with explaining the need for examinations and how online quizzes are helpful in achieving better learning outcomes. It moves on to discuss on four existing solutions to online quizzing: 2 centralized (QuizDB & MacQTEX Randomised Quiz System) and 2 decentralized ("Online quiz implementation using blockchain technology for result tampering prevention" & "Online Quiz Scheme Based on Double-layer Consortium Blockchain"). It talked about the strengths of those systems and their weaknesses as well. The 2 decentralized systems are considered good solutions; however, they are lacking in certain aspects, mainly due to the limitations of blockchain found in small networks and due to lacking other features essential to quizzing. To expand the research perspective, many related papers were discussed about, which could add to the existing solutions; perfecting them to a high degree. One of the prominent additions was turning online quizzing into an open, community-based system which could accept people from all over the world, like StackOverflow and Quora. This system would promote experts to put in more questions and answers as well as help students to gain access to a wide range of knowledge with the capability of assessing themselves at a small fee. Heading into this direction, the paper introduces a new system which compiles all the advantages of centralized and decentralized quiz systems as well as community-based systems together to build QuizzEth. The design plans and requirements, as well as its implementation using Reactjs, Nodejs, Mongodb, Express, Web3js, Solidity and Ethereum are then described followed by the link to QuizzEth's git repository and instructions on running it.
While it was possible to design and even implement certain components of the proposed solution, it was not possible to do everything with the limited time of 3 months. Integrating components such as automated proctoring and examination review into the systems would require a whole lot more research for designing as well as implementing it. More research has to be done on using Machine Learning for not only cheat detection but also perfecting the quiz system itself. Machine Learning could possibly used to read the interests of each user and derive their needs in terms of knowledge, thus suggesting them questions and topics, and even quizzes for assessments. It could also be used to suggest experts on which topics are lacking in the system so that they could focus on giving problems and solutions on them to boost their reputations. Apart from Machine Learning, more research is needed on the economy aspects and incentive mechanisms that could be used in the system to enable liquidity of knowledge and economic flow in the community-based quiz system.

Bibliography

[1] Shaowei, David, Lingxiao, March 2013, An empirical study on developer interactions in StackOverflow, Proceedings of the 28th annual ACM Symposium on Applied Computing

[2] Lorenzo Salas-Morera, Antonio Arauzo-Azofra, Laura García-Hernández, February 2012, ANALYSIS OF ONLINE QUIZZES AS A TEACHING AND ASSESSMENT TOOL, Journal of Technology and Science Education

[3] Dana Movshovitz-Attias, Yair Movshovitz-Attias, Peter Steenkiste, Christos Faloutsos, 2013, Analysis of the Reputation System and User Contributions on a Question Answering Website: StackOverflow, IEEE/ACM International Conference

[4] John L. Dobson, 11 July 2008, The use of formative online quizzes to enhance class preparation and scores on summative exams, The American Physiological Society

[5] Australia's Bush and Trees, 2011, How many trees are cut down each year for paper, Online: https://australiasbushandtrees.weebly.com/how-many-trees-are-cut-down.html

[6] Frances Griffin and Ross Moore, 1999, MacQTEX Randomised Quiz System, AUDC Conference

[7] Kazuaki Kajitori, Kunimasa Aoki, Sohei Ito, September 2014, Developing a Compact and Practical Online Quiz System, Modern Education and Computer Science

[8] Guang Chen, Bing Xu, Manli Lu and Nian-Shing Chen, 2018, Exploring blockchain technology and its potential application for education, Chen et al. Smart Learning Environments

[9] Ali Alammary, Samah Alhazmi, Marwah Almasri and Saira Gillani, 4 June 2019, Blockchain-Based Applications in Education: A Systematic Review, Applied Sciences

[10] R. Aroul Canessane, N.Srinivasan, Abinash Beuria, Ashwini Singh, B. Muthu Kumar, 2019, Decentralised Applications Using Ethereum Blockchain, Fifth International Conference on Science Technology Engineering and Mathematics

[11] Haojian Shen, Youan Xiao, 2018, Research on Online Quiz Scheme Based on Double-layer Consortium Blockchain, 9th International Conference on Information Technology in Medicine and Education, IEEE

[12] Elena-Carmen Tentea, Valeriu Manuel Ionescu, June 2019, Online quiz implementation using blockchain technology for result tampering prevention, ECAI 2019 - International Conference

[13] Naumova O.A., Naumov D.V., Svetkina I.A, 2019, The Main Limitations of Applying Blockchain Technology in the Field of Education, 2019 International Science and Technology Conference, IEEE

[14] Wolfgang Gräther, Sabine Kolvenbach, Rudolf Ruland, Julian Schütte, Christof Ferreira Torres, Florian Wendland, 2018, Blockchain for Education: Lifelong Learning Passport, Proceedings of the 1st ERCIM Blockchain Workshop

[15] Ian Mitchell, Sukhvinder Hara and M. Sherif, 2019, dAppER : Decentralised Application for Examination Review, IEEE Xplore

[16] Amiangshu Bosu, Christopher S. Corley, Dustin Heaton, Debarshi Chatterji, Jeffrey C. Carver, Nicholas A. Kraft, 2013, Building Reputation in StackOverflow: An Empirical Investigation, IEEE

[17] Sumanth Patil, Kyumin Lee, December 2015, Detecting experts on Quora: by their activity, quality of answers, linguistic characteristics and temporal behaviors, Springer-Verlag Wien 2015

[18] Mohammad A Sarrayrih, Mohammed Ilyas, January 2013, Challenges of Online Exam, Performances and problems for Online University Exam, IJCSI International Journal of Computer Science

[19] Yousef Atoum, Liping Chen, Alex X. Liu, Stephen D. H. Hsu, and Xiaoming Liu, July 2017, Automated Online Exam Proctoring, IEEE TRANSACTIONS ON MULTIMEDIA

[20] Kiran, 18/04/2020, "Is online Exam better suited for Academic Classes?", resultsfor.in, online: https://resultfor.in/online-exam-vs-offline-exam/