BlockChain E-Textbook

Final Report

Authors: Syed Arib Ali, Liam Gillies, Elizabeth Mulvaney

Client: Dr. Cliff Shaffer

CS 4624 Multimedia, Hypertext, and Information Access

Instructor: Edward A. Fox

Virginia Tech, Blacksburg VA 24061

December 08, 2021

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The goal of the Blockchain Etextbook group was to develop new content related to Ethereum under the Blockchain section of the OpenDSA textbook. OpenDSA is designed to inform students and researchers within the field of computer science about key topics within the field. This team specifically covered the content related to Ethereum. The expected audience for this topic is students and researchers either currently working in, or who are studying the topics within, Blockchain. This textbook aims to provide a single place for referencing material related to the topic.

Under the supervision of Dr. Cliff Shaffer at Virginia Tech, the team developed Blockchain content for the textbook. This included creating interactive exercises for the users to learn with and writing prose composed from researching resources about Blockchain and Ethereum. The original description of the project covered topics broadly within Blockchain but Dr. Shaffer narrowed his interest with the team down to Ethereum and topics related to that.

The team wrote textbook content related to the concepts of Ethereum including proof of stake, hard forks, crypto hacking, Ethereum Virtual Machine (EVM), and Gas. Our deliverables were reStructured Text files and HTML exercises related to these topics. In addition, the report gives users a tutorial on how to use the chapters within the textbook as well as giving future developers details on how to modify and improve chapters within the books. The team learned some of the issues with writing a textbook on new material since there is often limited or conflicting information regarding the topics.

Introduction

Overview

OpenDSA

OpenDSA is an online textbook geared toward topics in Computer Science. It includes topics such as Blockchain, Data Structures and Algorithms, Formal Languages, and Software Design. It serves as an introduction to several languages including Java. This system is free to use; however, there are sections that must be accessed through a Canvas course. Canvas is a digital learning platform used for organizing classes and submitting assignments. In fact, several courses at Virginia Tech use the content from OpenDSA to supplement the details covered in Canvas lessons. The OpenDSA textbook is composed of text for users to read, interactive exercises, images, and slideshows to demonstrate the main ideas of each topic it covers.

When added to a Canvas course, the OpenDSA chapters can be included as assignments. As students complete more of the exercises and quizzes available, they gain points toward their grade. The OpenDSA system is contained in a public Github repository, so it is open source. The main source code for this project is HTML and JavaScript, but some of the pages use Python as well. The pages are structured in restructured text files (RSTs) and the exercises are mainly in HTML and JavaScript. Dr. Clifford Shaffer asked the team to create new, and modify existing, files within the Git repository for topics within Blockchain. Specifically, his goal was to increase the content for Ethereum since another individual at Virginia Tech was contributing to the topics within Bitcoin.

Blockchain

The topic of Blockchain encompasses Bitcoin, Ethereum, Merkle Trees, applications of blockchain, and other cryptocurrencies [1]. Even though the topics of Blockchain are vast, our team focused on the specifics within Ethereum. Blockchain is a list of nodes on a network that contains a database shared amongst each node. This list of nodes is the backbone of cryptocurrencies since they are known to be secure and are not held under a centralized government or national bank system [1].

Ethereum

Ethereum is a cryptocurrency developed using Blockchain technology. While Bitcoin is primarily a cryptocurrency, Ethereum is significantly more than that. Ethereum uses Blockchain technology for its coin (ETH) and a more secure internet in its applications by using hashed blocks containing transaction data and the data on the Blockchain [2].

Where Bitcoin handles transactions without any additional fees, Ethereum does apply an additional fee [2].

There are two versions of Ethereum currently in existence: Ethereum Classic and Ethereum 2.0. The difference lies in a component called a consensus algorithm. Consensus algorithms are protocols enacted on a Blockchain ensuring that each new block is correct. The two main algorithms for this are proof of work and proof of stake. Proof of work is the algorithm investors mine to gain a new block. These algorithms are like puzzles to solve. As for proof of stake, this algorithm involves its users in investing a percentage of their coins to become validators. Validators work to choose the next blocks in the chain. Upon adding the block to the chain, validators are rewarded in accordance with how much coin they staked into the system [3]. Ethereum Classic uses a proof of work consensus algorithm to add new blocks to the Blockchain. Ethereum 2.0 uses both proof of work and proof of stake to add new blocks to the Blockchain. Ethereum is working toward converting entirely to a proof of stake consensus algorithm because of the issues revolving around proof of work.

Deliverables

At first, the deliverables were defined as updating content for Ethereum's Merkle Trees and consensus algorithms, Blockchain examples in real life, and good demonstrations for cryptocurrencies that use consensus algorithms different from proof of work. After talking with Dr. Shaffer, these deliverables changed because another student was contributing to the Blockchain section of the OpenDSA textbook.

The final deliverables for this project are defined as

1. Visualizations for existing and new topics
2. Prose for topics in:
    a. Ethereum
    b. Applications of Blockchain

These deliverables were condensed into content relating to Ethereum itself and its history, Ethereum Virtual Machine and Gas, and Proof of Stake. The team provided more content related to Ethereum and Blockchain including modifying the content related to consensus algorithms and crypto hacking.

Requirements

The requirements for this project, as laid out by Dr. Shaffer, were to develop new content for the OpenDSA textbook. Specifically, the team was tasked with adding new content to the Blockchain section. He requested the team work on sections related to Ethereum since this topic had very little detail.

As part of developing in the OpenDSA, the team works within the Github repository for OpenDSA. The team created a local instance of the textbook within their own Docker container to test the exercises and the visibility of the content. As for the content, the team created restructured text files (RST) and exercises for any content created in the textbook. In addition, the team created new exercises or visualizations to update the content existing in the textbook. Beyond this, there were no specific details given by Dr. Shaffer.

When writing content for the textbook, the RST files were reviewed by Dr. Shaffer on a weekly basis to ensure clarity of the material as well as ensuring the content is complete enough for the textbook. The team took feedback from Dr. Shaffer and edited the content according to his feedback. In addition, the team designed storyboards for content related to the exercises and gained feedback from Dr. Shaffer. The exercises were required to be easy for the user to understand and perform the tasks in each module.

Design

The design of the content for the OpenDSA website is done in place through the OpenDSA system. The RST files are implemented directly into the website using Docker. The RST files are the content of the pages, containing headings, and blocks of content, including links to other chapters within the textbook. The main aspect and room for design from the team is with the exercises and visual content within the textbook chapters.

The exercises must be easy for a user to use, and they must emphasize points made within the RST files. There are examples of exercises already created within the Github repository. In fact, these are good examples or templates for implementing new exercises in the content. Using these exercises for inspiration, the team developed storyboards and integrated the concepts from the RST files into visualizations to help understanding.

Implementation

Restructured Text (RST) Files

The main source of communicating details about the topics within Blockchain and Ethereum is the RST files. The RST files contain researched prose explaining everything appropriate for someone curious or learning about these topics. This is not a source of visualization, but it is the bulk of the material the team provides. These RST files are composed after completing research related to the topics and are further defined by the research done by the team. In fact, one RST file can turn into multiple as more branching topics are discovered or when discovering more topics that can be revised.

The interesting thing regarding these RST files is that there are already some of these topics written about in the textbook. The team linked the content within those RST files into the new content and revised them to fit their needs. Also, since these are already at least partially developed, it gave examples of content to go off of. However, the existing RST files provided another source of work for the team because there is always more information to add to them, especially if the team uncovered something by their research on the other topics.

The restructured text files contain regular text but are formatted for use in headings. The headings are defined by what is on the line underneath it. For example:

Ethereum

=======

is a title heading, while

Ethereum’s history

----

is a subheading and

Cyber Hacks in Ethereum

~~~~~~~~~~~~~~~~~~~

indicates that “Cyber Hacks in Ethereum” is a section in “Ethereum’s history.” In addition, comments for improvements or placeholders for other content are placed in where comments begin with “..” and can last for several lines. Also, the exercises are linked into the RST files by using:
~~~~~~~~~~~~~~~~~~~

.. avembed:: <PathToFileFromHomeDirectoy> ka

:long_name: <NameAssociatedWithTheFile>

An example of this is used for linking an exercise dedicated to identifying details of a phishing email.

.. avembed:: Exercises/Blockchain/PhishingEmail.html ka

:long_name: Phishing Email

In addition, linking images to the files is as simple as using:

.. odsafig:: <PathToImage>

:align: <Alignment>

:width: <SizeInPixels>

An example of this is:

.. odsafig:: Images/LocalBitCoins.jpg

:align: center

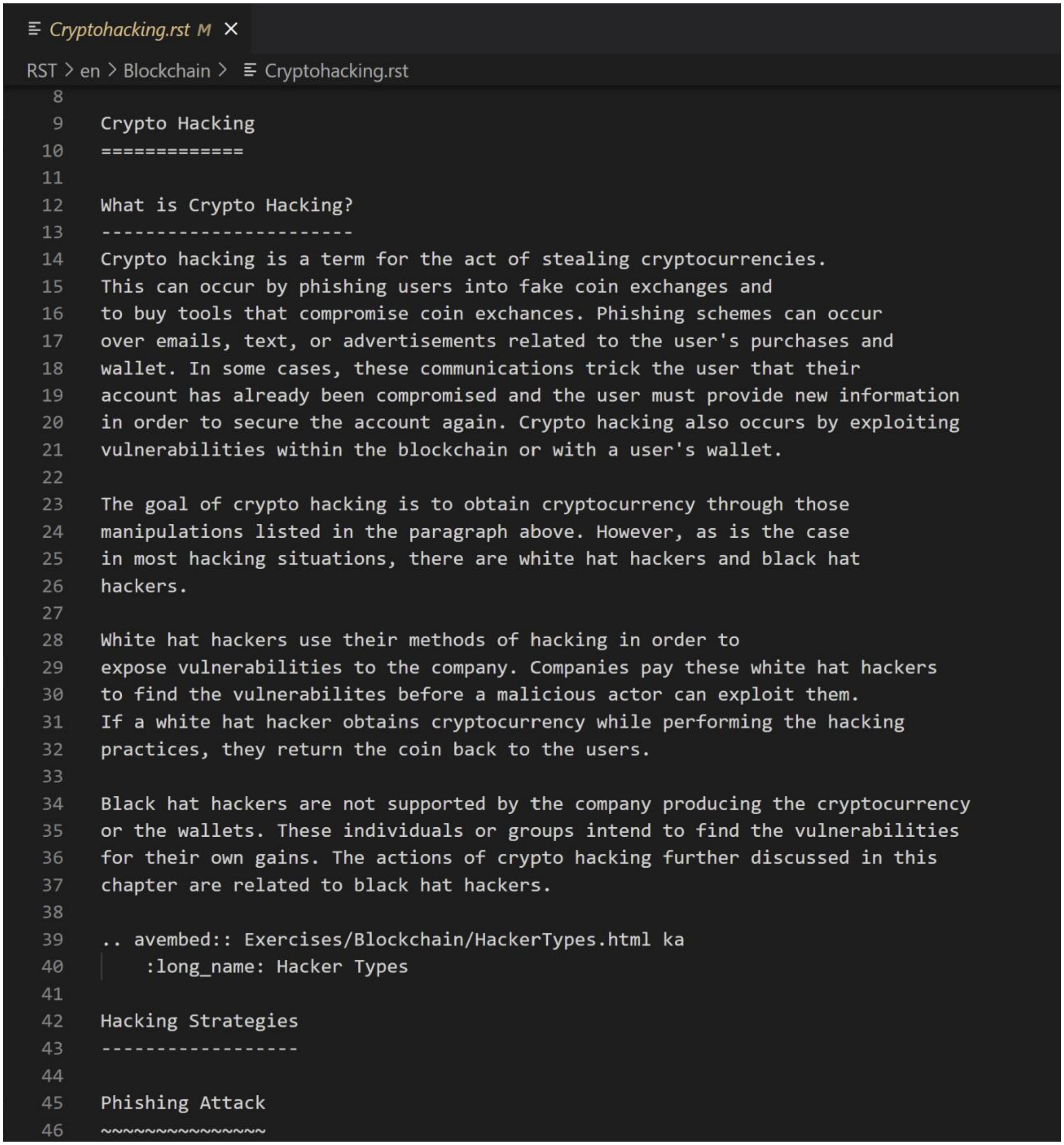
:width: 800

Linking between RST files can be done two ways: the first is using the “doc” keyword. It works like this :doc:`<title of the file to link>`

For example, :doc:`Consensus` creates a link to the Consensus Algorithm RST file embedded in the current RST file you are working in or reading.

The other way to link files is to use “ref”. It works like this :ref:`LinkText <PathToFile> <Section>`. An example of this is :ref:`gas <gas> <EVMandGAS>`.

An example of these linking and header methods is shown in Figure 1.




```

Crypto Hacking
=============

What is Crypto Hacking?
-----------------------
Crypto hacking is a term for the act of stealing cryptocurrencies.
This can occur by phishing users into fake coin exchanges and
to buy tools that compromise coin exchances. Phishing schemes can occur
over emails, text, or advertisements related to the user's purchases and
wallet. In some cases, these communications trick the user that their
account has already been compromised and the user must provide new information
in order to secure the account again. Crypto hacking also occurs by exploiting
vulnerabilities within the blockchain or with a user's wallet.

The goal of crypto hacking is to obtain cryptocurrency through those
manipulations listed in the paragraph above. However, as is the case
in most hacking situations, there are white hat hackers and black hat
hackers.

White hat hackers use their methods of hacking in order to
expose vulnerabilities to the company. Companies pay these white hat hackers
to find the vulnerabilites before a malicious actor can exploit them.
If a white hat hacker obtains cryptocurrency while performing the hacking
practices, they return the coin back to the users.

Black hat hackers are not supported by the company producing the cryptocurrency
or the wallets. These individuals or groups intend to find the vulnerabilities
for their own gains. The actions of crypto hacking further discussed in this
chapter are related to black hat hackers.

.. avembed:: Exercises/Blockchain/HackerTypes.html ka
    :long_name: Hacker Types

Hacking Strategies
------------------

Phishing Attack
~~~~~~~~~~~~~~~
```

Figure 1: Demonstrating linking and headers for the Cryptohacking.rst file.

One of the most important aspects of the RST files is that they need to be clear and concise for the reader to understand. This requires the writer to have someone proofread the content for understanding as well as grammar and to recommend new topics to cover in the pages. It is important to get that feedback from the proofreader because that can include questions that need to be clarified or addressed if not previously done so.

Exercises and Visualizations

Exercises

Exercises are created using a combination of HTML and JavaScript files. The exercises use the framework defined by Khan Academy for its exercises hosted on that tutorial-based website. These exercises are under the Creative Commons by-sc-na license, allowing for sharing and adaptations of the framework. Exercise design has been modified from the framework linked in the project Khan-exercises under the Khan project. This project is dedicated to creating the content that is included on the Khan Academy website [4].

To see examples given in the OpenDSA Github, in the home directory click on Exercises. Then click on SimpleDemo. Each example given in the SimpleDemo directory includes an HTML file and a JavaScript file detailing the format and behavior of the exercise quizzes. The examples given in this directory include a fill in the blank context, clicking an object within the exercise frame, and a multiple-choice option. In addition, if necessary, the team can create CSS files for the design of the exercises. However, this is not required for developing the exercises. Using this and other exercises as an example, the team formed other exercises for the topics under Blockchain. Figure 2 demonstrates some of the code used to create a quiz. In the quiz, each answer choice except for the answer is represented as a list element to an unordered list named choices. The answer is a separately divided element under the class of solution. Below the unordered list are the hints. Each hint would be its own paragraph element.

ProofOfStakeEnergy.html ×

Exercises > Blockchain > ProofOfStakeEnergy.html > ...

```html
<!DOCTYPE html>
<html data-require="math">
  <head>
    <title>Proof of Stake: Energy Consumption</title>
    <script src="//cdnjs.cloudflare.com/ajax/libs/require.js/2.1.14/require.min.js"></script>
    <script src="../../khan-exercises/local-only/main.js" ></script>
    <script src="https://google-code-prettify.googlecode.com/svn/loader/run_prettify.js"></script>
  </head>

  <body data-height="650" data-width="950">
    <div class="exercise">
      <div class="problems">
        <div id="ProofOfStakeEnergy">
          <p> Why does proof of stake use less energy than a proof of work system? </p>
          <div class="solution"> Currency collection does not take high computing power. </div>
          <ul class="choices" >
            <li> Proof of stake algorithms use the computing power to mine cryptocurrency. </li>
            <li> Users with higher concentrations of cryptocurrency prevent others from forging their own cryptocurrency. </li>
            <li> Proof of stake users need powerful computing machines to put stake into the system. </li>
          </ul>

          <div class="hints">
            <p>Look at each section in Proof of Stake versus Proof of Work.</p>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Figure 2: Code used for a proof of stake quiz discussing the environmental impact of proof of stake consensus algorithms.

These exercises are introduced by discussion with Dr. Shaffer, then storyboards for each exercise are drafted. Upon completion of these drafts, we discuss them with Dr. Shaffer and begin drafting the exercises. Each RST file needs at least one visualization or exercise to emphasize the points discussed in each topic.

For the Proof of Stake RST file implemented in the textbook, there are exercises designed using the modified Khan Academy framework used for quizzing the user on what would increase an investor’s chance of being selected as a validator in the proof-of-stake consensus algorithm. The user will read the question, click the answer they think is correct, and click the Check Answer button. If they’re wrong, they can click the “I’d like a hint” button to receive a hint. An example of this implementation is seen in Figure 3.

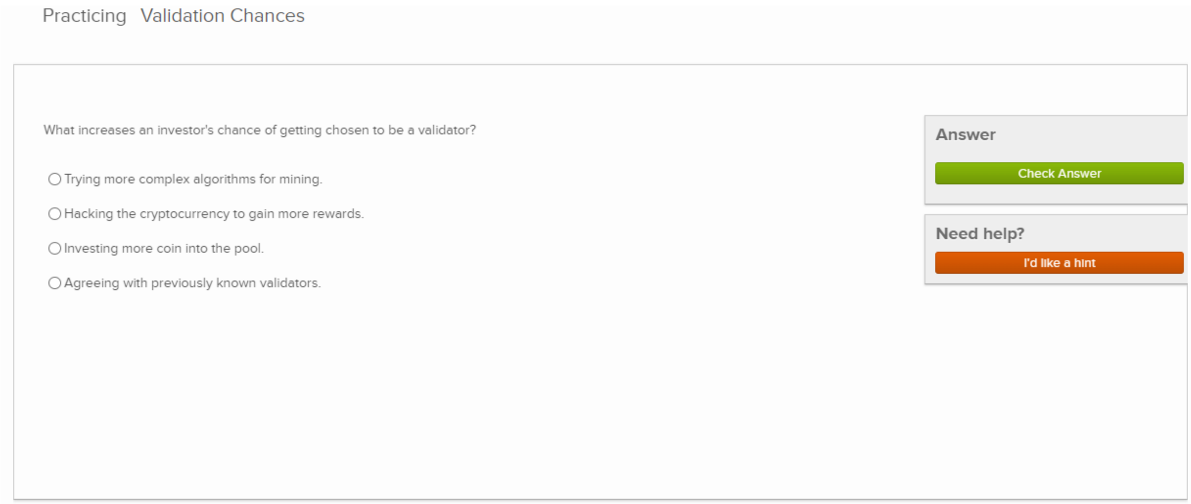


Figure 3: OpenDSA Khan Academy exercise in the Proof of Stake RST file.

If the user selects the correct answer, the quiz will tell them they got it right. Then they can continue reading onward past the quiz. If there are additional questions, they click on and continue to the next question as seen in Figure 4.

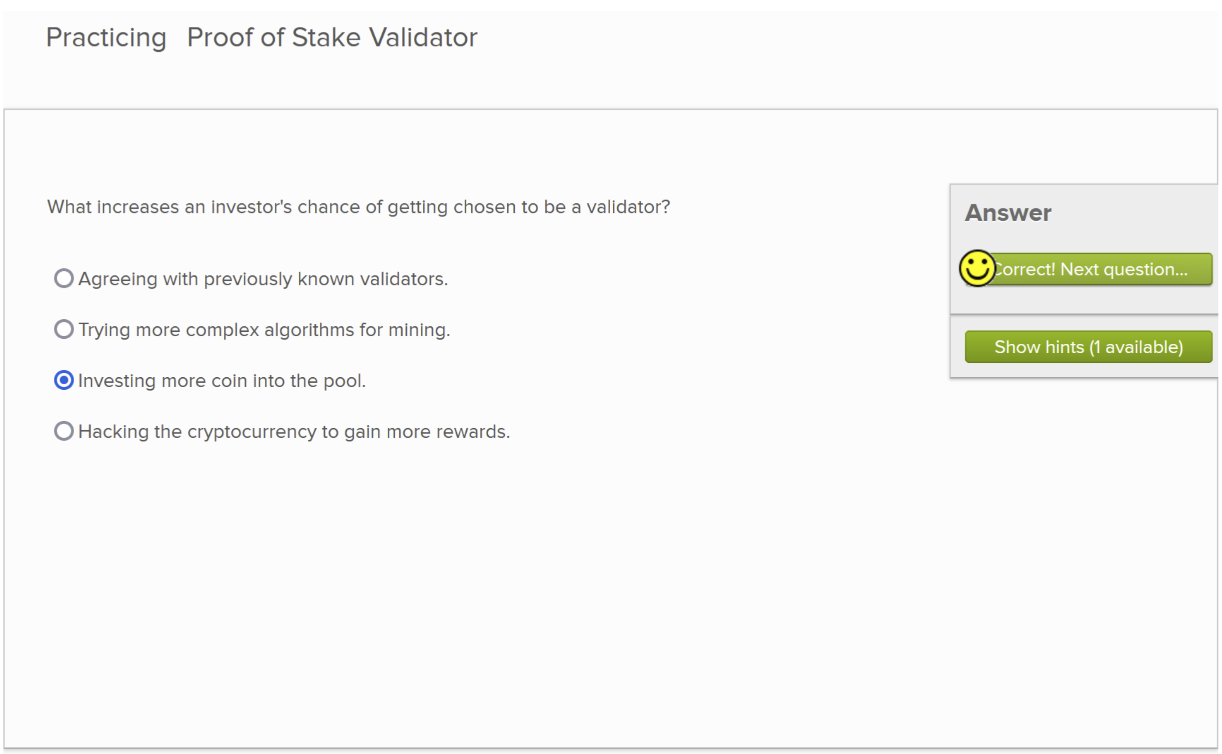


Figure 4: Demonstrating giving a correct answer to the 'What increases an investor's chance of getting chosen to be a validator?' question.

Visualizations

If the topic has a section that can be expressed with a visualization, they will be under the AV directory. Topics under Blockchain are sorted in the Blockchain folder. These visualizations or slideshows (collections of visualizations) are composed of HTML, CSS, and JavaScript files. These forms of media portray a topic such that the user can see what is happening in the events described in the topic rather than just reading them.

Online software can also be used to create desired visualizations. One example implementation of this can be seen in Figure 5, where an online graph maker [5] was used. To use the online graph maker, first select a graph style which contains the background, line style, and font. Then, input all x and y values to be graphed along with changing x and y-axis titles. Finally, export the graph and insert that as an avembed in the intended RST file.

In terms of visualization, this can also include graphs for data if this is needed, a timeline of events, or imagery found related to the topic. These images may be collected in a format like a presentation or slideshow. The user can choose to move forward within the slideshow or backward to see different images. The EVM and Gas RST file has a graph of data showing the price of gas at different increments, shown in Figure 5.

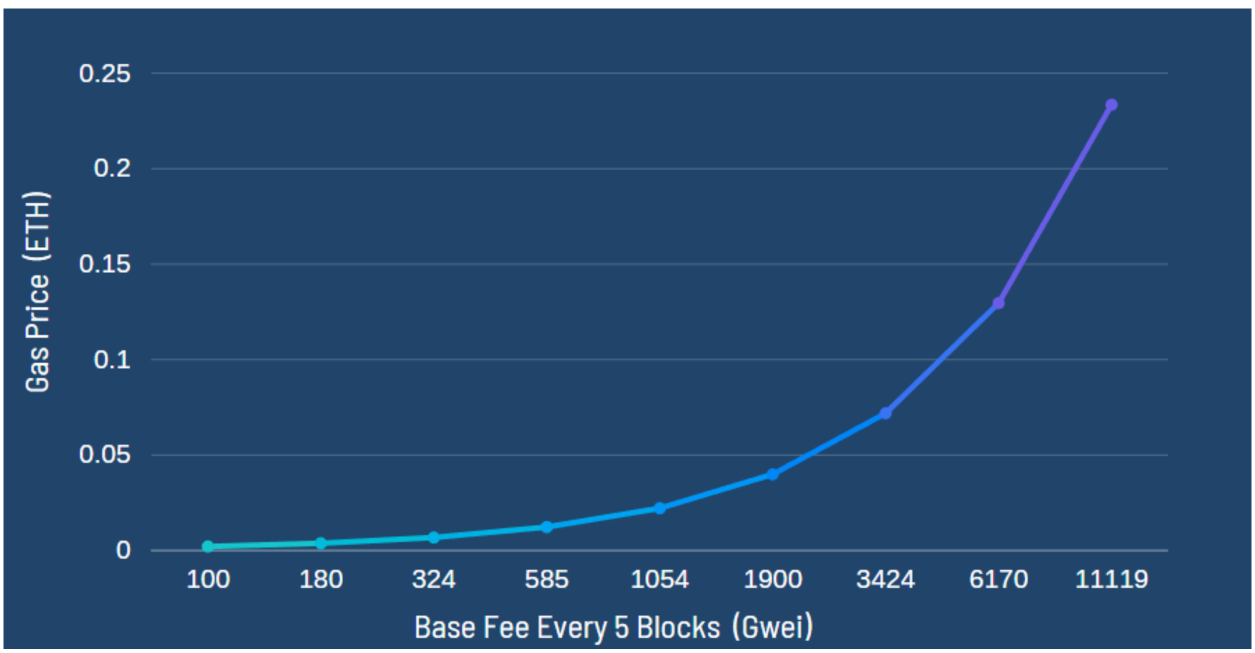


Figure 5: OpenDSA visualization in the EVM and Gas RST file

This visualization is used to demonstrate how the fees for an Ethereum transaction can grow exponentially depending on the load on the Ethereum network. It is more beneficial than just typing out numbers and is much easier to understand than other forms of media.

Testing/Evaluation/Assessment

Testing is done by all members of the team throughout production. As more exercises are created and modified, it is important that they be in working order. In addition, any content added to the textbook in RST files needs to be proofread for clarity. Just because you understand what is being said, does not mean the readers will.

For testing, this can be hosted locally and globally. The testing of the components occurs using Docker containers. In order to do this, ensure the user is in administration mode on their computer if their host user is not already the admin. Docker does not work for users who do not have administrator capabilities.

1. Start the service up for Docker in the directory that contains the OpenDSA git using the terminal by running 'docker-compose up'. Once the line 'opendsa_1 | * Serving Flask app "app.py"' appears on the terminal, the service is ready.
2. Change the directory to config and run 'make Blockchain.'
3. Open a browser to the page 'http://127.0.0.1:8080/.'

Once the local system is set up, go to the page the team member is working on and manually check the visualizations and exercises to ensure that each button works as expected. For the visualization presentations, ensure that the image changes to the next one in the sequence when clicking the next arrow and goes backward in the sequence when clicking the back arrow. For the exercises, run through all of the possible combinations of clicking and interaction to ensure it runs properly.

All of the team tested locally using Windows 10 computers. The testing was done for the visualizations and exercises since the RST files are text.

The RST files were tested through proofreading for clarity. The only way they would be tested via computing is to ensure the exercises, visualizations, and links to other chapters within the textbook are correct. This means that when the book is compiled, the components appear in the page in the appropriate space. When proofreading, the team members considered how the content would appear to someone with limited knowledge within Blockchain. If you know someone who fits that description, they can aid you in reading and give you feedback based on what they understand and what is not clear to them.

Users' Manual

The user manual explains the basic details of how to navigate and use the exercises seen in the Blockchain section of the OpenDSA project. Start by accessing the website the content is hosted on.

The website is at http://lti.cs.vt.edu/LTI_ruby/Books/Blockchain/html/ and is accessible through any browser. It does not need a Virginia Tech account to be accessed. Figure 6 displays the details of the table of contents for the Blockchain section. The content the team has edited is under Chapter 4. See Figure 6 for details on the different pages accessible on the website for Cryptocurrencies.

Chapter 4 CryptoCurrencies

Figure 6: Chapter 4 Directories Containing Information About Cryptocurrency

The sections particular to this team's work are Ethereum, The Ethereum Virtual Machine and Gas, and Proof of Stake. Clicking on the header will lead you to the page in question. For example, clicking "4.4 Proof of Stake" will take the user to a page detailing information about proof of stake consensus algorithms. The subheadings such as “4.4.1 What is a consensus algorithm?” lead to a section of a textbook page so clicking on that link would send the user to a section like in Figure 7.

4.4.2. What is Proof of Stake?

Proof of stake is a way for an individual to validate (that is, claim the right to add) a new block to a distributed ledger. Proof of Stake is in contrast to the Proof of Work approach used in BitCoin. Proof of Stake is used by the Ethereum cryptocurrency, and its most important distinction is that this validation process does not require “mining” in the sense of expending a significant amount of computing resources to claim the right to validate (and add) the next block. As of December 2020, Ethereum 2.0 uses proof of stake to validate blocks. To indicate the distinction, the process of making the claim to validate the next block (and thereby gain the associated coin as a reward for this contribution to the community) is referred to as “forging” instead of “mining”.

Examples of cryptocurrency that uses proof of stake as their consensus algorithm include Peercoin, Ethereum, Tezos (XZT), Binance coin (BNB), NEO, PIVX, Neblio (NEBL), Cardano (ADA), and Stratis (STRAX).

Figure 7: The “4.4.2 What is Proof of Stake?” subsection demonstrating how accessing the subsections are done.

As for the exercises, follow the instructions given in the exercise. If the component is a drag and drop, click on the component needed to drag and move it to the right location. If the component is a quiz, click on the correct answer and click submit. If the user wants to retry the material, if there is a reset button, click on that. Otherwise, refresh the screen.

Work Distribution

Everyone in the group chose a topic to write about pertaining to Ethereum. The division of work is shown starting in Table 1. Table 1 shows the work done by Liam Gillies. Liam worked on the topic related to Ethereum Virtual Machine (EVM). Within this topic, he created content pertaining to what Ethereum State is, the cost of gas, and a visualization relating the cost of gas and the base fee for every five blocks in a blockchain. Accessing these pages is easy because the HTML files are linked into the RST files. Once you click on the Ethereum Virtual Machine and Gas link, the figure and the exercises automatically load. Within this page, you will read about details pertaining to what an Ethereum Virtual Machine is and what it does, what Ethereum State is, and what gas is and what it does. It gives a brief description of the cost of gas.

Contribution Number	File Name	Content Type	Word Count
1	CostOfGas.html	Exercise: Quiz Question	111
2	EthereumState.html	Exercise: Quiz Question	105
3	EVMandGas.rst	Content	2,427
4	GasPricePerBaseFee.png	Visualization	N/A

Table 1: Contributions, file type, and word count for files created by Liam Gillies.

As for the second teammate on the team, Arib Ali, his contribution is displayed in Table 2. He wrote the content for fundamental concepts of Ethereum, where anyone new to Ethereum can attain some background knowledge and motivation to read about Ethereum. In his section, he discusses Ethereum's background and how it relates to Blockchain. He discusses Ethereum's history, where he talks about Ethereum's early days, its stages of process, how Ethereum Blockchain gained popularity, and its price through time. Arib also discusses some current issues in cryptocurrencies like Bitcoin and how Ethereum works to solve such issues. Lastly, he discusses key differences between Ethereum and Bitcoin, where he discusses the differences between the two technologies' hashing algorithms, scaling solutions, and average blocktime. Blocktime is the time for new blocks of data to be added, which determines the time it takes to confirm transactions [12].

Contribution Number	File Name	Content Type	Word Count
1	Ethereum.rst	Content	1052

Table 2: Contribution, file type, and word count for the file created by Arib Ali.

As for Elizabeth Mulvaney, she produced content as seen in Table 3. Her content pertained to such topics including Crypto Hacking, Proof of Stake, and Hard Forks.

Crypto hacking discusses some of the threats to cryptocurrency. This includes phishing, 51% attacks, and crypto jacking. In addition, she has a quiz asking the user to identify what a white hat hacker is and another exercise pertaining to recognizing phishing emails. She finishes this page with a discussion about how to mitigate the risk of a crypto hacking event happening to the user. The second page she worked on is the Proof of Stake page. On this page, there are details about what proof of stake is as well as how it differs from the other main consensus algorithm: proof of work. In this page, there are two full length quizzes with three questions each pertaining to each section. Lastly, she worked on a section for forking within the blockchain including what a hard fork is, how it impacts the chain, and how consensus algorithms impact a hard fork. She included an exercise detailing how a hard fork impacts the Blockchain.

Contribution Number	**File Name**	**Type**	**Word Count**
1	ProofOfStake.rst	Content	1,927
2	HardForks.rst	Content	783
3	Cryptohacking.rst	Content	1,122
4	HackerTypes.html	Exercise: Quiz Question	77
5	PhishingEmail.html	Exercise: Quiz Question	105
6	Proof_of_stake_chances_validation.html	Exercise: Quiz Question	94
7	Proof_of_stake_chances_tf.html	Exercise: Quiz Question	75
8	ValidationRoles.html	Exercise: Quiz Question	95
9	ProofOfStakeValidatorsSumm.html	Exercise: Quiz Grouping	38
10	CoinWorth51%Attack.html	Exercise: Quiz Question	78
11	ProofOfStakeRisks.html	Exercise: Quiz Question	108
12	ProofOfStakeEnergy.html	Exercise: Quiz Question	122
13	PoWPosSumm.html	Exercise: Quiz Grouping	40
14	HardForksQuiz.html	Exercise: Quiz Question	105

Table 3: Breakdown of Elizabeth Mulvaney's Contributions to the Blockchain Section for OpenDSA. This includes word count and what type of contribution was made.

Developer's Manual

Methodology

Our users are Computer Science students who are interested in learning more about Blockchain technology since our project is hosted on OpenDSA.

Persona A: Bob is a male 20-year-old Computer Science student who has minimal knowledge of Blockchain technology and is interested in learning more about the topic. He thinks that cryptocurrency might be widely adopted in the future and is hoping to invest in it. In his free time, he likes to play video games and chat with his friends.

Persona B: Katy is a female 18-year-old Computer Science student who has some knowledge of Blockchain technology but is looking to solidify it. She is already invested in cryptocurrencies but has had a hard time finding easy-to-understand articles to figure out how cryptocurrencies really work. In her free time, she likes to play with her pets and drink tea.

Persona C: Bernard is a male 68-year-old retired worker who has heard his children talk about this new craze, cryptocurrency. He is looking for ways to figure more out about the subject and is actively looking for resources that are easy for him to read. He has deteriorating eyesight and trouble reading from a screen for long periods of time, so his options are limited. In his free time, he watches TV and drinks lots of coffee.

Goals for all types of users:

1. Learn about Blockchain
2. Experience a simple user interface that is easy to navigate
3. Have interactive learning through things such as animated exercises

Goal 1: Learn about Blockchain

Subtask a: The content should be specific to Blockchain. The content should be understood by anyone who is interested in Blockchain and has no prior knowledge about it. There will be content which will make more sense to CS students such as hashing algorithms that blockchain utilizes; however, this should not discourage non-CS learners.

Goal 2: Experience a simple user interface that is easy to navigate

Subtask a: Consider font color for easier readability

Subtask b: Consider font type and font size for comfortable readability

Subtask c: Ensure that there are figures explaining concepts which can better be explained visually. If there is only text on the E-Textbook, then users will not enjoy learning from this book.

Goal 3: Have interactive learning through things such as animated exercises

Subtask a: Consider the usability of interactive content; users should not have difficulty navigating the exercises.

Subtask b: Ensure that the animations are visually appealing and not distracting.

The diagram in Figure 8 represents a workflow that each of us followed in writing prose and designing exercises for OpenDSA. We carefully considered each goal and their subtasks in our researching and drafting process for the textbook. Then, in writing prose and designing exercises, we evaluated each of our contributions at each step to ensure they were still in line with our goals. Finally, we reviewed feedback from people who were unfamiliar with Blockchain and read our book to gauge our success.

Workflow

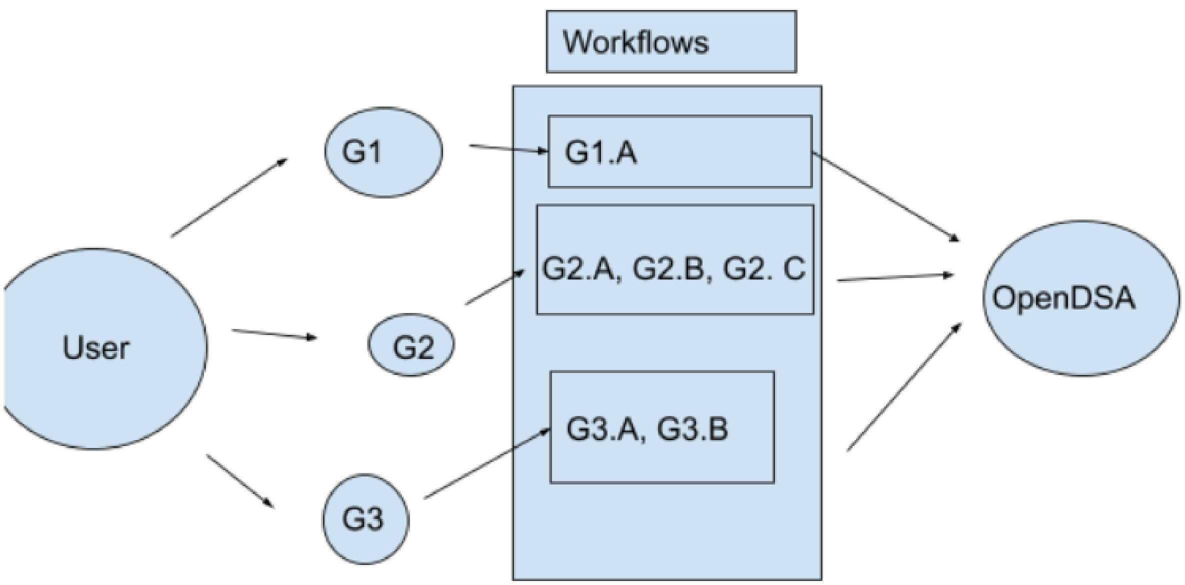


Figure 8: Workflow G1, G2, and G3 refer to Goals 1, 2, and 3. G1.A refers to part a of goal 1, etc.

Contributing to OpenDSA

OpenDSA is an open-source textbook that exists on Github. To contribute without having direct access, you must either submit a pull request with your changes and get it approved by someone who does have access or request access from someone who has been added to the project. The first option for cloning the repository and pushing your changes is achieved by the following steps:

1. Navigate to the Git Repository using the link: https://github.com/OpenDSA/OpenDSA.
2. Fork the repository to your account using the Fork button in the upper right corner of the page.
3. Open a bash enabled terminal such as Git Bash or if you are using Linux, use your command line.
4. Navigate your terminal to the directory you want to set up your files.
5. Copy the cloning link from your forked repository.
6. Type ‘git clone <copied_link>’.

Adding your changes to the repository is done by creating a pull request using the steps below:

1. ‘git pull’ -- to make sure to do this so your repository is up to date
2. ‘git add changed_file_name.rst’
3. ‘git commit -m “example commit comment”’
4. ‘git push’
5. Navigate to your forked version of OpenDSA on Github in your browser and click on the Pull Requests tab.
6. There should be a button at the top of the page asking you if you want to create a pull request. Click this button.
7. Fill out the pull request form.
8. Notify someone with write access to the repository to review and merge it. Once it’s been approved and merged, you’re done!

As for the methods without pull requests, you will need a SSH key set up on your computer and Github account for authentication. Information for setting up SSH keys can be found on this page.

1. Reach out to your client, the person requesting you to add content to the textbook, and give them your Git ID. They will grant you access to write permissions within the repository.
2. Open a bash enabled terminal such as Git Bash or if you are using Linux, use your command line.
3. Navigate your terminal to the directory you want to set up your files.

4. There are two links on the repository. ‘git clone https://YOURGITHUBID@github.com/OpenDSA/OpenDSA.git’ includes the right link to the repository for reading and writing.

After making your changes to the repository, including adding new files or modifying existing ones, perform these steps to commit those changes:

1. ‘git pull’ -- to make sure to do this so your repository is up to date
2. ‘git add changed_file_name.rst’
3. ‘git commit -m “example commit comment”’
4. ‘git push’

As for running an instance of the website on your host machine to see your changes and test the exercises and visualizations, follow these steps:

1. Ensure you have Docker installed on your computer. If not, the link to download Docker is here. Note: the user running Docker must have administrator privileges on your computer or it will not open.
2. Open a bash enabled terminal such as Git Bash or if you are using Linux, use your command line.
3. Ensure you have make installed onto your command line by using the command ‘find / -iname "make" 2>/dev/null’. If make is not on your system, use the command ‘sudo apt-get install make’.
4. Navigate to the directory where your cloned version of OpenDSA resides.
5. Enter the command “docker-compose up”.
6. Wait until you see ‘opendsa_1 | * Serving Flask app "app.py"’.
7. Open another terminal in the same directory.
8. In the new terminal, enter the command ‘docker-compose exec opendsa bash’. If you are on Windows, run ‘winpty docker-compose exec opendsa bash’ instead.
9. Once that finishes loading, run “make Blockchain” to get the Blockchain eTextBook up and running.
7. Navigate to https://opendsa.localhost.devcom.vt.edu/Books/ to view the book contents.
8. Closing the Docker instance can be done by either “ctrl + C” in the terminal hosting the Docker instance or run the command ‘docker compose down’.

Once this is done, you can start making your own changes. Books are written in reStructuredText files, also known as RST files. You can edit existing files or create new ones. Then when you want to view your changes, run “make Blockchain” again.

When you add new RST files to the textbook content, make sure you add it to the config file. This file is in the config folder in the OpenDSA directory. Click on the file named Blockchain.json. In this file, go to the JSON object named “chapters” as seen in Figure 9 below. In that JSON object, add a new line containing the path to your RST file. For the Ethereum.rst file, this is “Blockchain/Ethereum”:{},. If you have exercises linked in the RST file, you will include them within your JSON you just created. See “Blockchain/ProofOfStake” for an example of this in Figure 9.

{} Blockchain.json ×

config > {} Blockchain.json > ...

```
    },
    "chapters": {
      "Overview": {
        "Blockchain/Introduction": {},
        "Blockchain/Cryptography": {},
        "Blockchain/Blocks": {}
      },
      "Ledgers": {
        "Blockchain/Ledgers": {}
      },
      "Data Structures": {
        "Blockchain/MerkleTrees": {}
      },
      "Consensus Algorithms": {
        "Blockchain/Consensus": {},
        "Blockchain/Mining": {},
        "Blockchain/Permissionless": {},
        "Blockchain/Permissioned": {}
      },
      "CryptoCurrencies": {
        "Blockchain/BitCoin": {},
        "Blockchain/Ethereum": {},
        "Blockchain/EVMandGAS": {},
        "Blockchain/ProofOfStake": {
          "Validation Chances Quiz": {
            "points": 1.0
          }
        },
        "Blockchain/Cryptohacking": {
          "Hacker Types": {
            "points": 1.0
          }
        },
        "Blockchain/OtherCrypto": {}
      },
      "Applications of Blockchain": {
        "Blockchain/SmartContracts": {},
        "Blockchain/OtherApplications": {}
      }
    }
```

Figure 9: The config file for Blockchain.

Lessons Learned

Timeline and Schedule

Table 4 features the timeline and schedule the team followed to complete their project.

September 13	Install Docker and test compilation on individual host systems.
September 14	Collect research onto Discord and in a format accessible by Professor Shaffer. Meet with Dr. Shaffer to discuss updates for research and gain more feedback to continue.
September 21	Pitch exercise ideas to Dr. Shaffer.
September 24	Draft content for the website pages.
September 28	Draft and edit content for the website pages.
September 30	Begin development on the website.
October 1	Implement the skeleton of the pages we are working on.
October 22	Review Content of the Pages with the teammates for revision.
October 31	Website is drafted. Begin testing of the website and its components.
November 2	Implement the skeleton of the pages we are working on. These skeletons include the text content we drafted.
November 3	RST files resubmitted to Dr. Shaffer for review.
November 9	Drafted Exercises are complete for the pages of Ethereum, Proof of Stake Algorithms.
November 11	Meeting with Dr. Shaffer to discuss more about exercises.
November 16	The previously drafted exercises are implemented into the website. Each component enhances learnability for the user.
November 22	The website is tested. Revisions of each page have been made. Final edits have been made in the next week.
December 02	Present our work to the class.
December 08	Final presentation and report due.

Table 4: Timeline for work in the semester.

Challenges and Solutions

This class was the first where the team worked together on a semester long project. This was a new experience for all the members of the group and there were many challenges faced throughout the semester.

One of the biggest challenges that was encountered was that the group leader dropped the course after one third of the semester. There were many plans that were made during that time, and there were different roles assigned to each person. Also, each member of the team was given various sections to work on. After the leader left, the project was downsized, and the sections were redistributed to the members. There were many modifications made to the initial plans and timeline. Also, the leadership role was assigned to a different person.

Another one of the challenges that was faced was learning new technology and Blockchain concepts. Prior to this project, the group members were not very familiar with Blockchain or Ethereum concepts. Moreover, they had limited knowledge of JavaScript and Docker, technology that was utilized to carry out this project. There was a steep learning curve to this project, but each of the members was successfully able to utilize new skills they learned to complete this project.

Another challenge in this project was performing the research related to the topics. Articles on Blockchain are not consistent and often have missing information. This causes issues when writing specifics on different topics. There may be one piece of information on one article that either gets directly contradicted on another article or is omitted completely. This led to topics within Blockchain and even Ethereum being incomplete between drafts. One of the solutions to this would be to cross check references and talk with a knowledgeable source such as Dr. Shaffer.

Overall Lessons

Writing textbook prose is harder than it seems

Writing a textbook is very difficult -- you have to consider the reader's perspective when you are writing content. OpenDSA is oriented such that someone who knows nothing about Blockchain should be able to read your section, and none of us initially took that into enough consideration. We each had to go through multiple revisions to get our prose to be understandable to everyone. By the end of these revision stages though, we are able to write strong and understandable textbook prose.

Don't underestimate the amount of time it takes to research Blockchain

Initially looking at the project, it might not seem like much work. However, we all learned that the time investment to learn Blockchain to the extent of writing a textbook about it is much larger than it seems. There are a ton of details you must know to not have any gaps in your knowledge. And if you have gaps in your knowledge, it will be apparent to the reader. Most of the time was spent researching Ethereum, and only a small fraction on writing the textbook.

Future Work

There can be many additions and improvements made to the Blockchain ETextbook. This group focused on Ethereum, which is one of the real systems that currently utilizes Blockchain. In the future, more of Blockchain's concepts can be discussed such as motivation and more details pertaining to cryptocurrencies' wallets and smart contracts. As legislation is developed for the growing cryptocurrency market, content can be added. Also, more content can be added for demonstrations of cryptocurrencies that use consensus mechanisms other than proof-of-work.

Another improvement that can be made to this project is the addition of more interactive exercises which help the reader understand the concepts better. Initially, there were some exercises which asked the user to enter some data, and the website would process that information and display a specific output. These interactive exercises engage the reader more and help strengthen blockchain concepts. As more content is added, more interactive exercises can be created, so the reader can stay engaged as they read.

Acknowledgements

The team's gratitude is extended toward Dr. Clifford Shaffer, who is a Professor and Associate Department Head for Graduate Studies. Dr. Shaffer heads the OpenDSA project. We have been meeting with him on a weekly basis, and he provides us with critical feedback of our completed work and informs us of what modifications are needed in the project. Also, we would like to thank our professor, Dr. Edward Fox, for providing us this opportunity and for his assistance throughout the project. Lastly, we would like to thank Benjamin Par for discussing topics of consensus algorithms and crypto hacking with us.

Dr. Clifford Shaffer, the Associate Department Head for Graduate Studies in Computer Science:

Phone Number: 1(540)231-4354

Email: shaffer@cs.vt.edu or shaffer@vt.edu

Dr. Edward Fox, Professor of Computer Science:

Phone Number 1(540)231-5113

Email: fox@vt.edu

References

[1] L. Conway, "Blockchain Explained", *Investopedia*, 2021. Accessed on: Nov. 10, 2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp.

[2] S. Richards, P. Wackerow, K. Ziechmann, and R. Cordell, "What is Ethereum?", *Ethereum*, Dec. 06, 2021. Accessed on: Dec. 08, 2021. [Online]. Available: https://ethereum.org/en/what-is-ethereum/.

[3] Meet Patel, "Consensus Algorithms in Blockchain", *GeeksforGeeks*, 2021. Accessed on: Nov. 10, 2021. [Online]. Available: https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/.

[4] Khan Academy, "khan-exercises", *Github*, Dec. 4, 2018. Accessed on: Dec. 08, 2021 [Online]. Available: https://github.com/Khan/khan-exercises.

[5] Canva, "Graph Maker", *Canva*, 2021. Accessed on: Nov. 28, 2021. [Online]. Available: https://www.canva.com/graphs/

[6] J. Douglas et al., "Proof of Stake (POS)", *Ethereum*, 2021. Accessed on: Oct. 15, 2021. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/.

[7] J. Frankenfield, "What Is a 51% Attack?", *Investopedia*, 2021. Accessed on: Oct. 28, 2021. [Online]. Available: https://www.investopedia.com/terms/1/51-attack.asp.

[8] J. Frankenfield, "Proof of Stake (PoS)", *Investopedia,* Apr. 21, 2021. Accessed on: Oct. 12, 2021. [Online]. Available: https://www.investopedia.com/terms/p/proof-stake-pos.asp.

[9] V. Buterin, "Why Proof of Stake (Nov 2020)", *Vitalik*, Nov. 06, 2020. Accessed on: Sept. 28, 2021. [Online]. Available: https://vitalik.ca/general/2020/11/06/pos2020.html.

[10] R. G. Dillak, D. R. Suchendra, R. Hendriyanto, and A. A. G. Agung, "Proof of work: Energy inefficiency and profitability", *Journal of Theoretical and Applied Technology*, Mar. 15, 2019. Accessed on: Oct. 15, 2021. [Online]. Accessed: https://www.researchgate.net/publication/332767387_Proof_of_work_Energy_inefficiency_and_profitability.

[11] Cointelegraph, "History of ETH: The rise of the Ethereum blockchain," *Cointelegraph,* 2021. Accessed on: Dec. 01, 2021. Accessed: https://cointelegraph.com/ethereum-for-beginners/history-of-eth-the-rise-of-the-ethereum-blockchain.

[12] Cointelegraph, “What is Ethereum: A beginners guide to ETH cryptocurrency,” *Cointelegraph,* 2021. Accessed on: Dec. 01, 2021. Accessed: https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum-a-beginners-guide-to-eth-cryptocurrency.

[13] Cointelegraph, “Bitcoin vs. Ethereum: Key differences between BTC and ETH,” *Cointelegraph,* 2021. Accessed on: Dec. 01, 2021. Accessed: https://cointelegraph.com/ethereum-for-beginners/bitcoin-vs-ethereum-key-differences-between-btc-and-eth.