CIS272 Final Report

The purpose of my project is to educate people about cybersecurity and its vulnerabilities. I've created an app that allows users with no prior experience to grasp simple concepts related to vulnerabilities. I feel that cybersecurity is a field that many people don't know much about. Even at our school, while there are many computer science projects, there aren't many resources dedicated to cybersecurity, and I would like to address that. Additionally, I am passionate about this project because it aligns with the field I plan to pursue after completing my studies at BMCC.

I have developed an application that users can interact with. When you open the application , you are first presented with a brief description of cybersecurity. Then, you can select the type of vulnerabilities you want to focus on and learn about. While designing this app, I prioritized simplicity for the user, which is why the interface is straightforward and user-friendly.

To run this mobile app, you don't need much beyond an Android or iOS device. Since we are using App Inventor, it is easy to scan the QR code or even use an emulator to run the app. An updated, modern mobile device is preferred, as the user interface relies on up-to-date functionality. Additionally, a reliable internet connection is required.

This project was completed independently over several months. On November 15, I submitted my first proposal, which required coming up with the initial idea for the application. Initially, I wasn't sure what project to choose because there were so many options, but then I thought about something I am passionate about, and the idea for a cybersecurity application came to mind. After the proposal, we used Figma to design the user interface (UI) and visualize how the app would look to users. This was the first step in the application development process. The Figma design was due on November 18, and on that day, we presented our designs to the class. Each of us had five minutes to showcase our work.

During Week 11, on November 26, we created PowerPoint slides to document our progress. Each group had to present a test case table, their implementation progress, and any challenges faced during the project. On December 3, we submitted another progress update, which included an updated test case table. The final presentation took place on December 18 and included a demo of our application. Although I've listed specific dates above, I worked on the project daily, constantly adding new features or troubleshooting issues. Some days, I spent hours resolving a single problem, while on others, I added a feature and returned to it later for refinement. Eventually, I completed the application.

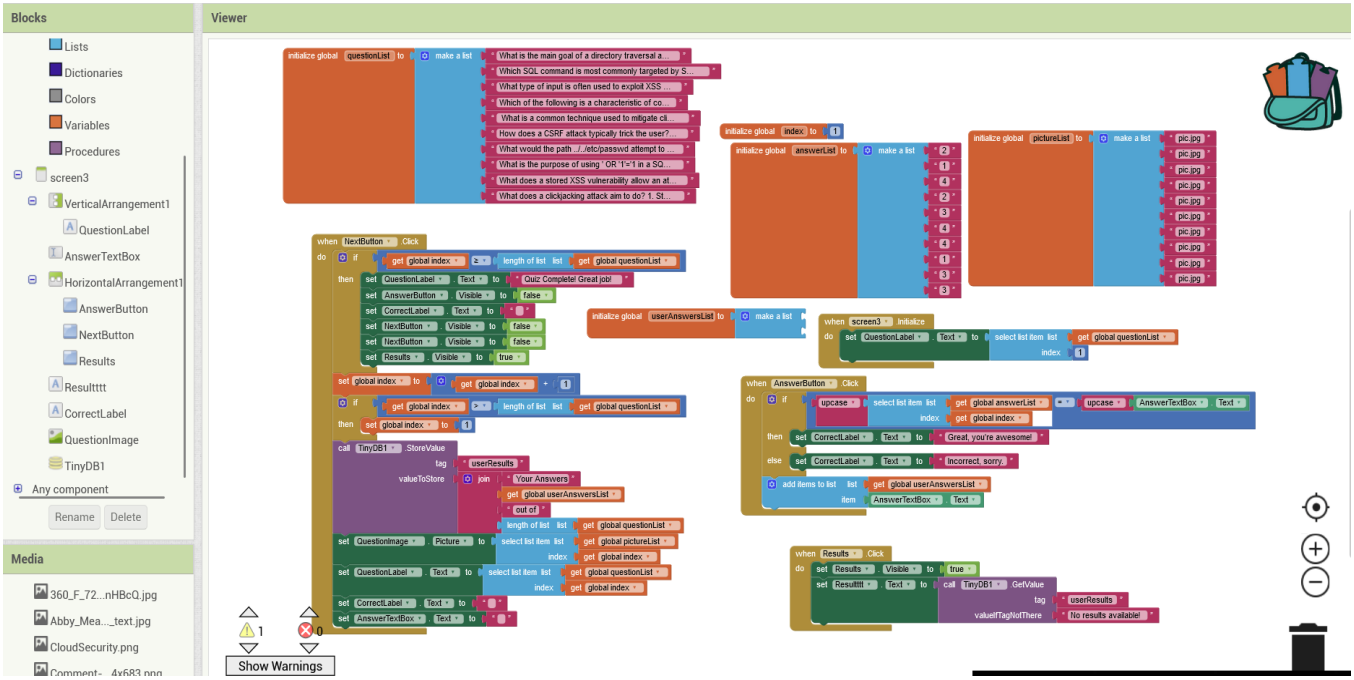
My application consists of 30 screens in total, each connected to one another. The main home screen is called "Screen 1." On this screen, the user is provided with a brief description of the application and its mission. This screen has three buttons at the bottom: "Menu," "Quiz," and "Career."

The Menu button takes the user to Screen 2, which displays a list of vulnerabilities the user can learn about. The vulnerabilities included are Directory Traversal, Cross-Site Scripting, Clickjacking, SQL Injection, Command Execution, and Cross-Site Request Forgery (CSRF). I chose these six vulnerabilities because they are some of the most critical today and are very important in the field of cybersecurity. After selecting a vulnerability, the user is taken to a webpage with an interactive lab where they must read and type in commands to simulate an actual attack.

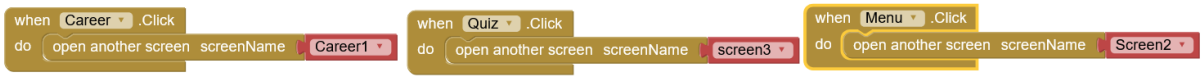
The Quiz button is designed to test the user's knowledge of cybersecurity. The quiz consists of 10 questions, and at the end, the user is provided with their results. The quiz also uses TinyDb in order to store user input, to achieve this I use a **questionList** to store all my quiz questions and an **answerList** to keep track of the correct answers. When the user enters their response in the text box and clicks the "Answer" button, the app checks if their answer matches the correct one. If it does, it shows a message like "Great, you're awesome!" Otherwise, it says "Incorrect, sorry." At the same time, I save the user's response into a **userAnswersList** so I can keep track of what they answered. When the user clicks the "Next" button, the app moves to the next question, and when all the questions are finished, I store the user's answers in **TinyDB** under the tag "**userResults**". Later, when the user clicks the "Results" button, the app retrieves the saved answers from **TinyDB** and displays them in a results label. This way, my app gives feedback, saves the user's progress, and lets them review their answers easily

The Career button takes the user to a screen showcasing various cybersecurity careers. On this screen, users can explore different job roles and what they entail. The buttons available include "Cyber Crime," "Network Security," "Security Analyst," "Penetration Tester," "Cloud Security," and "Malware Analyst." After selecting a career, the user is taken to another screen specific to that career, where they can learn more about it and watch a related video. Once the video is over, the user can press the EXIT button to return to the Career page and explore other options.

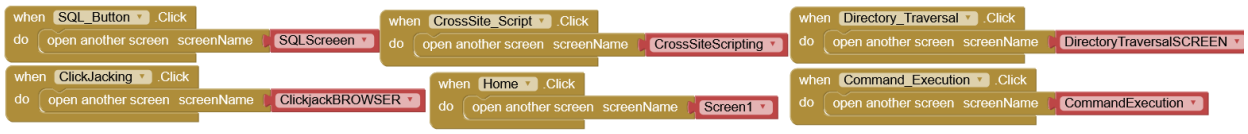
Here is a snapshot of my code for the Quiz, showcasing the logic behind my quiz functionality, including how questions are displayed, user responses are tracked, and results are stored and retrieved.



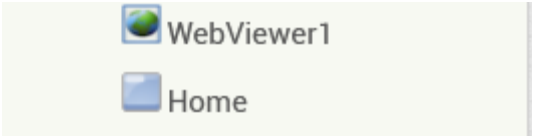
As for workflow and algorithms. The Main Screens are “Screen 1”, "Screen 2” “Career” and “Quiz” These screens will allow you to navigate to other screens. Below is an image of the blocks behind the “Screen 1”.



To achieve this goal, we are using the "When .click" function, which opens the next screen. This method is applied throughout the entire application, as there are 30 screens in total, each with a different function. After navigating to Screen 2, you are presented with options to learn about different attacks. Below is an image of the navigation for Screen 2. As you can see, each button opens another screen. I refer to these as "sub-screens" because, while they are screens, they are not the main screens of this application.



When the user selects a vulnerability to learn about or navigates to the career page to explore a specific type of screen, they are presented with a browser that plays a video. This feature is implemented using the WebViewer1 component in App Inventor (shown below).



Overall workflow of this app is shown below

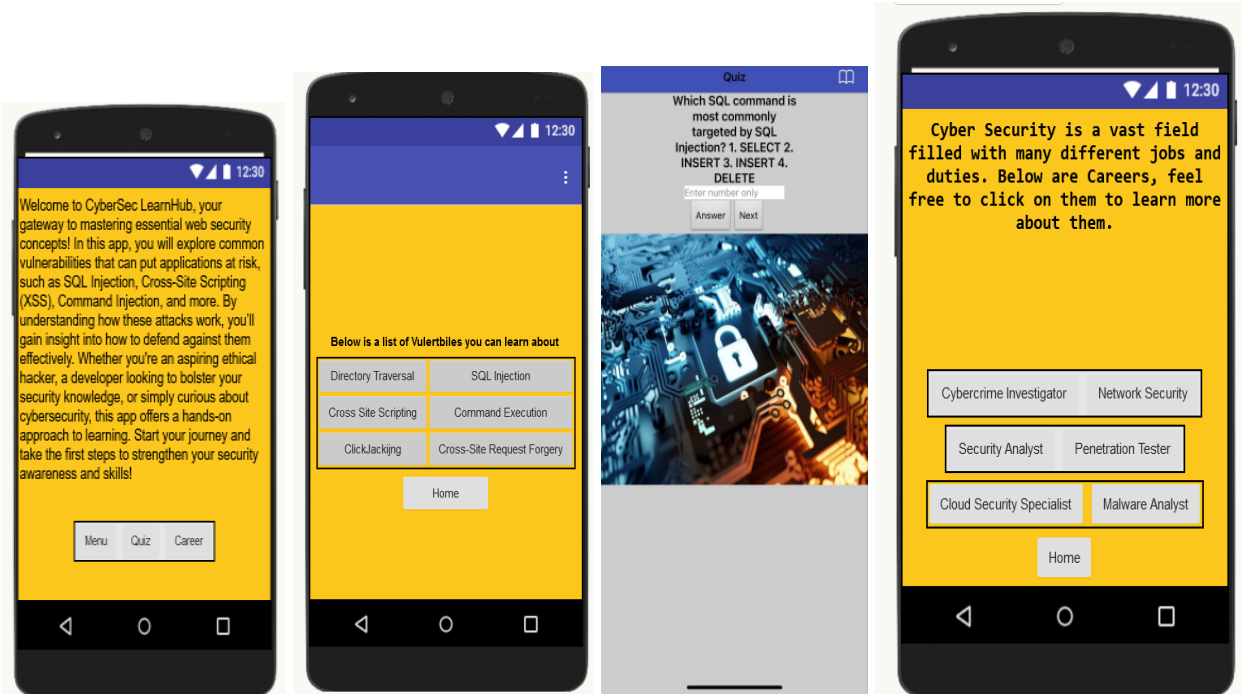
Main Screens (Screen 1, Screen 2, Career)

⬇️

Sub Screens

⬇️

The core pages of the application, as mentioned above, are "Screen 1," "Screen 2," "Quiz," and "Career." These pages provide the majority of the application's functionality. After making a selection on one of the main screens, the user is taken to the sub-screens, each designed to cover a different aspect of cybersecurity.



Below is the prototype of the Youtube link & Figma showing a demo of the Application.

[Youtube Link to protype]

[Figma UI]

Creating this project we had to also make tables with Components used and a test table, Below I have pasted it.

Component	Type	Description	Key Properties
QuestionLabel	Label	Displays the quiz questions to the user	Text alignment, font size
AnswerTextBox	TextBox	Allows users to input their answers	Text alignment, hint text
NextButton	Button	Navigates to the next question	Text, visible/invisible settings
Results	Button	Shows the quiz results to the user	Text, visible/invisible settings
TinyDB1	TinyDB	Stores user answers, counts, and other persistent data	Non-visible

Test table for quiz

Case	Description	Expected Outcome	Result
Open Quiz	User taken to quiz Screen	Quiz interface opens with first question	Displays correctly first question
Submit Answer	User inputs answer and clicks next	Quiz stores users input	Answer stored in TinyDB
Next Question	User clicks 'Next' after answering	Next question is displayed for user	Next question is displayed
Complete Quiz	User finishes last question and submits	Results calculated and displayed	The answers user got wrong is displayed

Component for rest of application

Component	Type	Description	Key Properties
Label1	Label	Displays the main welcome message	Text, Font, Color
Menu	Button	Navigates to the lesson selection	Text
Quiz	Button	Initiates quiz activity	Text
Career	Button	Opens career information related to cybersecurity	Text
WebViewer1	Web Viewer	Displays content or links for deeper information	HomeUrl, EnableJS

Test table for rest of app

Case	Description	Expected Outcome	Result
Navigation to Lessons	Tap on 'Menu'	Show topic selection screen	Pass: Screen transitions correctly
Start Quiz	Tap on 'Quiz'	Quiz starts with the first question	Pass: Quiz loads properly
Career Exploration	Tap on 'Career'	Display different cybersecurity careers	Pass: Careers displayed correctly

All tests were successful.

For this project, we were required to upload it to the MIT App Inventor Gallery. Below is the link to the application, which is now publicly available for users to modify and use:
[MIT App Inventor Gallery Link]

Additionally, here is the link to the Google Site for the application:
[Google Site - Cyber Sec Learn Hub]

Making this application wasn't simple. It took a lot of time and thought to develop. One of the strengths I love about my application is that it is easy for users to understand how it functions. Everything is straightforward: you load the application, and the instructions are right there. Another strength is that it uses the browser function, which educates the user on various topics.

However, one of the weaknesses of this application is the user interface. It's not fancy at all—just simple. While making the quiz for this application, I felt very limited due to App Inventor, which restricted the capabilities I could implement. Another weakness or issue with this application is that I use an iOS device, and to make the application work for Android, you have to constantly adjust the design to meet the needs of both platforms. I think there should be an option to automatically optimize the app for its users. I had to use the pixel feature to try and center everything on the screen, hoping it wouldn't display incorrectly on a different device.

I also tested the app on my iPad, which has a larger display, and I encountered no problems. Another issue was the placement of buttons for the user. On my mobile device, the button is positioned at the very bottom of the screen, which can be difficult for users to reach. Also putting images on the screen caused a lot of bugs where even my application would crash. This is why my background color for the application is yellow instead of putting a nice background image because that was my original goal to make it look nice. I also wanted to use a list picker so the user can pick an answer from the list then store it in the database but I had a problem with that also so I just kept it simple with buttons.

In the future, I would like to recreate this application using a different platform that isn't App Inventor because it presents many limitations. This application has the potential to become a real app if I follow the same idea!

This application wasn't simple to make at all. One of the reasons I chose to do it alone is that cybersecurity is something I am passionate about, and my schedule is very random along with all the classes I am taking this semester. I didn't want to make that an issue for another student. Although this was a project that could require more people, doing it alone has taught me a lot about App Inventor, and I feel like I really learned more by doing this project alone than at the start of the semester, when the teacher would give us homework and quizzes. I feel like during those times I didn't learn as much since it was very fast-paced, but for a project and something I am passionate about, I could go at my own pace, research, and learn new things along the way to make my project better over time. I hope this reflection can help other students at BMCC since this is a new elective class. As for writing, I really love how this will satisfy my writing-intensive requirement. I do enjoy writing, and any opportunity to write will make me a better writer.

The aspect of research has really helped me and will also help me in the future when making an application or in my career. I feel like it wasn't just about making an application but also about the experience we gained along the way. I was also able to help and ask for help from other classmates who were doing their own projects, and they also said they enjoyed the project more than the first part of the semester. Having to make slides and present in front of the class also helped me a lot because I know in the future I will have to do the same thing in my line of work, and presenting in public is a skill I need to work on over time. The second part of the research involved making the Figma, PowerPoints, application websites, test tables, and writing this paper itself because it really helped me think about how to compile a project in a professional setting, including sources, explaining how my project works, and much more. Even though I did this project alone, there was room to work with other classmates to finish this project and learning how to work together is a skill that will help me later on.

In the future, I would like to make this application again to educate users on cybersecurity as a whole. This software is the foundation for me and its ideas because I know what I need to work on now. I do feel like my application is unique in its own way and can be improved in the future. Since cybersecurity is the field I want to go into, putting this application on my resume or counting it as a project will help me stand out as a student. There was a lot of research involved and reading, some days I ran into problems, other days I fixed the problem and this helped me grow as a student. This application was a journey of trial and error, where every step taught me something new. The process of building, testing, and improving the app allowed me to apply my knowledge in a practical way while learning from mistakes. Each challenge I overcame showed me how much I am capable of and motivated me to keep improving my skills

Appendix

A. Weekly Progress Updates and Work Plans

- **Weekly Presentations:**
  - Presentation 1
  - Presentation 2

B. Course Materials Utilized in the Project

- **Figma Design Tool:**
  - Figma Project Link

C. Additional App Inventor Tutorial Links Used

1. **General Tutorials:**
   - App Inventor Quiz tutorial
   -
2. **Specific Features and Advanced Techniques:**
   - App Inventor Penetration testing video
   - App Inventor Security Analyst Video
   - App Inventor Network Security Video
   - App Inventor Malware Analyst Video
   - App Inventor Cloud Security Video
   - App Inventor Cyber Crime Video