Codemonkey

<other team member> and Daniel Tomov

Landstown High School

2022-2023

Mrs. <teacher name>

Contents

Abstract

The cybersecurity field has 750,000 open job positions in the United States with 60,000 being in Virginia (Cyberseek, 2023). With the growing amount of technology that encompasses everyone's daily lives, more people are needed in the cybersecurity field. More valuable information is being stored online than ever before. Almost every organization is moving towards paperless documents and putting all of them online. To get more people into the field, our team created an online course that teaches users how to code in Python, an essential coding language in the field of cybersecurity. With many open cybersecurity jobs, a vulnerability is created in America's cyberspace. If too many jobs are open, then there are not enough people defending cyberspace, leaving openings for hackers to steal valuable information. Our project is meant to combat the openings in the cybersecurity field and reduce the risk of storing information in cyberspace. This paper goes in-depth with the problem, project objectives, and possible solutions to the problem.

Problem Statement

The job gap in the field can lead to more insecure systems. The Bureau of Labor Statistics projects an increase of 35% in cybersecurity jobs between 2021-2031. That's a projected 56,500 increase in job openings. The shortage of cybersecurity professionals contributes to this problem. Additionally, increasing the number of professionals is not enough; it is also imperative to ensure that these professionals are highly trained and knowledgeable about the industry. If workers in the field are unable to understand how their technical actions affect the over-security posture of an organization, more hacks and system intrusions could occur. Practical work is almost a requirement to truly ensure that systems are protected

Introduction

When the group first came together, we had known each other through the academy at Landstown High School for a few years. We were part of the cybersecurity strand, where we learned cybersecurity terminology and how to secure systems. While doing this, we were tasked with researching information about the cybersecurity field. It shocked us to find out that there are over 750,000 open job positions in the United States and 60,000 in Virginia (Cyberseek, 2023). Virginia is the third state with the most openings, after California and Texas. The purpose of these jobs is to ensure the security of systems and the information contained within them. Methods of securing a system include installing firewalls that restrict the specific transmission of data, fixing any code that can be exploited, and actively defending against attacks. With a lack of professionals defending and patching systems, every person's information who is on that system is now at risk. If that data is obtained by someone with malicious intent, the life of the person whose information was obtained can be in jeopardy. Depending on what information was obtained, the malicious attacker could conduct identity fraud on the victim, heavily impacting the victim's life. So, when tasked with helping solve a problem in our local community, we saw it fitting to help provide a solution for one that affected our state. This solution is Codemonkey. With this online course, we aim to educate elementary and middle school students on how to program in Python, a widely used and versatile coding language. Teaching students from an early age how to program will gain their interest in the cybersecurity field, so when they apply for a job, they will have a strong foundation to excel.

Project Objectives

The objective of this project is to create a functioning online course to give elementary and middle school students a foundation in the Python programming language and stimulate their

interest in cybersecurity. We chose Python because it is “used across fields, including web development, machine learning, data analysis, server management, software building, and more. It’s incredibly versatile and considered easy to learn, making it a great fit for beginners because its simple syntax is closer to natural language” (Tech, 2022). In order for the group to effectively complete the project, Daniel did the back end while <redacted> did the front end. In the project, the back end represents the system on the server side. This includes the servers running the Codemonkey program, the email server, the reverse proxy to forward traffic from a domain to the Codemonkey program, and communication between the servers and the front end. The front end represents everything a user sees on their screen. For example, the modules and questions in each module and the layout of the website.

The team started development at the beginning of October 2022 and gave themselves a deadline of March 2023 to complete the project. The team decided to split the development process into three different phases: research, development, and final adjustments. The first phase lasted one week, during which the team looked at similar projects. Phase two consisted of developing the project, which lasted twenty-five weeks. Phase three consisted of making final adjustments to the project, which lasted four weeks. Test runs were conducted to make sure that the project was user-friendly and bug-free. Once the program was completed, the team gave the ODS Cybersecurity Club sponsors a test server to test the project with students. Our benchmark for success was a 50% course completion rate. After the club meeting, which lasted ninety minutes, the team compiled the data and found that 65% of the students who participated in the meeting that day had completed at least half of the course. This was higher than our original expectations. These statistics show high engagement from elementary and middle school students and a strong foundation in cybersecurity as they were able to complete many of the

foundational questions. The students enjoyed their time using the website, which further strengthens our belief in their interest and engagement in coding. We plan to continue developing Codemonkey to create more difficult lessons and challenges for more experienced coders and older students. We also plan to add more languages to the website as businesses desire experience and knowledge in multiple languages and different languages excel at different tasks. By continuing to educate these students, we are continuing to prepare them for a future job in the cybersecurity field.

Body

Identify the Problem

The first step of our project was finding a local issue that we could solve by applying the lessons and practices that we learned throughout our four years in the academy. We thought back to our school experiences and recognized the lack of education on cybersecurity and information technology topics in elementary school, middle school, and high school. Lessons were taught about how to be safe on social media and how nobody should reveal too much information as it can be used against them. However, there is significantly more to cybersecurity than watching what someone posts on the internet. Upon further research, we discovered data that shows an absence of workers in the cybersecurity field. As aforementioned, there are over 60,000 openings in the cybersecurity field in Virginia alone (Cyberseek, 2023). Through this hole in our lives and the hole of workers in the cybersecurity field, we discovered our problem: schools are not properly educating or introducing students to cybersecurity and information technology topics that could affect their life.

Explore Previous Solutions

While exploring other solutions, we discovered that everyone had a feature that was good but didn't have a feature another website had. For example, Sololearn gives great explanations, however, it limits users on the number of questions they can get wrong before having to wait to continue further. To remove this limit, users have to pay. Runestone Academy provides feedback to the user on what should be implemented in their code and what the expected output is. However, it does not give good explanations.

Design a New Solution

Codemonkey takes all the benefits from other online courses and eliminates the negatives. The course uses easy-to-understand language, a friendly interface, no personal data collection, and no paywalls.

Testing

The team gave a private URL to the leaders of the ODS Cybersecurity Club to test with students at ODS. Each student was given the length of the meeting, 90 minutes, to complete as much of the course as they could. During testing, the leaders and the team were able to see in real-time the data which was collected about each student. The data collected includes the percentage of the course each student completed and the percentage of students who completed each module. Using the data, we can see which students struggled and which modules were more challenging.

Discussion

We found that all of the students in the ODS Cybersecurity Club completed more than half of the course (refer to Figure A1). Students completed more of the course at the beginning than at the end (refer to Figure A1). This could presumably be due to the fact that they had only

ninety minutes in the course before the meeting was over. As learning a coding language takes time and practice, the team finds it reasonable that the modules towards the end of the course were not completed by all students (refer to figure A1). Before being exposed to Codemonkey, the students had mixed feelings about programming and about pursuing a career in cybersecurity (refer to Figure A2). After the meeting, a majority of the students felt better about coding (refer to Figure A2). However, the average wants to pursue a career in cybersecurity went down after using Codemonkey (refer to figure A2). This was expected because the team can not expect everyone to understand coding and cybersecurity. It is normal for students of that age to change their minds about which job they want to pursue in the future.

Project Budget

No funding was required for the creation of this project. All resources used were free and available for use. Refer to Figure B1 for the corresponding expense report.

Project Description

When a user wants to start using Codemonkey, they go to the website codeomonkey.tk and make an account on the register page. The register page requires users to provide a username, email, and password. For security reasons, the form only accepts valid emails and passwords greater than eight characters. After the user sends the credentials, they are sent an email with a verification code. The code expires after five minutes. This prevents malicious users from creating many fake accounts and crashing the server.

After they verify through the link, users are given the pre-program survey, where they are asked how excited they are about coding and how likely they are to pursue a career in the cybersecurity field (see Figure C9).

The user is then brought to the modules page, where they are presented with all the modules from the course (see Figure C10). They can choose which module to do freely. If they want to start from the end of the course, they can.

Individual module pages have text, code blocks, and questions (see Figure C11). The text on the page is explanations about the information in the module. It can be formatted with **bold** or *italics* if the administrator of the site needs to emphasize something in the text. Codeblocks provide the administrator to show users example code without saving it to their profile. Questions allow the administrator to test the user's knowledge. Questions are used to gauge the percent completion of a certain module. Once all the questions on a module are answered, a request is sent from the front end to notify the back end that the user has completed the module. The back end saves this information and displays a green checkmark on the main modules page to the user to indicate they have completed the module.

When the user completes all of the modules, they are automatically redirected to the post-survey, where they asked the same questions as on the pre-survey: how do they feel about programming and will they pursue a career in cybersecurity.

Administrators have access to a page where they can edit the text, code blocks, or questions on module pages. They can also view statistics about the program (see Figures A1 and A2). Statistics such as the percentage of the course each user has completed, the percentage of users who completed each module, and results of both surveys (see Figures A1 and A2).

Conclusion

While making this project, the team learned that making an online course to teach students how to code was not as easy as they thought it would be. However, the team started earlier and was able to finish on time. Over the development of Codemonkey, the team learned

about important cybersecurity concepts which they only learned about before. Concepts such as remote code execution was implemented in the project and it was quickly learned of its dangers. Overall, the team is proud of the product they created and has plans to continue to improve on it in the future.

Academy Courses/Pathway

Daniel Tomov

While the project was created for elementary and middle school students, it could be given to courses at the Technology Academy. Programming in Python is taught in LT STEM Cybersecurity Fundamentals. The program can be used to alleviate the teacher from teaching an entire class to a program. Normally, a teacher would have to show on the board how to program. However, with Codemonkey, the teacher is free to help students one-on-one as they have questions.

Codemonkey can also be used to start an introductory course to programming at the Technology Academy. The teacher will spread the workload over the course of the year and add in assignments where necessary. Codemonkey covers the same curriculum as AP Computer Science A, instead it does not teach Java, but it teaches Python.

Daniel Tomov has earned multiple certifications while at the Technology Academy. In ninth-grade IT Fundamentals, he earned the Internet and Computer Core Certification (IC3). In tenth-grade Computer Information Systems, he earned the Microsoft Office Associate Certification, which requires candidates to pass certification exams for Microsoft Word, PowerPoint, and Excel. In eleventh-grade Database Fundamentals, Daniel earned the MTA Database Fundamentals certification and would have earned the MTA Cybersecurity Fundamentals certification if Microsoft did not cancel it earlier the same year. He also earned the

CompTIA Security+ certification because the LT STEM Cybersecurity Operations teacher at the time offered him a voucher. In twelfth-grade Cybersecurity Operations, Daniel earned the CompTIA Network+ certification because he already had the Security+ which everyone else in the class took. See Appendix D1 for all of Daniel's certifications paid for by the Technology Academy.

<other team member>

Landstown High School's Governor's STEM Academy has taught me many lessons that I will later apply to my future career. Within the academy, there are several paths that a student can take to hone their skills in a specific field. Examples of this include 3D modeling, game design, cybersecurity, etc. I chose the cybersecurity strand, which consisted of two basic cybersecurity courses, because I found that to be the most interesting. By taking these classes and by being a part of the academy, I have found what I want to do as my future career. The topics within those classes grasped my attention and caused me to want to learn more. Codemonkey aligns with topics that are discussed in the classes because a major part of designing a website for public use is the security of the product. Codemonkey collects a small amount of data and without any security, small, but still revealing, pieces of information could be obtained by those with malicious intent. Through the cybersecurity courses, I learned effective ways to protect the website.

The Governor's STEM Academy not only taught me topics that I will apply later in life, but it also gave me the skills and certifications that will help me succeed in a future career. Cybersecurity classes have taught me how to effectively work in Linux to complete various tasks. These classes also allowed me to get three certifications: the Microsoft Office Specialist in Excel certification, the Internet and Computer Core Certification (IC3), and the CompTIA

Security+ certification. All certifications are in Appendix D2. This proves my skills to companies and will help me succeed.

Reflections

Daniel Tomov

I personally enjoyed doing the Senior Design Project. I like to program for hours without stopping. I like the feeling of being challenged by a problem that can be solved using programming. However, I would not like programming as a career because I would be doing it every day and it would get repetitive. I program once in a while and when I feel like it. If I had to restart this Senior Design Project, I would not choose to make a website that teaches students how to code. Simply because it was too much. The amount of work on the back end was a lot more than the work on the front end. Implementing functions such as saving code, checking code, session tokens, accounts, saving completions, and setting up the email server, all took time. I liked doing each of these, but I definitely struggled with some of them. I would recommend to someone trying to duplicate my project to not duplicate it. They could make a Capture the Flag (CTF) framework that operates very similarly to Codemonkey. Both have to save accounts and sessions. Both have to collect statistics about the users, just in different forms. Both have to send emails and send/receive data between the front end and back end. However, a CTF framework will not have to verify user code and save it; only answers and those are easy to check. My experiences in the Technology Academy have not influenced the idea for my project much. We looked at previous projects and decided we did not want to make a CTF or a game that teaches cybersecurity. After looking at the number of jobs open in the cybersecurity field, we decided that teaching elementary school and middle school students would close that hole.

<other team member>

The Senior Design Project was a mighty task to take on. It took large amounts of time and perseverance through stressful situations where I had to balance regular school work with developing a website and all of the other aspects of the project. I also had meetings that took up valuable class time from the academy courses. This made preparing for the Security+ exam significantly more difficult as getting through all of the content was much more difficult. However, I was able to get through everything and pass the certification exam, making everything worthwhile. I took out just as much as I put in for the project. The Senior Design Project taught me new lessons that I will carry with me.

Appendices

Appendix A

Figure A1: Current program status page

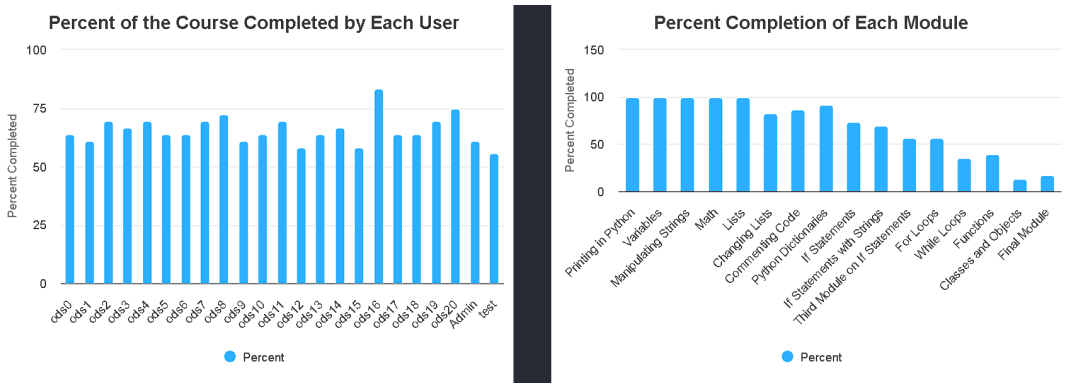
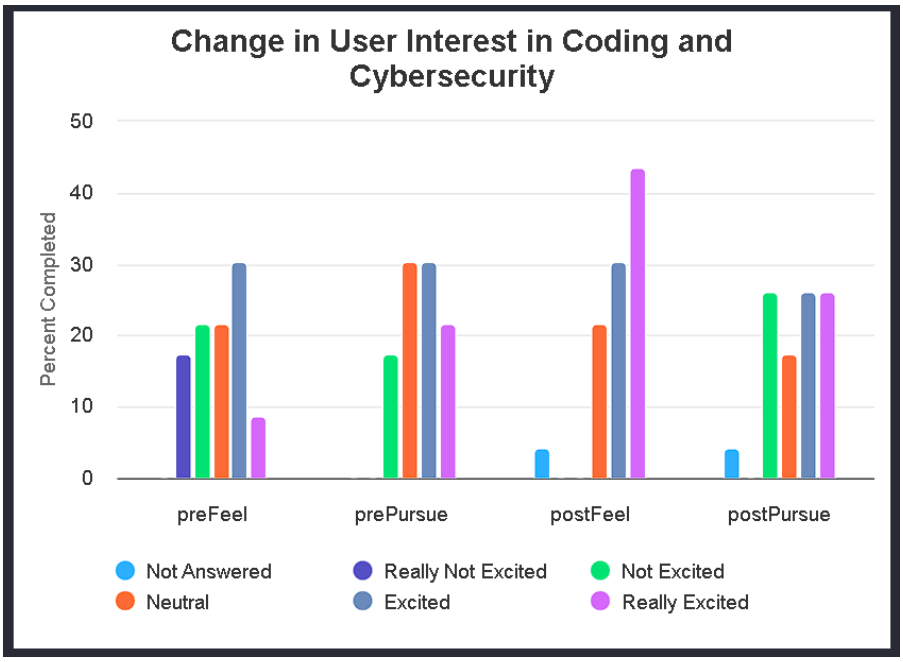
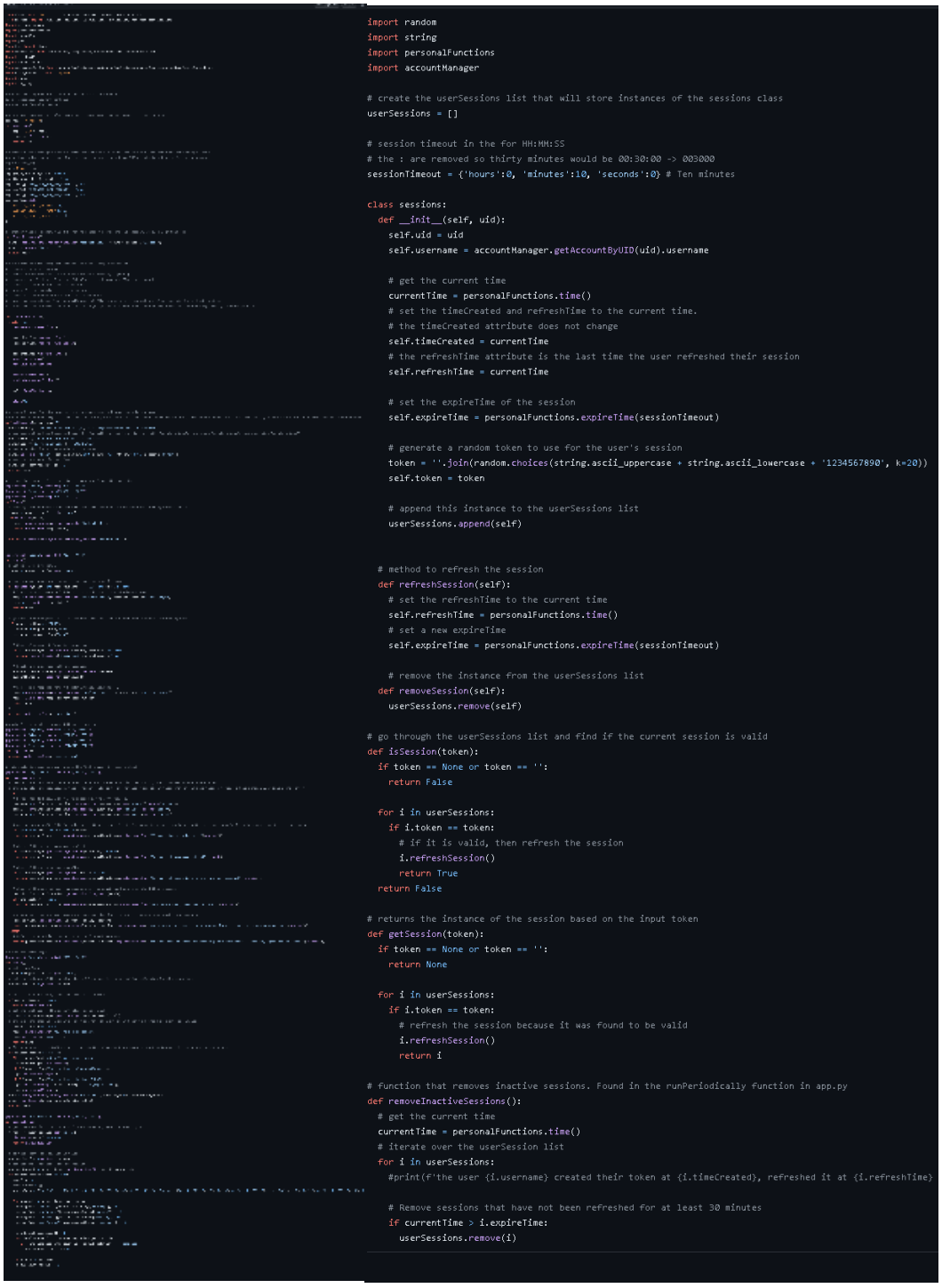


Figure A2: how students feel about programming and whether they will pursue a career in cybersecurity before and after using Codemonkey


Change in User Interest in Coding and Cybersecurity
Percent Completed
50
40
30
20
10
0
preFeel
prePursue
postFeel
postPursue
Not Answered
Really Not Excited
Not Excited
Neutral
Excited
Really Excited

Appendix B

Figure B1: backend code of Codemonkey



```
import random
import string
import personalFunctions
import accountManager

# create the userSessions list that will store instances of the sessions class
userSessions = []

# session timeout in the for HH:MM:SS
# the : are removed so thirty minutes would be 00:30:00 -> 003000
sessionTimeout = {'hours':0, 'minutes':10, 'seconds':0} # Ten minutes

class sessions:
  def __init__(self, uid):
    self.uid = uid
    self.username = accountManager.getAccountByUID(uid).username

    # get the current time
    currentTime = personalFunctions.time()
    # set the timeCreated and refreshTime to the current time.
    # the timeCreated attribute does not change
    self.timeCreated = currentTime
    # the refreshTime attribute is the last time the user refreshed their session
    self.refreshTime = currentTime

    # set the expireTime of the session
    self.expireTime = personalFunctions.expireTime(sessionTimeout)

    # generate a random token to use for the user's session
    token = ''.join(random.choices(string.ascii_uppercase + string.ascii_lowercase + '1234567890', k=20))
    self.token = token

    # append this instance to the userSessions list
    userSessions.append(self)


  # method to refresh the session
  def refreshSession(self):
    # set the refreshTime to the current time
    self.refreshTime = personalFunctions.time()
    # set a new expireTime
    self.expireTime = personalFunctions.expireTime(sessionTimeout)

    # remove the instance from the userSessions list
  def removeSession(self):
    userSessions.remove(self)

# go through the userSessions list and find if the current session is valid
def isSession(token):
  if token == None or token == '':
    return False

  for i in userSessions:
    if i.token == token:
      # if it is valid, then refresh the session
      i.refreshSession()
      return True
  return False

# returns the instance of the session based on the input token
def getSession(token):
  if token == None or token == '':
    return None

  for i in userSessions:
    if i.token == token:
      # refresh the session because it was found to be valid
      i.refreshSession()
      return i

# function that removes inactive sessions. Found in the runPeriodically function in app.py
def removeInactiveSessions():
  # get the current time
  currentTime = personalFunctions.time()
  # iterate over the userSession list
  for i in userSessions:
    #print(f'the user {i.username} created their token at {i.timeCreated}, refreshed it at {i.refreshTime}

    # Remove sessions that have not been refreshed for at least 30 minutes
    if currentTime > i.expireTime:
      userSessions.remove(i)
```

Figure B2: Project Expenses

Software	Price
Github Codespaces	$0.00
Python3	$0.00
Proxmox	$0.00
Cloudflare	$0.00
Roundcube Mail	$0.00
Total	$0.00

Appendix C

Figure C1: Early home page

Figure C2: Early login page

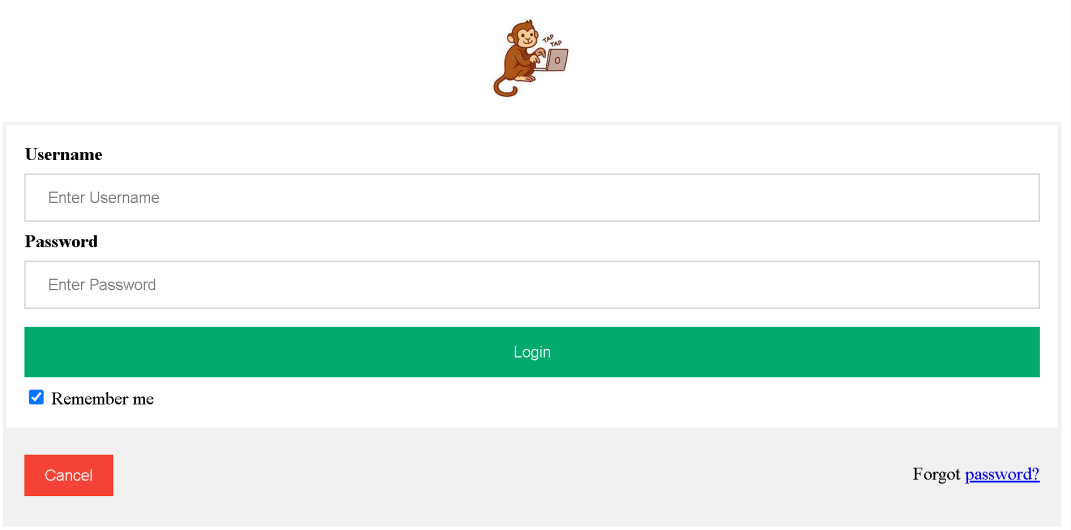


Figure C3: Early modules page

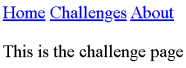


Figure C4: Unstyled modular modules page

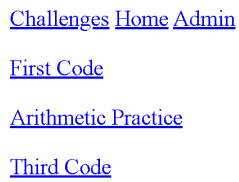


Figure C5: modular module page and user code and module completion saves

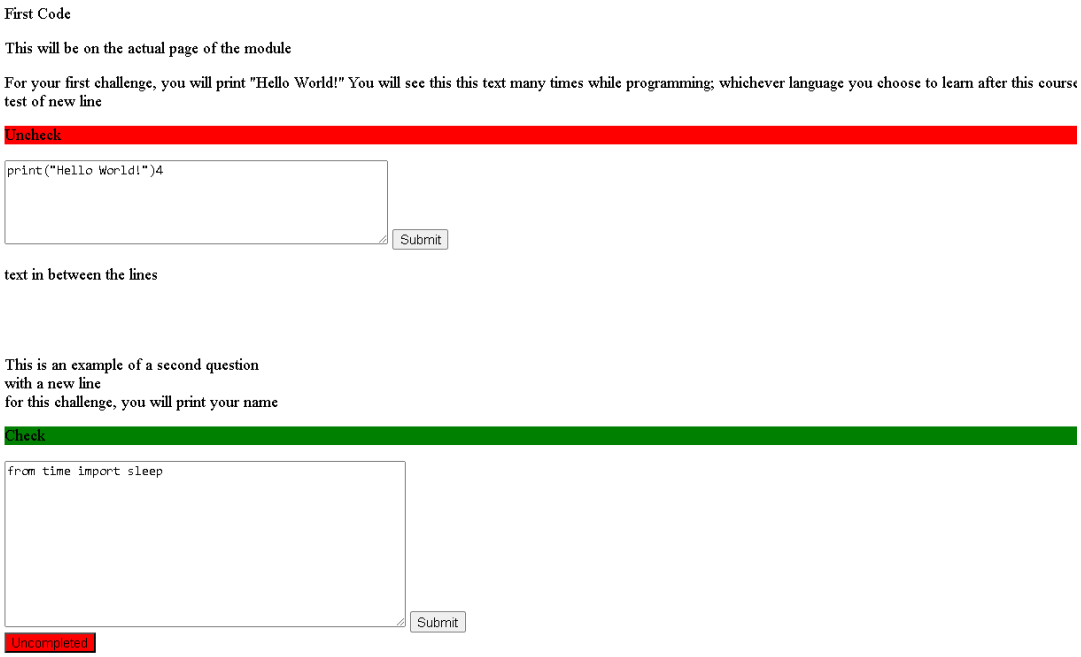


Figure C6: Early Admin page

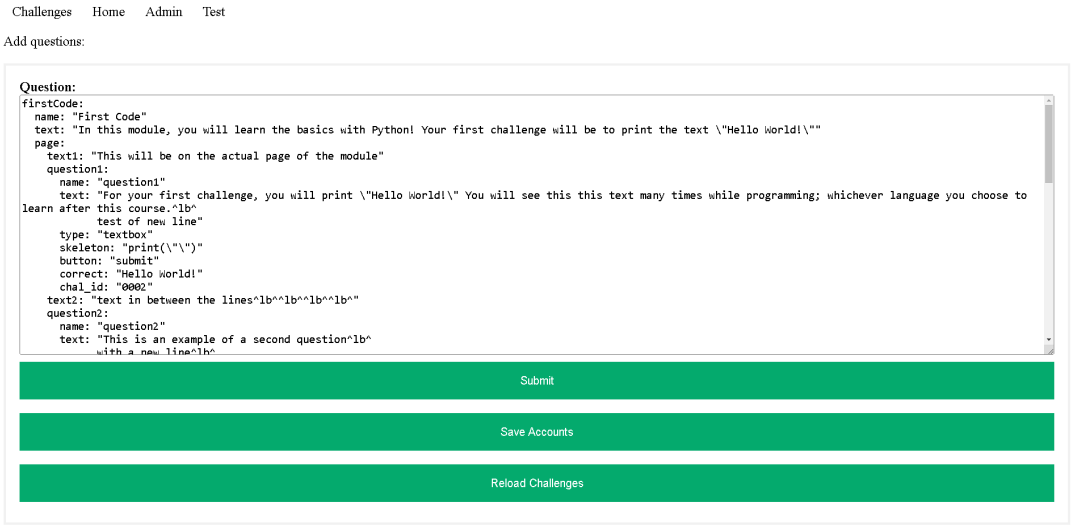


Figure C7: Current home page

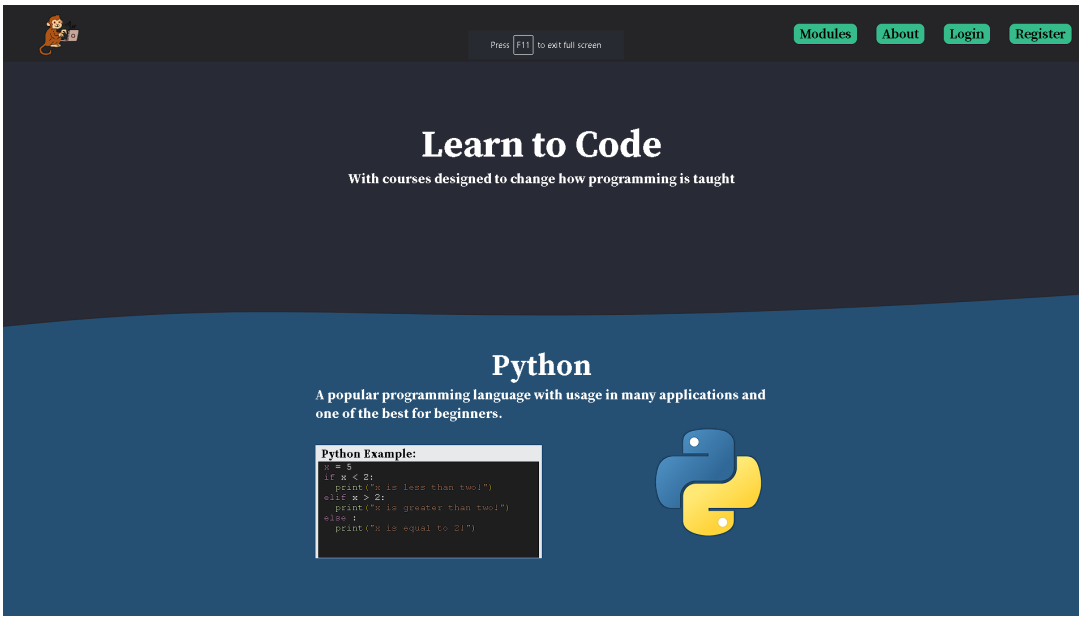


Figure C8: Current login page

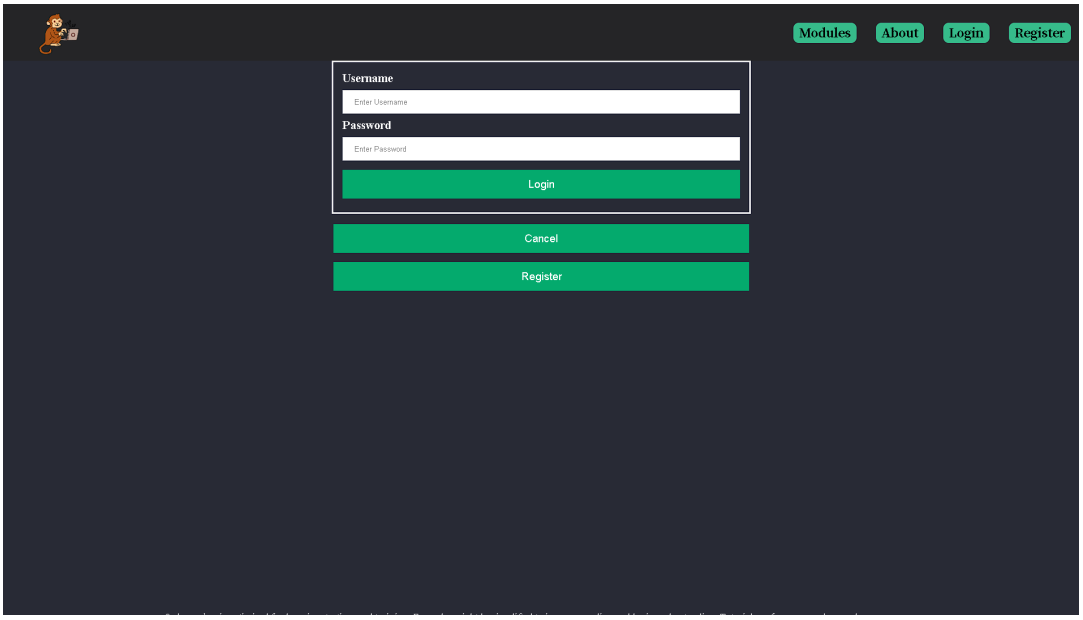


Figure C9: Survey Page

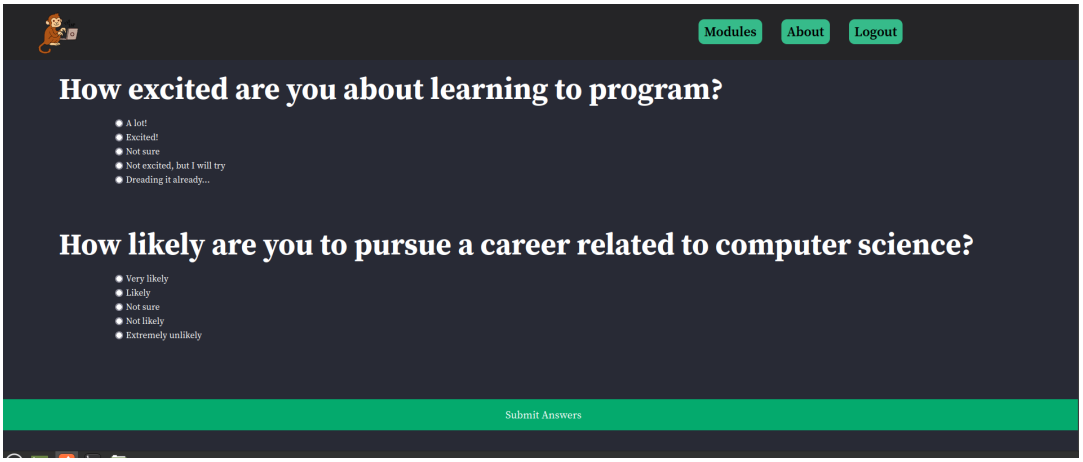


Figure C10: Current modules page

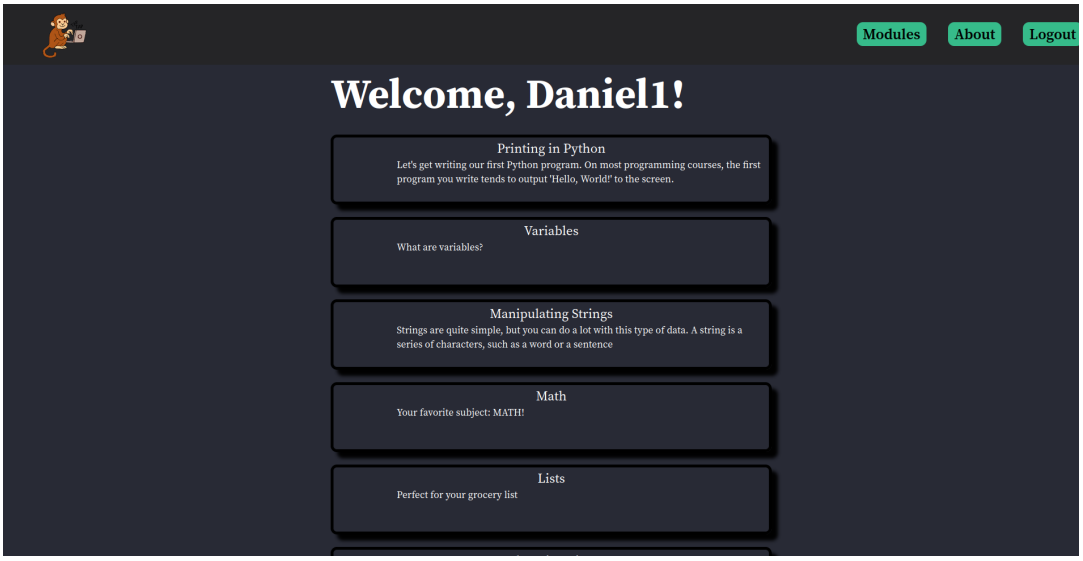
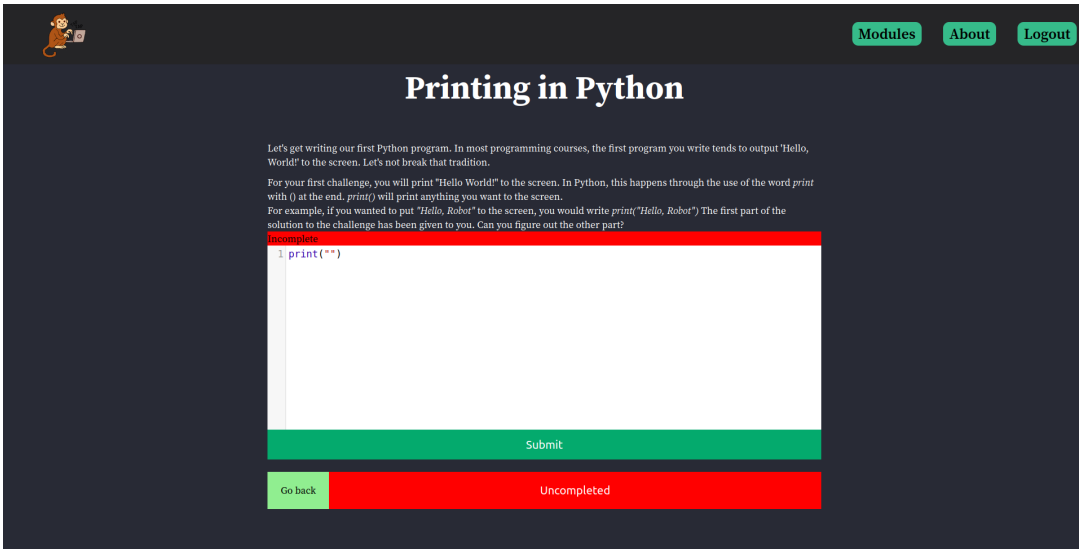


Figure C11: Current modules page



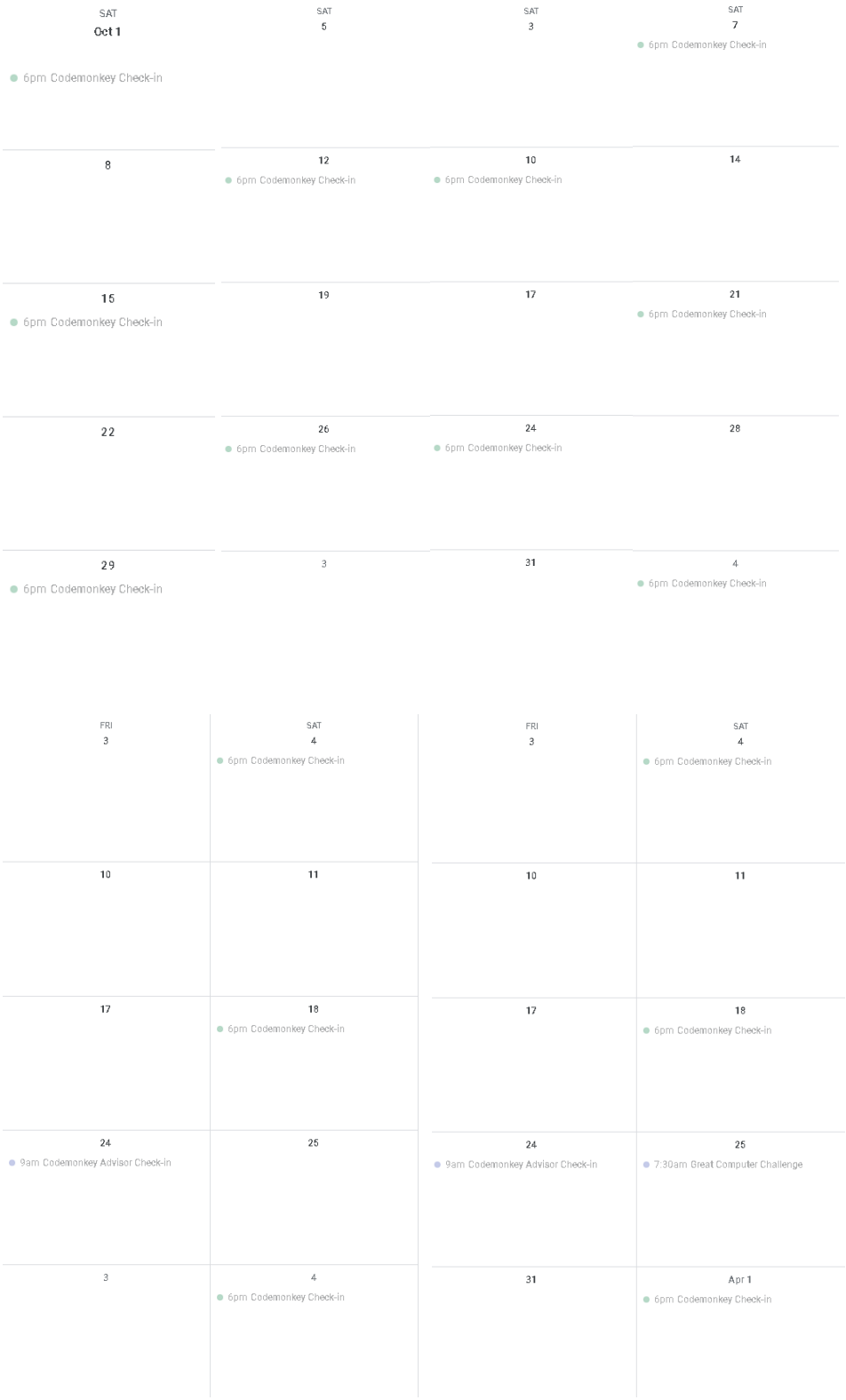
Appendix D

Figure D1: Daniel Tomov's certifications

Figure D2: <other team member>'s certifications

<other team member>'s certifications

Appendix E


SAT
Oct 1
6pm Codemonkey Check-in
SAT
5
SAT
3
SAT
7
6pm Codemonkey Check-in
8
12
6pm Codemonkey Check-in
10
6pm Codemonkey Check-in
14
15
6pm Codemonkey Check-in
19
17
21
6pm Codemonkey Check-in
22
26
6pm Codemonkey Check-in
24
6pm Codemonkey Check-in
28
29
6pm Codemonkey Check-in
3
31
4
6pm Codemonkey Check-in
FRI
3
SAT
4
6pm Codemonkey Check-in
FRI
3
SAT
4
6pm Codemonkey Check-in
10
11
10
11
17
18
6pm Codemonkey Check-in
17
18
6pm Codemonkey Check-in
24
9am Codemonkey Advisor Check-in
25
24
9am Codemonkey Advisor Check-in
25
7:30am Great Computer Challenge
3
4
6pm Codemonkey Check-in
31
Apr 1
6pm Codemonkey Check-in

Appendix F

Glossary

Back end -

Capture the Flag (CTF)

Front end -

Uniform Resource Locator (URL) - the link that is used to direct to a website

Old Donation School (ODS) -

Remote Code Execution (RCE) -

References

Cyberseek. (2023). Cybersecurity supply and demand heat map. Cyberseek. Retrieved December 16, 2022, from https://www.cyberseek.org/heatmap.html

Tech, G. (2022, December 9). The 5 easiest programming languages to learn. Georgia Tech Boot Camps. Retrieved January 6, 2023, from https://bootcamp.pe.gatech.edu/blog/easiest-programming-languages-to-learn