Gokoins Miner

Brennan Busza, Bryan Gonzalez-Moyano, Tyler Prehl, Ani Tapia, Matthew Weigand

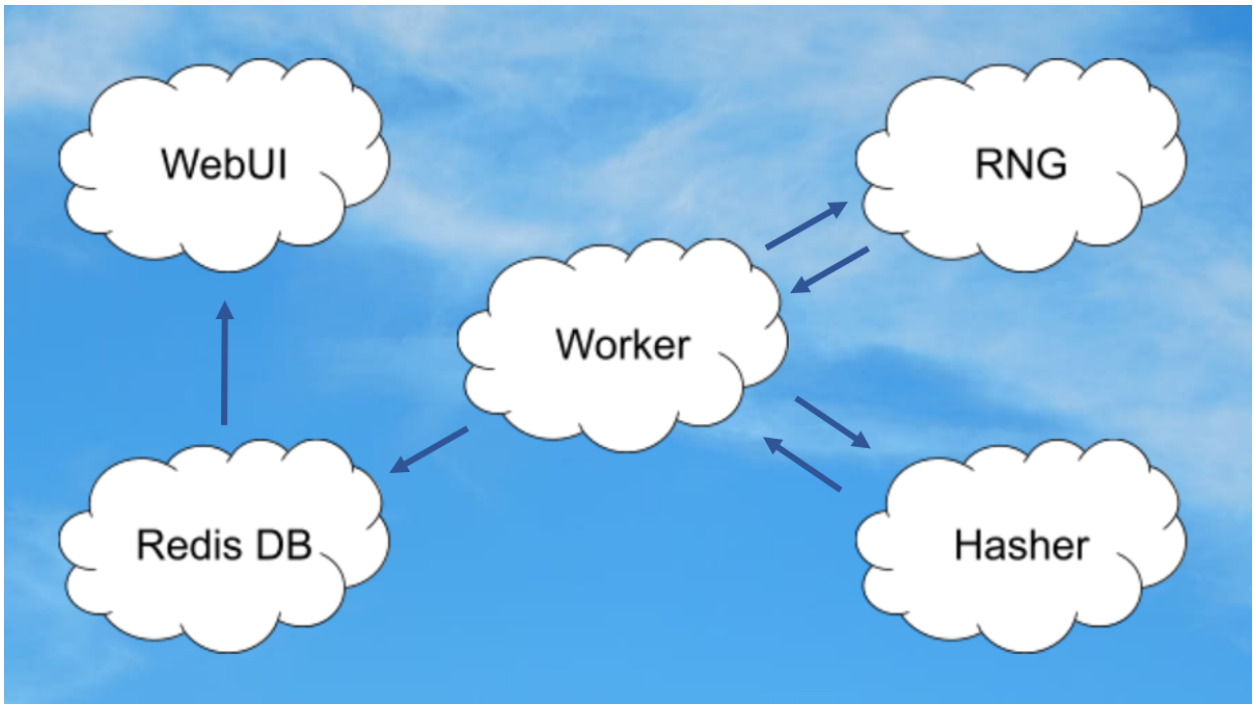
West Chester University of Pennsylvania

Table of Contents

Abstract

The purpose of this project is to help the authors learn more about continuous integration and continuous development (CI/CD) implementation in a cloud environment. The cloud-based project that the authors decided to use as a baseline for what is actually produced is Dr. Linh Ngo's ram_coin project, found at "https://github.com/CSC468-WCU/ram_coin". Through extensive research and training on tools and topics including Docker, Kubernetes, Jenkins, CloudLab, Linux, and GitHub, the authors aim to create a similar project that is fully built and deployed through a Jenkins pipeline that builds from our GitHub repository (link below) when changes are committed to the worker, hasher, rng, and WebUI branches. Because the authors are essentially creating a simulated cryptocurrency mining experience, the project name is titled after a famous fictional character named Goku to be "Gokoins."

Gokoins GitHub repository link:
https://github.com/Every-Villain-Is-Lemons/CSC468-Team-Project/



Chapter 1 - Vision

The overall goal of our coin miner process is to carry out a full stack deployment including CI/CD services that will entail creating random hashes to be solved and logged for users via a website.

The coin miner will consist of five distinct parts - the upgraded WebUI, a Redis database for storing data about the coin mining process, the Random Number Generator (RNG) which produces a random byte string, the Hasher that turns the byte string into a hash, and naturally a Worker that acts as the orchestrator between the RNG, Hasher, and Redis database. In the diagram above, there is a basic flow of how each piece interacts with the others. For example, the Worker makes requests to the RNG for random byte strings, which then gets returned to the Worker to be passed off to the Hasher for hashing.

The CI/CD services will entail a fully automated Kubernetes deployment into a CloudLab four-node environment from a web-hooked GitHub repository. The Kubernetes/Jenkins deployments will be separated into four distinct pieces, based on the Worker, Hasher, RNG, and WebUI sections of the project. Of note, the Redis deployment will be included in the Worker deployment. Additionally, a kubernetes-dashboard will also be deployed so as to view the actively running pods and services on the four CloudLab nodes. An extremely difficult aspect of this task is learning to work with Jenkinsfiles and YAML files to properly build and push the different components of the Gokoins project to our CloudLab nodes. Learning the syntax alone is a challenge in and of itself, and there will be need to master each well enough to execute all necessary functions automatically to take the CloudLab head node from empty to containing all of the parts necessary to run Gokoins for any user to access by URL.

Furthermore, while strictly adapting the Worker, Redis, and RNG pieces of the original ram_coin project by Dr. Ngo, Changes will be made to the UI and Hasher pieces. The WebUI will undergo a serious React upgrade to make the webpage more interesting to the user, and Hasher will be modified from its original Ruby format to Python, taking advantage of Flask API calls to do so.

Chapter 2 - Proposal

The purpose of this proposal is to break down each of the five components of the coin miner process - the upgraded WebUI, Redis database, Worker, Hasher, and RNG. For each component, we will discuss the tool used to develop it, what it will do (with a loose definition of how), and how it applies to the process as a whole.

The best place to begin the description is with the very start of the overarching process - the RNG Generator. In real blockchain technology, the headers of blocks are hashed which serves as the target hash to be solved. The person whose coin miner solves the hash first while adding in new data is rewarded in coins, and the solution hash becomes the next target hash to be solved as it represents the newest block in the chain. For our project, because we are not implementing an entire blockchain, we will use Python to simply choose a unique hash function (TBD) to hash randomly generated byte strings. These will serve as the target hashes to be solved by the Hasher. As a goal, these hashes should be sufficiently difficult to be solved such that solving them takes longer than a few minutes, if not many.

The Hasher will be provided with the target hash by the Worker (to be discussed more in depth below) in order to “solve” the hash. Solving the hash entails adding another randomly generated hash - serving as the “new data” being added to the blockchain - to the original hash, and then finding a nonce to tag onto the combined hashes such that when the whole combination of the original hash, the “new data” hash, and the nonce is hashed, the new hash is has a lesser value than the original hash. Using Ruby, we will create a process that loops through nonces - starting with the number zero and increasing by one in each loop - to be combined with the original and “new data” hashes to, through brute force, find a solution.

Orchestrating the interactions between the RNG Generator and the Hasher is the Worker. Also written in Python, the worker will act as the middleman that makes requests to the Generator for target hashes and then pushes the target hash to the Hasher to be solved, logging the time it takes for the hasher to solve the target hash in the process. Perhaps the most essential function of the Worker from the user perspective is that it also pushes all the information about each successfully mined coin to the MySQL database. Because all of these interactions are occurring in a cloud environment, special attention will be given to making sure the Worker properly makes requests to the Generator and Hasher, and properly uploads the data collected to the MySQL database.

The Redis database will collect five columns of information from the Worker, with each row representing a successfully mined coin. The five columns will include: the start time of the coin mining process for any given coin, the end time of that process, the difference between those times, the target hash that was solved, and the solution hash that was found. The database will also include columns for login information and how much money these accounts have spent on the workers. This data will be essential for the functionality we hope to provide our users through the upgraded web page.

Regarding the upgraded WebUI, the basic concept is to use HTML and CSS to produce a web page that pulls data from the Redis database to inform a user about the coin mining rate of the coin miner process. The “how to” would involve a simple query to the Redis database to determine the average amount of time it takes for each coin to be mined. The additional functionality we are looking to provide our users is the ability to find the coin mining rate during a specific period of time. The user would be able to input a start time/date and end time/date, which would then make a query to the Redis database about average coin mining times within the range dictated by the user. To polish off the user experience, we intend to implement cosmetic changes to our WebUI that will make the web page more interesting to view for the user. Cosmetic changes could include a coin mining rate “speedometer” that measures coins mined per hour, as well as an overall sharp and modern web page design, with the necessary input boxes and buttons to give users the ability to find out the coin mining rate during a specific

period of time. Our WebUI will also include a login/signup page since we intend to have multiple users. The username and password data for the login will also be stored in the Redis database.

As far as issues that could arise throughout the project, we are reserving our caution for a couple specific areas. While our group has strong experience in Java, Redis, and HTML/CSS, our knowledge of Python and Ruby are a bit lacking, so creating the backend of the process could prove more challenging. Additionally, pulling data from a Redis database for a WebUI is a new concept for all of us, so we will have to designate research hours for learning how to push data to/pull data from a Redis database constantly and in real-time. Our last concern at the moment revolves around designing a proper RNG Generator that will create hashes difficult enough that it takes the Hasher a fair amount of time to solve, but not so long that it takes days of working to produce just a couple coins. Finding that balance will be critical in our overall goal to produce a semi-realistic coin miner.

These five components are all individually crucial to the success of the overall coin miner process, and each will be treated as such. Future endeavors will include learning how to move the process from a local environment to the cloud in a CloudLab experiment environment.

Chapter 3 - Intermediate Milestones

Currently, there is a lot of progress in setting up the Jenkins server with different pipelines for each part of the project - the Worker, Hasher, RNG, and WebUI (the Redis deployment is included in Worker). This helps streamline multiple branches of our project without having to worry about manually rebuilding and redeploying each part constantly. Not only has this reduced any errors in our manual deployment, but vastly increases the speed at which we can test new features. During the process of building the pipelines, the kubernetes deployments in class and the sample “hello” Jenkins deployment helped immensely with understanding the pipelines. A bash script was then created to smoothly the deployment of Kubernetes and to help keep track of the steps in the Jenkins pipelines.

Analyzing Dr. Linh Ngo’s “Hello” GitHub Repository

When first analyzing the “hello” project (https://github.com/CSC468-WCU/hello), it took some time just to decipher that the only applicable aspect of the project to the Jenkins pipeline could be built for it was the go_app branch. Knowing this, we started analyzing each file of the go_app branch to make comparisons to Gokoins. Having the “hello” Jenkinsfile for reference was extraordinarily helpful for the construction of our own Jenkinsfiles due to our previous lack of background knowledge with Jenkins. We started realizing the slight ways we could change the “hello” Jenkinsfile to serve our needs for Gokoins. We found that the “Publish” and “Deploy” sections were necessary for Gokoins, and that the other minor differences between the “hello” Jenkinsfile and our Gokoins bash script were some sign-in and terminal-specific commands. For example, one of the commands in the “hello” Jenkinsfile is:

```
sh 'ssh -o StrictHostKeyChecking=no lngo@155.98.37.91 kubectl apply -f
/users/lngo/deployment.yml -n jenkins'
```

so we simply had to edit the “/users/lngo/deployment.yml” to “dashboard-insecure.yml” with our own CloudLab head node ssh information to begin the process of creating the Kubernetes dashboard. Finding out what other commands meant in the “hello” Jenkinsfile, such as “sed” and “scp” also helped us understand the true actions the Jenkinsfile was taking, allowing us to cut out unnecessary commands for Gokoins and restructure necessary commands for our own purposes.

Going through this process also helped us develop a stronger understanding of what configuring our Kubernetes cloud in Jenkins was actually doing for us. By analyzing the “hello” Jenkinsfile, we discovered how Jenkinsfiles inherits the container template data provided when configuring the cloud settings and its use, such as DOCKER_TOKEN and

DOCKER_REGISTRY for signing into DockerHub and accessing a designated repository when running the Jenkins pipeline.

Continuous Integration/Continuous Deployment Services

Building the Jenkinsfile also helped us build a better foundation for drafting Dockerfiles, as we had to slightly edit our Dockerfiles to fit our Jenkins pipeline commands, especially as it regarded accessing various files at different points in the build process. An issue we were struggling with was Jenkinsfiles' finicky double and single quote escapes to actually produce the desired result; we could not manage to successfully run our patch commands for changing the NodePorts for the WebUI and Kubernetes dashboard within the Jenkins pipeline. With some light guidance, we are now looking into other ways to set these values. For the Kubernetes dashboard, we successfully dictated its NodePort value by adding a NodePort value in the port specifications of the service section in the dashboard-insecure.yml file. Currently, we are struggling to manage the same for the WebUI, but are taking steps to ensure our success.

Back-End - Hasher

We made progress on changing the Hasher from Ruby to Python. First, we took time in understanding what each line of code meant within the Ruby file and have been looking at similar ways to write it in Python. With the DSL called Sinatra, it was easier to read and understand the Ruby code, even though it came with some difficulties while translating code into Python. One issue we had was trying to write the Hasher in Python. Since Ruby is a more generalized language and it was using Sinatra, it was simplified in a way that made it difficult to find a one-to-one translation. Fortunately, we were at least able to get our Ruby Hasher to successfully build and run on a Jenkins pipeline. Going forward, testing our new attempts at a successful Python version of Hasher will be made easier by our Jenkins pipeline, as it is general enough to run either. A possible point of issue with the Python Hasher is the Dockerfile. Our current attempts have revolved around trying to adapt the RNG Dockerfile to our Hasher needs since RNG is similar (but not identical) to Hasher in how it is accessed by Worker and what it needs to run, including Python and Flask. It is possible that our Dockerfile needs further addressing to prevent further errors with building our Python Hasher.

Front-End - WebUI

Changing up the WebUI so it is more related to our coin miner has also progressed well. We currently have a static version of the project to keep as a base point. Multiple group members have experience with HTML and JavaScript, so we believe this part of the project will not be too difficult to complete. While we have not made many design choices for color scheme, background, and etcetera, we have begun to create the important parts of the WebUI. The main sections are a welcome/ login/ signup screen and then the user's page for when they are actually logged in. The both main parts will be broken down into smaller pieces to help make it easier to create. The way we have decided to do this was by using Reactjs to set up a server and web page. We looked at the Jenkins documentation and found that there are simple ways to set up a React server through Jenkins by running simple scripts. That has also become a slight challenge because there are extra steps in the process to ensure that the web page can be brought up. Since the user logs in and a first time visitor can sign up, we need a connection to redis to store the user's data to ensure a smooth login each time. This has become slightly troublesome because none of our group members have any background with redis, so there has been a lot of research for this part of the project. While having a static web page is an accomplishment, the challenge comes with introducing all the moving parts to update along with the WebUI. One challenge with our project has been creating a log in/sign in pop-up for our WebUI. We have been having difficulties with setting it up, so that idea may be scrapped as a whole for our project in order to ensure the rest of the project is well made.

Since there is a login page, we are trying to have it so multiple users will be able to see basic information about one another. We plan on adding a pie chart that shows the total amount of coins split between each of the users to show who has mined the most. One issue with this is trying to get all user's information from redis into the WebUI and also updating it in real time. While we definitely want to keep this part of our project, we may have to scrap this idea because we may not have enough time to get a completed WebUI if we add this part to our project. We decided to make it a single page format with buttons that hide and unhide information. We are using Reactjs to help with the conditional rendering because it makes hiding information easier.

Finally, another challenge for the WebUI is the design aspect of it. We imported bootstrap as a way to help simplify the user interface for the time being. What we did not realize was in order to change the bootstrap design choices, we would have to make those changes before importing bootstrap. So currently, we will need to figure out color schemes and design choices and then uninstall bootstrap and make the necessary changes before reinstalling.

Conclusions

Overall, our project is looking to be a partial success. It will not manage to fulfill all of our lofty original goals that we had in mind, but it will be a fully operating project with end-to-end CI/CD services that we can call (mostly) our own.

Chapter 4 - Final State of Gokoins Miner

For the final state of the project, our experiment has been completely deployed on Jenkins. Although it is on Jenkins, it is not completely finalized as some parts of our project are not completely finished or up to the standard that we wanted it to be.

Continuous Integration/Continuous Deployment Services

As noted above, the Jenkins/Kubernetes CI/CD services aspect of Gokoins Miner is now a full and complete process. Originally, our Jenkins pipelines were referencing old Docker containers built locally for previous testing purposes, but this caused our Jenkins deployments to give us false-positive results. The original project that was containerized in pieces on the head node of our CloudLab experiment was able to be used via the commands in our previous Jenkinsfiles, so the containers we were building to DockerHub from our GitHub repository branches with our pipeline were never used. Fortunately, we were able to resolve this issue through several changes to the Jenkinsfiles and service and deployment YAML files, and our project now successfully builds using the Docker containers built using our GitHub repository branches.

In addition to our efforts to use the correct Docker containers, we also changed the service.yaml files of each branch (for Worker, Hasher, RNG, and WebUI) to include specified NodePorts upon building instead of using the randomly generated NodePort value on the Kubernetes deployment. With this success, our webpage can now immediately be found using the external IP address of any of the nodes of our CloudLab project on port 30080 after WebUI Jenkins build.

Back-End - Hasher

Although we left the RNG and Worker exactly as they are, to fully produce our originally desired goals of using Hasher to solve real hashes, we needed to convert the Hasher Ruby to be a Python file since we all lacked enough Ruby knowledge to build out our desired process with it. Previously, we failed to swap out hasher.rb with hasher.py, but the Gokoins Miner is now successfully running using hasher.py. Unfortunately, due to our time constraints and previous lack of progress with our Jenkins pipeline, we were unable to change the actual functionality of

hasher.py and worker.py to be more robust in how coins are mined, so the current status is simply a hasher.py file that exactly replaces hasher.rb in functionality.

Front-End - WebUI

Our updated graphical WebUI is successfully built and deployed through Jenkins for users to access freely, but we were unable to create all the features that we wanted originally. Our WebUI was going to include a login / signup feature, a button to pay for an increased mining speed, and a leaderboard showing all users and who has the most coins mined. We were unable to implement these features due to a lack of testing ability because of our setbacks with our Jenkins pipelines using old Docker containers. Currently, our “Welcome” (**Figure 4.1**) and “About Us” (**Figure 4.2**) pages are finalized the way we originally designed, but the remaining pages, the “User” (**Figure 4.3**) and “Other Users” (**Figure 4.4**) pages are incomplete in functionality and design. All figures can be found at the end of Chapter 4.

Although we originally wanted to implement the WebUI using React for its versatility and because of our strong background knowledge in its application, our current WebUI is designed solely using HTML, CSS, and JavaScript. Additionally, due to our lack of experience with Redis, we struggled with using the Redis database to build a different form of graph, so the current User Page utilizes the original graph, noticeably sized incorrectly as it can be seen where the gray box on the left side of the page is meant to be the background for the entirety of the graph. The Other Users Page also lacks the functionality to track users’ current mining statistics, partially due to the issue that we failed to ever successfully implement the ability for users to log in and have saved data specific to each user. The sign-in popup box appears after clicking on the “Login” button (**Figure 4.5**), but while clicking “Enter” brings the user to a “Login” Page (**Figure 4.6**), it is an empty page that contains no information excepting an error message.

Conclusions

Although the final state of our project is far from what we originally intended, we are proud to present our project in its current state. Our successes with the Jenkins pipelines, conversion of hasher.rb to hasher.py, and complete overhaul of the WebUI interface in only a one week timespan leads us to believe that given more time, we are confident in our ability to fully implement our original design. Unfortunately, because we will not have the resources to do so given our expiring access to CloudLab at the end of our school semester, we will never be able to prove it.

We are thankful for Dr. Linh Ngo and his guidance throughout the semester and for Jérôme Petazzoni for providing a baseline coin-miner project for us to learn from and augment as we fleshed out our Jenkins pipelines and redesigned various components.

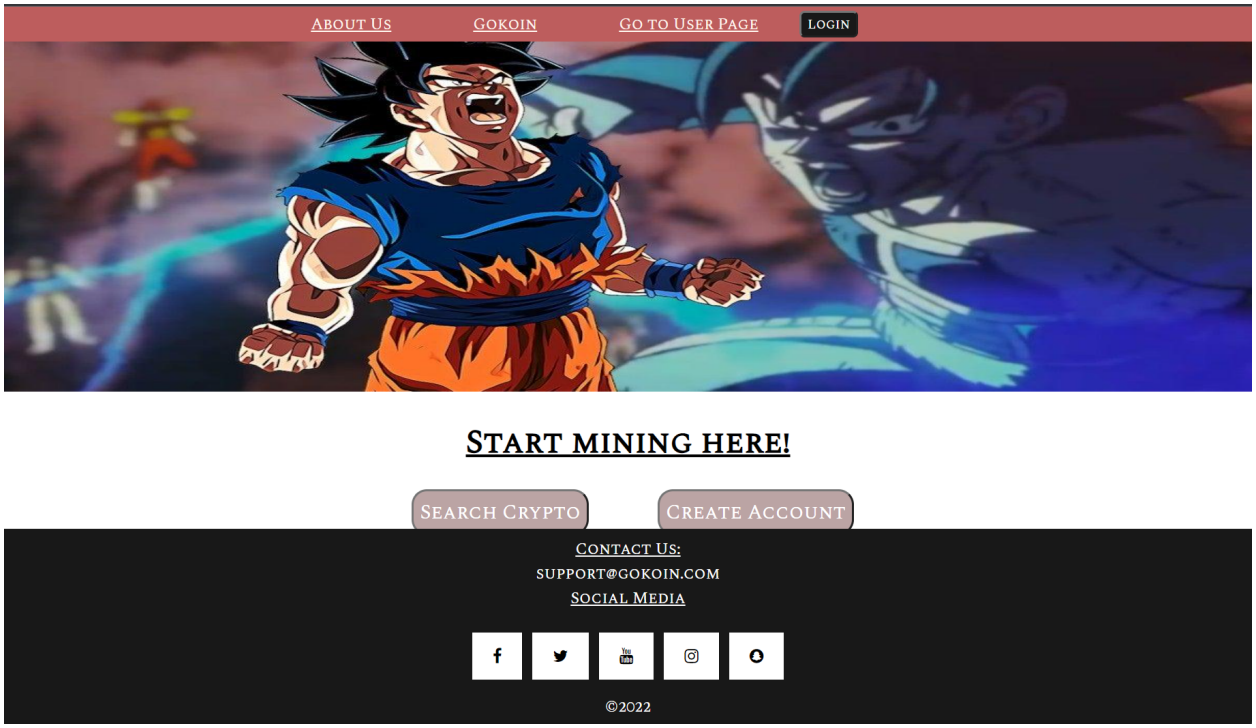


Figure 4.1 - Welcome Page

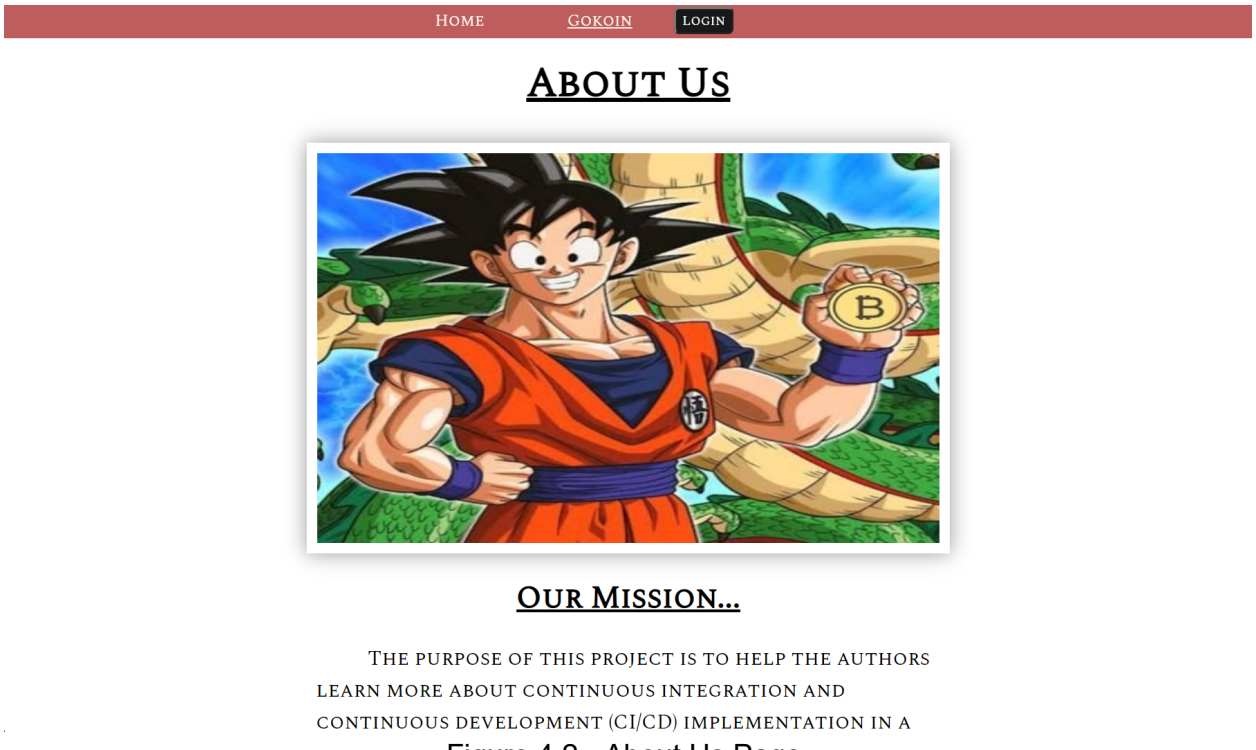


Figure 4.2 - About Us Page

Figure 4.3 - User Page

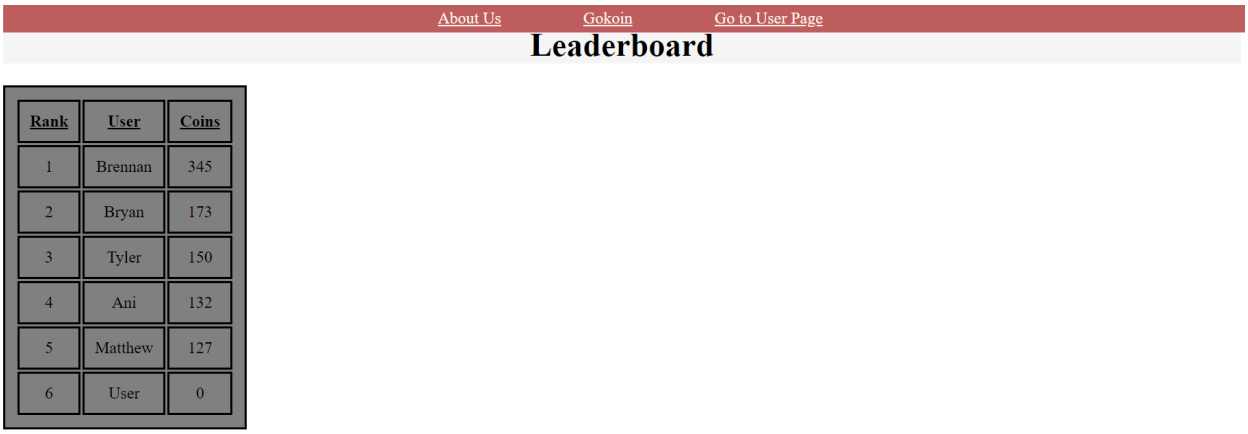


Figure 4.4 - Other Users Page

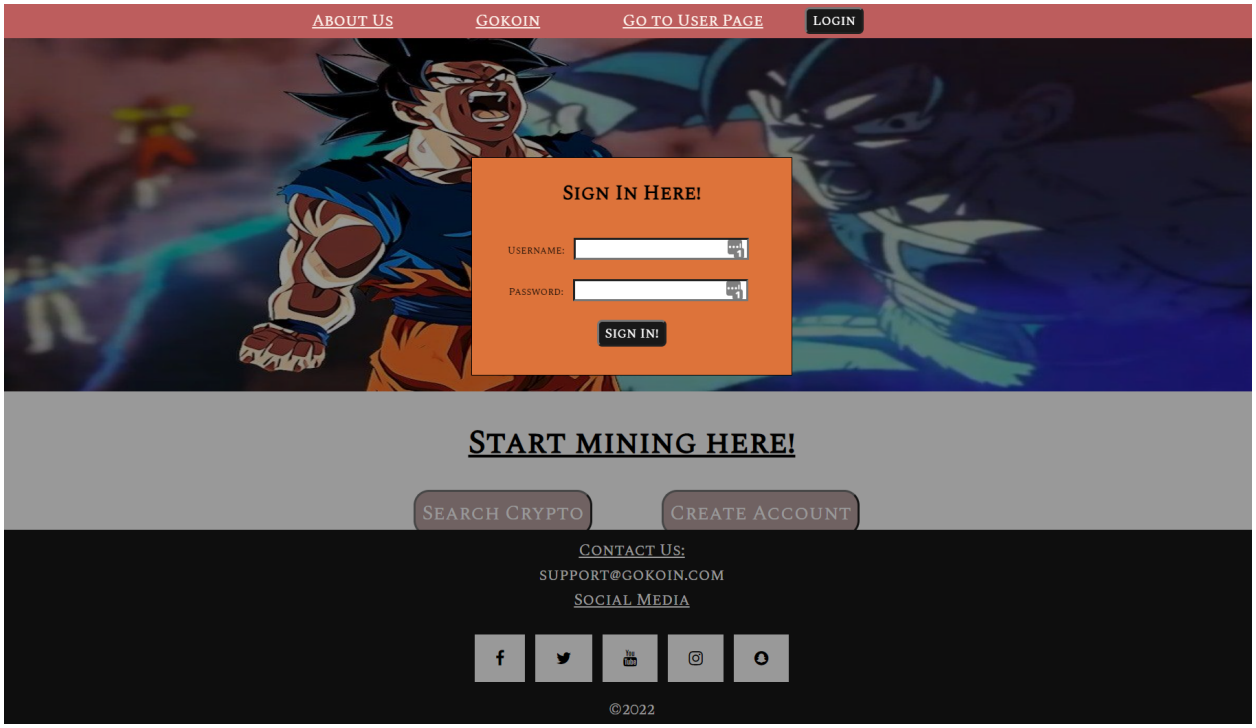


Figure 4.5 - Login Popup

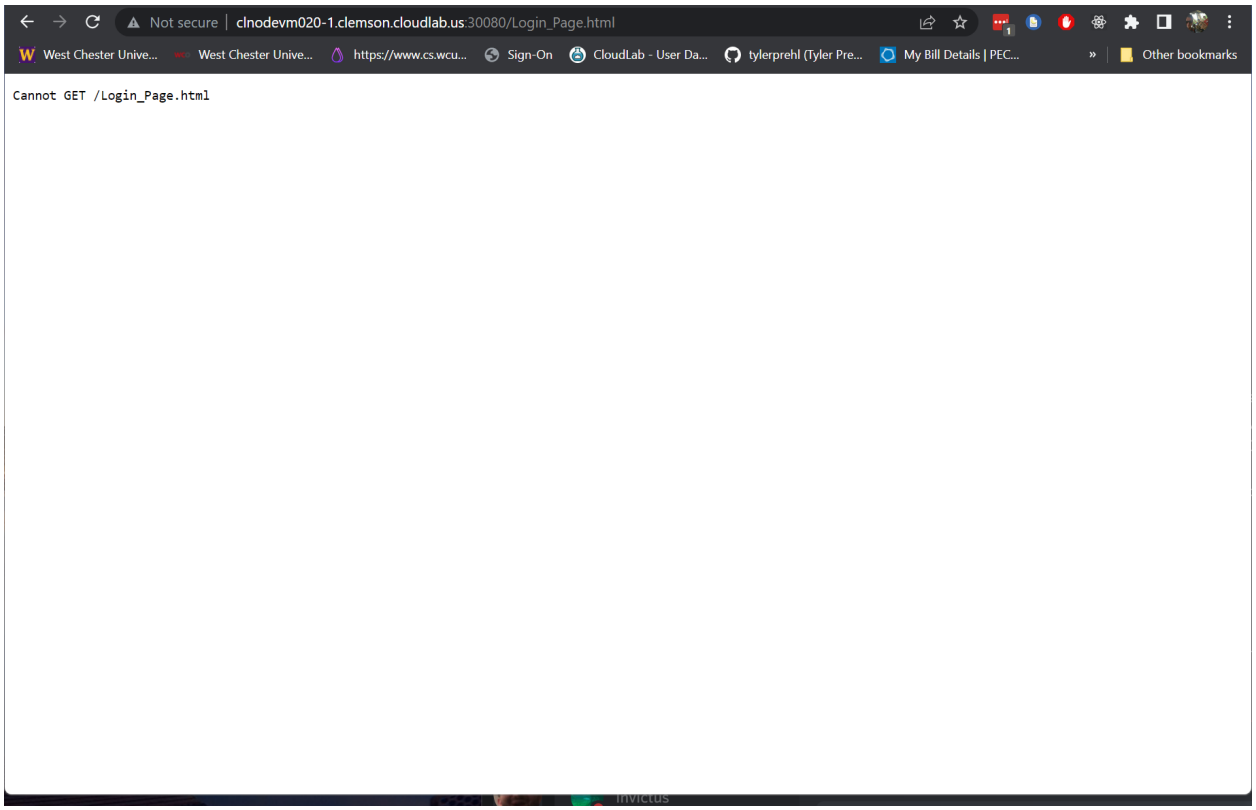


Figure 4.6 - Login Page

Brennan C. Busza

Cell: 610-766-1262 • Email: brennan.busza22@gmail.com • Springfield, PA 19064

EDUCATION:

- West Chester University (anticipated graduation date: May 2022); B.S in Computer Science
  Dean's List 2020 & 2021; Current GPA: 3.66
- Associates of Science Degree, Delaware County Community College, May 2020
  Major: Computer Science; Dean's List; Graduated with Honors
- Saint Joseph's Preparatory High School, Graduated with honors

COMPUTER LANGUAGES & RELATED COURSEWORK:

- Programming and Related Courses: C++, Java, Haskell, Linux, Python, Rust, SQL; Data Structures & Algorithms, Discrete Math, Operating Systems, Computer Security, Software Security
- Proficient in the Following Software programs: Microsoft Word, Excel, Power Point

TECHNICAL CLASSROOM PROJECTS

- **Recreation of Video Game "Pong"** — using GUI, programmed an interactive Pong game with controllable paddles for two players; Java.
- **Joe's Grocery** — programmed a simulated grocery store check-out experience which calculated customer volume, fluctuations in wait time, and number of cashiers needed to optimize resources; Java.
- **Student Catalog** — programmed a master database which compiled student names, majors, ID numbers, and related data, allowing cross reference searches; Java

EMPLOYMENT:

Marple Sports Arena | Marple, PA — Summer 2019, 2020 & 2021
Camp Counselor

- Oversaw safety and welfare of children ranging from 4-8 years of age; groups of up to 30 kids.
- Recognized as of "Counselor of the Week" and Winner of "Best Counselor of the Summer" Awards each year.

320 Market — Deli & Restaurant | Swarthmore, PA — Summer 2018
Deli Counter

The Second City Hollywood | Los Angeles, CA — May 2017 - September 2017
Intern, Summer Camp Assistant & Student

Delaware County Community College | Newtown Square, PA — January 2017 - May 2017
Magazine Editor for *Pegasus* — DCCC Literary Magazine

Tague Lumber | Media & Kennett Square, PA — June-Sept 2014, 2015 & 2016
Hardware Assistant

American Diabetes Association's *"Camp Freedom"* — Summer 2013 & 2014
Camp Counselor | Schwenksville, PA

ACTIVITIES, ACHIEVEMENTS & AWARDS

- **Member of WCU Computer Science Club & WCU Competitive Computer Science Club** | 2020 & 2021
- **Participated in the Annual CCSE Programming Competition 2020** | Team: West Chester University
- **Personal Computer Build** | using – AMD Ryzen 5 2600, tomahawk b450, Radeon RX 570
- **Quell Foundation Scholarship Winner, Fall 2019** | Quell Foundation National Award
- **Counselor of the Season, Summer 2019 & 2020** | Marple Sport Arena, Summer Camp Program
- **Counselor of the Week, Summer 2019, 2020 & 2021** | Marple Sports Arena Camp Program
- **Cypher Prime Game Jam** | Interviewed co-owner of local Game design company at Game Jam; played game.
- **International Game Developers Association** (IGDA) Student Member, 2013

BRYAN GONZALEZ-MOYANO

6108885586 | bryan.agm@icloud.com | 103 Stephanie Lane, Collegeville, PA 19426

PROFESSIONAL SUMMARY

Technical Analyst that likes to learn and can create custom customer solutions. Proficient with escalations, understanding of complex topics, great interaction with customers, a strong believe in communication, team productivity and collaboration.

SKILLS

- Troubleshooting and diagnosis
- Technical reporting
- Workflow improvement
- Fluent in Spanish
- VoIP and IT support
- Microsoft Office Applications
- Java, HTML5, SQL
- Technical Documentation
- Application Support

EXPERIENCE

Senior Technical Analyst - Evolve IP, 2019 - Current

- Analyzed PCAPS to identify call quality issues that include call drops, choppy audio, echo static and one way audio
- Collaborated with Engineers to identify root cause of incidents
- Troubleshoot, reviewed and modified network equipment to ensure optimal call quality and network connectivity
- Troubleshoot Polycom and Yealink device settings along with creating, deleting, and modifying phone configurations

Preferred Client Services - Iron Mountain, 2016 - 2018

- Manage customer accounts for high level customers (Xerox, Chrysler, United Rentals)
- Work as a team with sales representatives to update accounts according to sales needs and
- Manage escalations and any issues that customers are having on a personal level
- Manage customer's billing and provide custom solutions

Customer Care Representative II - Iron Mountain, 2015 - 2016

- Iron Mountain Connect Support team
- Created specialized reports for both Internal and external contacts.
- Assist customer with Billing Issues and finding the root cause
- Assist customers with Account updates
- Participated in the mentor program assisting new hires

Customer Care Representative - Iron Mountain, 2014 - 2015

- Process customer's orders in timely manner
- Follow the appropriate procedures to resolve customer's issues
- Provided the Cornerstone Team with afterhours support. (After 7PM)
- Part of the PNC Bank dedicated orders team.
- Data Management orders team.

EDUCATION

Bachelor of Science in Computer Science
West Chester University of PA, West Chester 2022

Associate of Science in Computer Science
Montgomery County Community College, Blue Bell 2019

Tyler T. Prehl

tyler@prehl.us Allentown, PA, 18104
Cell: (484) 619-1389 linkedin.com/in/tylerprehl/
github.com/tylerprehl

Education

West Chester University of Pennsylvania, West Chester, PA — Anticipated Graduation: May 2022
Current GPA: 3.96 — B.S. in Computer Science, Mathematics Minor, Honors College
Year: Senior — Board of Governor's Scholarship Recipient, Montemuro Award Recipient

Computer Science Skills

Languages:	Java, Python, SQL, OCaml, C
Tools:	GitHub, VSCode, NetBeans, IntelliJ IDEA, Microsoft SSMS, Jenkins, VMWare, PuTTY
Platforms:	Windows, Linux

Work Experience

Computer Aid, Inc., DevOps Engineer Intern — June 2021 - July 2021

- Deployed IBM Integration Bus (IIB) major releases to a UAT environment using Jenkins, and pushed occasional hotfixes to production with direct client observation
- Learned how to manage databases in SQL Server Management Studio and how to create effective and efficient queries
- Developed skills in configuring Linux servers manually and was introduced to automatic configuration using Ansible
- Strengthened my professional leadership skills through trainings in interpersonal communication, emotional intelligence, team collaboration, and communication foundations

West Chester Univ., Learning Assistance & Resource Center, Peer Tutor Coordinator — August 2019 - Present

- Conducted College Reading & Learning Association-certified training sessions to train future tutors
- Review key concepts of physics and various math subjects to increase students' understanding of the course material
- Introduce and cultivate helpful study skills to help students become independent learners

West Chester Univ., Residence Life & Housing Services, Resident Assistant — August 2019 - March 2020

- Informed residents of university and Residence Life policies and documented all incidents
- Fostered a friendly community atmosphere on my assigned floor and throughout the building
- Provided residents with learning opportunities about diversity, health and wellness, sustainability, and student development
- Prepared to handle emergency situations ranging from fire emergencies to suicide prevention

Achievements / Leadership

West Chester University

- Secretary, WCU Competitive Programming Club — May 2021 - Present
- Member, WCU Computer Science Club — August 2020 - Present
- Founding President, WCU Fitness Club — January 2021 - Present
- Member, Upsilon Pi Epsilon Computer Science Honor Society — November 2020 - Present
- Treasurer, WCU Golden Gamers — January 2021 - May 2021
- Treasurer, West Chester University's Serpentine Yearbook Club — January 2019 - June 2019
- College Reading & Learning Association Master Certified Tutor, Level III — November 2020
- Completed the West Chester University Leadership Challenge Series 2018 — October 2018
- Traveled to South Africa to work with the beneficiaries supported by the Honors Student Association — June 2019

Scouts BSA Troop 12

- Eagle Scout - Scouts BSA — February 2018
- Assistant Scoutmaster (adult advisor) for Troop 12 — December 2017 - December 2020
- Completed several High Adventures, including a week-long sailing trip and two separate two-week backpacking trips

Volunteer Experience

WCU Golden Gamers, Extra Life — April 2019

Raised $600 for the Children's Hospital of Philadelphia through the WCU Golden Gamers 25 Hour Gaming Event by acquiring sponsors to play games for 25 hours without sleep. The WCU Golden Gamers collectively raised over $9,900.

Eagle Scout Projects & Troop 12 Service Projects — 2014 - 2018

Completed a church kitchen remodel including painting, replacing flooring, and cabinet/shelving repairs for my own Eagle Scout Project. Assisted with other Eagle projects including replacing 20-year-old carpet with wood laminate flooring in a church fellowship hall and planting trees to replenish habitats for the National Wild Turkey Federation.

Ani Tapia
Philadelphia PA 19131
267-808-4021
Tapia.ani28@gmail.com

Education

West Chester University of Pennsylvania — Expected Graduation Fall 2022
Computer Science Courses: — Current GPA: 3.61
Computer Sci III, Computer Security & Ethics,
Computer Systems, Foundations of CSC,
Discrete Math, Statistics

Delaware County Community College — Graduated: Fall 2020
Computer Science Courses: — GPA: 3.84
Network Communications, Intro to Computer Science,
Intro to Java Programming, Intermediate Java Programming,
Intro to C++, Intro to Information Technology,
Data Structures and Algorithms

Employment

Whole Foods Market Philadelphia, PA 19147 — 2020-2021
Amazon Shopper

- Prepared grocery orders for delivery
- Shopped the store for customer items
- Used smartphones to manage apps and scan bar codes
- Examined order for quality
- Communicated with customers about their orders

Additional Skills

- Java
- C
- Haskell
- Microsoft Word
- C++
- PowerPoint
- Excel
- Bilingual (English-Albanian)

Matthew George Weigand
mattweigand99@gmail.com
443-655-7737
2402 Hickory Hill Road, Chadds Ford, PA
Github: github.com/MatthewWeigand99

Education

West Chester University — West Chester, PA
Cumulative GPA: 3.62 — Computer Science Major
Year: Senior — Expected Graduation: Spring 2022

Moravian College — Bethlehem, PA
Transferred — Fall 2018 - Spring 2019
-Nursing Major: GPA 3.54 - Dean's List
-Recipient of Moravian College Provost's Scholarship

Leadership

Sound Crew, Unionville High School — Kennett Square, PA
Sound Designer, Head Mic Wrangler — Fall 2015 - Spring 2018
-Created sound effects and worked with microphones for four shows each school year. Worked during school assemblies to ensure microphones and presentations were set up and running properly.

The 1742 Experience — Bethlehem, PA
Volunteer — August 2018
-Went to multiple sites around Bethlehem, Pennsylvania to volunteer and give back to the community. Sites included the YMCA, two animal shelters, and two elementary schools.

Pledge Class Treasurer for Delta Tau Delta — Bethlehem, PA
Treasurer — February 2019 - May 2019

Employment

Moravian College Information Technology Department — Bethlehem, PA
Information Technology Technician — September 2018 - May 2019
-Answered phone calls and emails about technical issues. Helped fix issues and taught the users troubleshooting methods. Fixed printers and Apple products.

Wendy's Fast Food Company — Kennett Square, PA
Shift Supervisor — August 2021 - present
-Managed the restaurant and ensured proper food handling while on-site. Managed the money after shifts and counted food supply to ensure proper ordering was done.

Computer Science Skills

Languages: Java, JavaScript
Tools: Eclipse, Github, Visual Studio Code
Operating Systems: Windows, Mac