Cyber Warfare: Research and Development

Project overview

The aim of this project is to develop a Cyber City that is controlled by Arduinos and Raspberry pies. The Cyber City will be simulating different government departments and other functions and responsibilities. For example; some of the government departments being simulated are Parliament House and the missile launcher. Some examples of other city functions would include; train track controls and traffic lights. The final goal of the project is to run a Cyber War Game using the Cyber City where users will have to hack into the project's devices and "takedown" the city. The hacking may include circumventing usernames and passwords or attempting SQL injections.

The focus of the project

At the beginning of the project, our group had to decide the theme of the project and what functions the project would have. After discussing the themes of previous projects, we decided that the theme of our project would be Attack of the Kiwis. In Attack of the Kiwis, New Zealand is attempting to hack into Canberra's government departments because Australians made fun of their accents. Once we had decided the theme of the project we broke off into small groups which each group focusing on a different component of the project. The components included; the train track controls, the traffic lights, Telstra Tower, the missile launcher, Parliament House and Emergency Services.

Now that each group had a focus, we began to create the logic planning for the components and design the wiring diagrams. The logic planning was done on google docs and used simple code so that anyone could read the logic plan and understand how the device was supposed to work. Once the logic had been created we used a program called Cirkut to design the wiring diagram for the component. The reason that we used Cirkut instead of Fritzing was that Cirkut allowed us to create our own components which gave us the freedom to use components that Fritzing didn't have.

Following the wiring diagrams that we made, we began to test the components. Testing the components has two purposes. The first is to ensure that the hardware was working correctly and that we wouldn't have to replace any components. The second is to test the logic that we had previously created to see if changes had to be made to fit the project outline.

Finally, after the components were tested and the basic code was designed, we began to draft plans for the website and an evaluation checklist. The website will be where the controls and information for the components will be found and could possibly be used in the cyber warfare game.

When designing the website we had to consider how the website could be best designed for user accessibility. Would there be a place to enter a username and a password, would there be a menu bar that allowed users to navigate the website and how will the users interact with the components. The evaluation checklist shows our understanding of the project as a whole and is a checklist of the key points that our project will be evaluated against at the end of development. The checklist will check things like; whether or not the project runs, can the user authenticate and does the website update correctly according to the components.

Cybersecurity

There are many different types of cybersecurity and threats to cybersecurity. These include; SQL injections, Brute Force Attacks, password hashing, IoT security issues, network and packet sniffing and WPA2 vulnerabilities. When talking specifically about our project, there are two main cyber security threats that we could incorporate into our projects so that the users could hack into our devices.

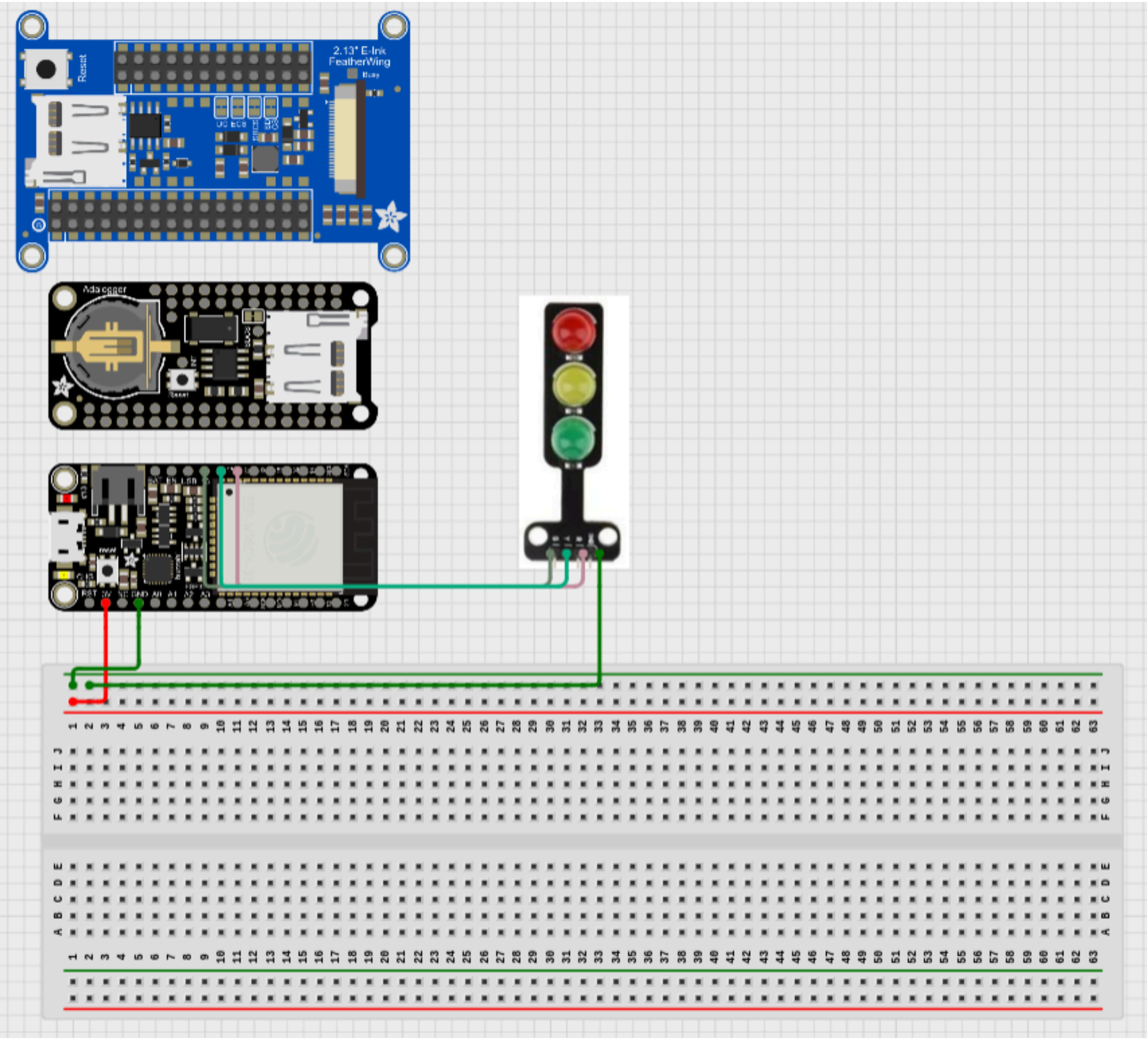
The first threat we could implement into our project is SQL injections. SQL injections are a code injection technique that can be used to change the functionality of an input. SQL injections are usually used on inputs such as usernames or passwords but can be used on anything that requires the user to input something into the website. When the statement is inputted, it will run on the database and can do any number of things such as; leak the database’s information or wipe the database. To implement SQL injections into our project we change the code of our website so that when certain inputs are used, they change the functionality of one of the devices. For example, an SQL injection could be used to change the traffic light settings or fire missiles from the missile launcher.

The second threat we could implement into our project is brute force attacks. Brute force attacks use trial and error to guess login info, encryption keys, or find hidden web pages. Hackers and even some brute force software will work through all possible combinations hoping that they guess correctly. To implement brute force attacks into our project, we could simplify the passwords related to some of the devices such as the voting system or emergency services so that they could be easily guessed by users. For example, passwords such as 1234. Additionally, we could even give the users hints for the password that have been encoded using cryptography. Such as Caesar cyphers to other simple-shifting-cyphers.

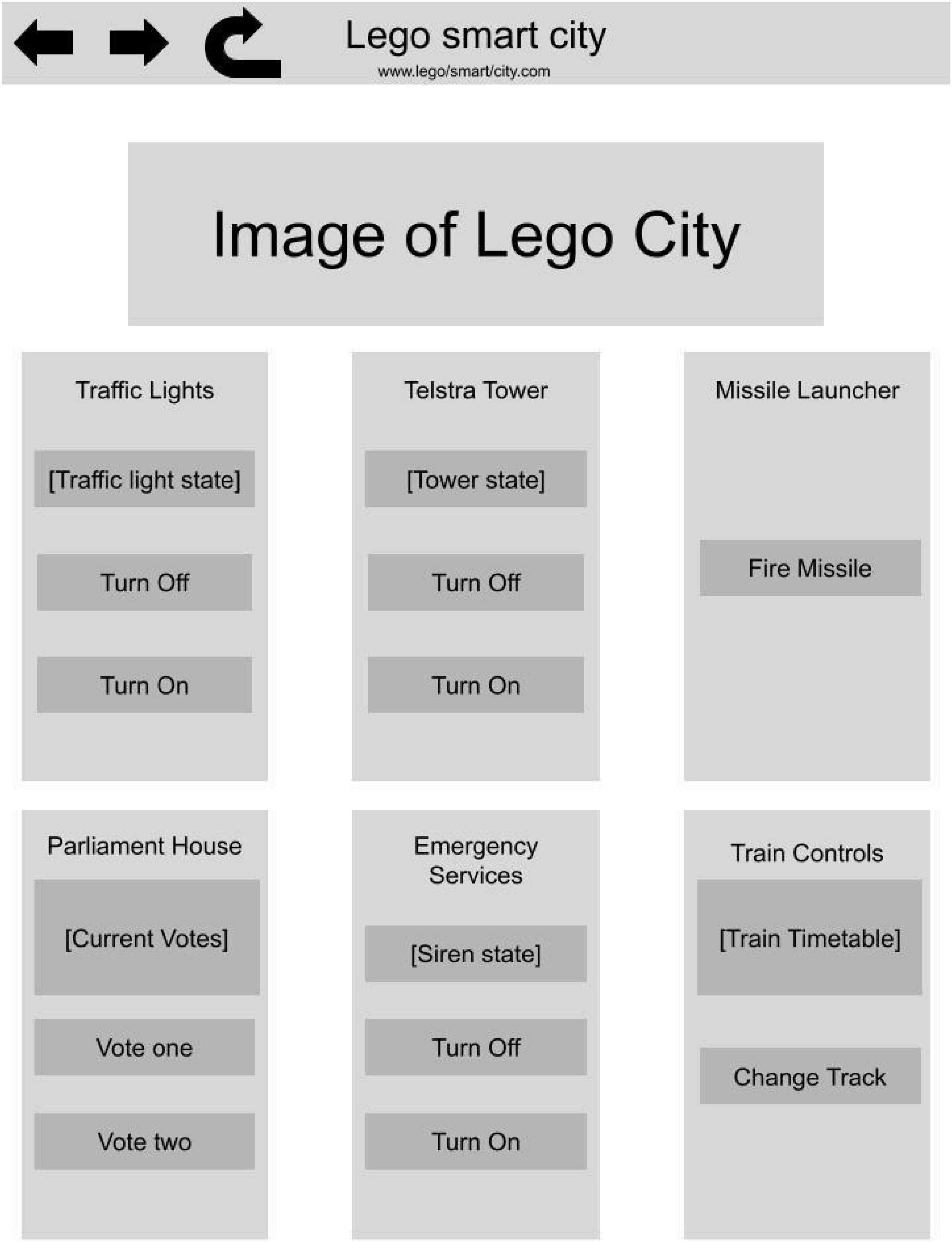
In conclusion, we could implement SQL injections into our project using visual studio code and brute force attacks with encoded clues. This would allow the users to hack into certain devices to earn points in the cyber warfare game.

Planning diagrams

There were two diagrams/sketches made in this stage of the project. The first diagram was the wiring diagram. The wiring diagram was made on Cirkut as opposed to Fritzing because it allowed us to create our own custom assets. The second diagram/sketch was the website sketch. The website sketch was made using google drawings and allows us to begin to consider how the website could be best designed for user accessibility.

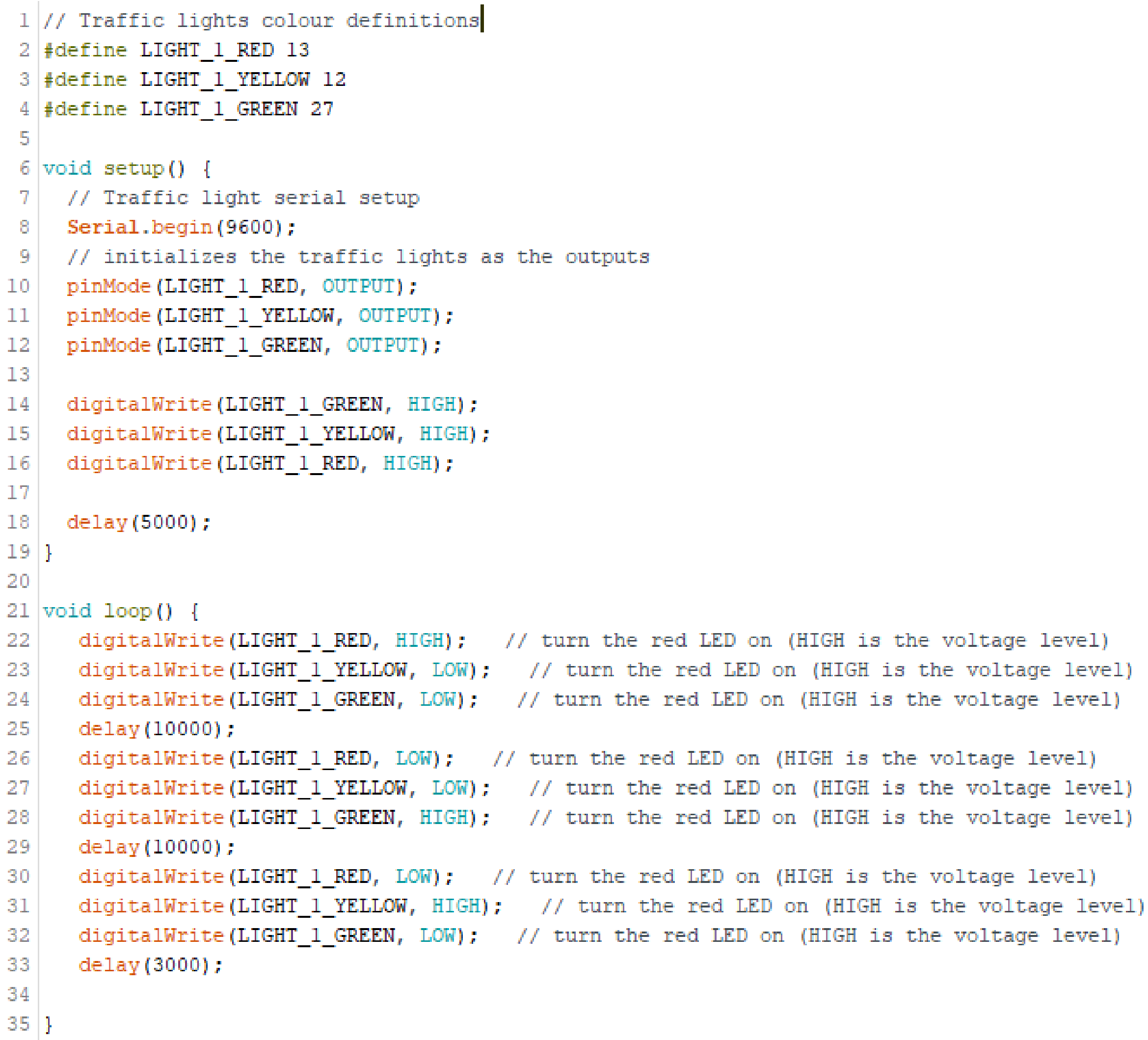


In the wiring diagram, the Huzzah 32's GND and voltage pins are connected to the breadboard. The other component of the wiring diagram is the traffic light module. The traffic light module is also connected to the breadboard through GND but is also connected to the equivalents of the analogue pins. These are pins; 13, 12 and 27. These pins were used because I thought they would be the easiest pins to use without interfering with any other components that might be added at future stages of the project, such as other traffic light modules.



In the website sketch, I designed a basic design for the website using last year's website as a basis. The sketch includes all of the components, including the train track controls, the traffic lights, Telstra Tower, the missile launcher, Parliament House and Emergency Services. On the traffic lights controls, you can see the current state of the traffic lights and also have the ability to turn them on or off. This is the same for Telstra Tower and Emergency Services. The missile launcher only has one control which is to fire the missile. Parliament House has a state and two different functions. The state reads how many votes have been cast for each side and the buttons allow you to vote for either side. Finally, the train track controls have a state and a button. The state displays the train timetable and the button switches the train track.

Functionality plan



```
// Traffic lights colour definitions
#define LIGHT_1_RED 13
#define LIGHT_1_YELLOW 12
#define LIGHT_1_GREEN 27

void setup() {
  // Traffic light serial setup
  Serial.begin(9600);
  // initializes the traffic lights as the outputs
  pinMode(LIGHT_1_RED, OUTPUT);
  pinMode(LIGHT_1_YELLOW, OUTPUT);
  pinMode(LIGHT_1_GREEN, OUTPUT);

  digitalWrite(LIGHT_1_GREEN, HIGH);
  digitalWrite(LIGHT_1_YELLOW, HIGH);
  digitalWrite(LIGHT_1_RED, HIGH);

  delay(5000);
}

void loop() {
   digitalWrite(LIGHT_1_RED, HIGH);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_YELLOW, LOW);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_GREEN, LOW);   // turn the red LED on (HIGH is the voltage level)
   delay(10000);
   digitalWrite(LIGHT_1_RED, LOW);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_YELLOW, LOW);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_GREEN, HIGH);   // turn the red LED on (HIGH is the voltage level)
   delay(10000);
   digitalWrite(LIGHT_1_RED, LOW);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_YELLOW, HIGH);   // turn the red LED on (HIGH is the voltage level)
   digitalWrite(LIGHT_1_GREEN, LOW);   // turn the red LED on (HIGH is the voltage level)
   delay(3000);

}
```

Currently, the traffic light cycle works by; setting the red LED to HIGH and the others to LOW for ten seconds. Then set the green LED to HIGH and the others to LOW for ten seconds. And Finally, set the Yellow LED to HIGH and the others to LOW for three seconds. This code then repeats until the module is turned off.

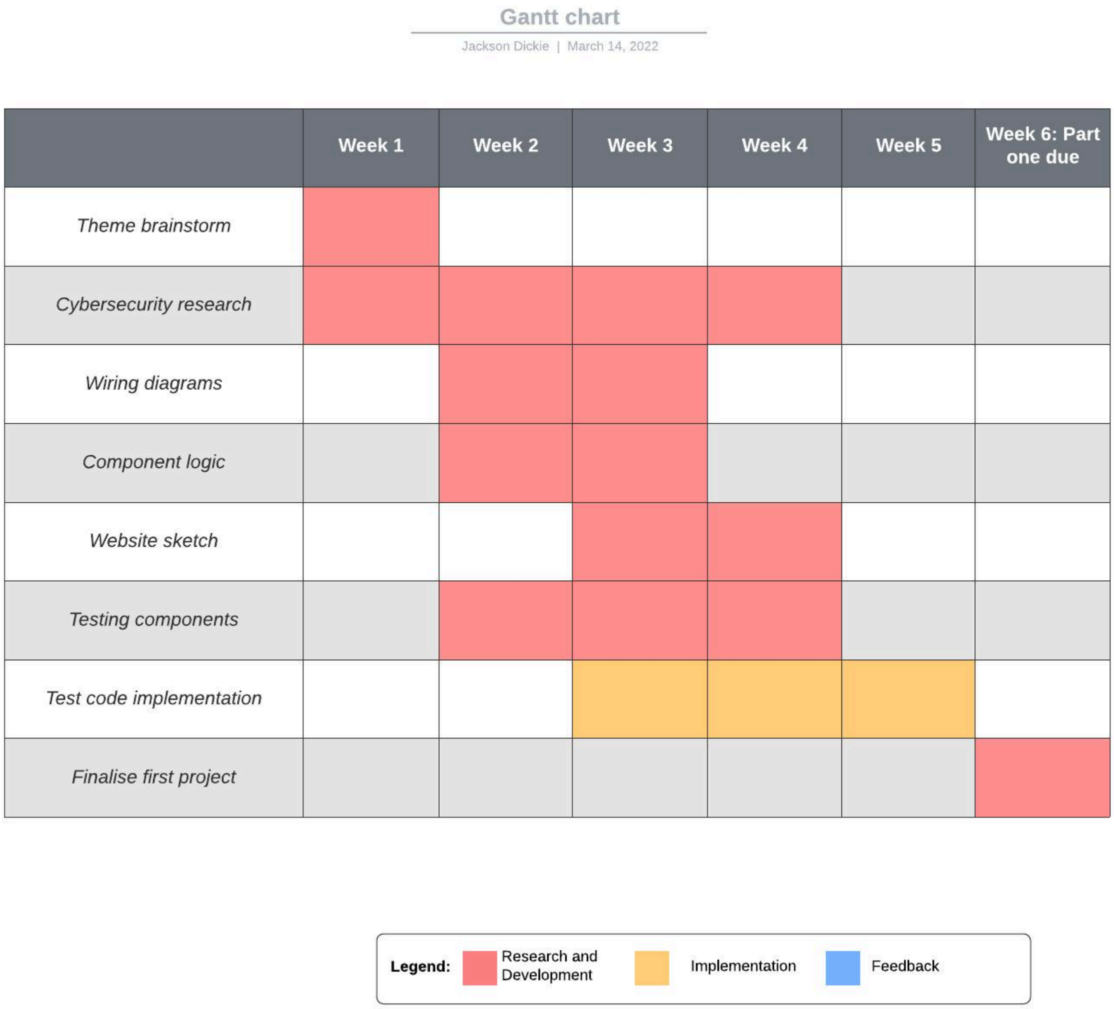
When the traffic lights are fully implemented into the project, the plan is to have another traffic light module that is running opposite to the first traffic light module. This means when the first traffic light is green, the other traffic light is Red. Additionally, in the final product, the user will have the ability to hack the traffic lights and freely change what setting the traffic lights are on or turn them off completely. The traffic lights might then also give the user something they will need to hack into another component.

Gameplay descriptions and analysis

At some point during the cyber warfare game, the user will have to hack into the traffic lights. At this point of the project, there will be two alternating traffic light devices. To hack into the traffic lights the user will have to use one of the hacking methods. The hacking method could be an SQL injection or a password that the user must input that would have gotten for hacking into a previous device.

When the user correctly hacks into the traffic lights the traffic lights will rapidly alternate between different light settings. The user will also be awarded a point and given a clue or a way to hack into the next device. This could be a specific input for an SQL injection or a key for a cypher needed for a password.

Project plan



Gantt chart

Jackson Dickie | March 14, 2022

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6: Part one due
Theme brainstorm	🟥					
Cybersecurity research	🟥	🟥	🟥	🟥		
Wiring diagrams		🟥	🟥			
Component logic		🟥	🟥			
Website sketch			🟥	🟥		
Testing components		🟥	🟥	🟥		
Test code implementation			🟨	🟨	🟨	
Finalise first project						🟥

Legend: 🟥 Research and Development 🟨 Implementation 🟦 Feedback

Evaluation checklist

Goal	**Description**

Theme of project	The theme of the project has been fully discussed and implemented as planned into the project.
Website connectivity	The website is properly connected to the devices so that the users can hack into the different devices and change how the devices interact with one another.
Device functionality	All of the devices are fully functional in terms of the original intent for their functionality within the project. For example, the traffic lights can cycle between different states and the missile launcher can fire its missiles.
Cybersecurity	The cybersecurity threats have been fully implemented into the project so that they can be used to hack into different devices.
Gamification	The cyber security game has been fully developed so that users can hack into different devices, earn points and earn clues/keys that let them hack into the other devices.

Bibliography

Cyber War Games – Operation: Tsunami 2019 – 2-6 September 2019 – Canberra, ACT

Cyber War Games – Operation: Tsunami 2019 – 2-6 September 2019 – Canberra, ACT. (2022). Retrieved 14 March 2022, from https://cyberwargames.humanservices.gov.au/

Notion – The all-in-one workspace for your notes, tasks, wikis, and databases.

Notion – The all-in-one workspace for your notes, tasks, wikis, and databases. (2022). Retrieved 14 March 2022, from https://ryancather.notion.site/Cryptography-d8d404a4df21425d8608fb7f6b8f7b84

SQL Injection

SQL Injection. (2022). Retrieved 14 March 2022, from https://www.w3schools.com/sql/sql_injection.asp

Brute Force Attack: Definition and Examples

Brute Force Attack: Definition and Examples. (2021). Retrieved 14 March 2022, from https://www.kaspersky.com/resource-center/definitions/brute-force-attack

Hashing algorithms and security - Computerphile

(2022). Retrieved 14 March 2022, from https://www.youtube.com/watch?v=b4b8ktEV4Bg

What is IoT Security?

What is IoT Security?. (2022). Retrieved 14 March 2022, from https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security

What Is a Network Sniffer?

What Is a Network Sniffer?. (2022). Retrieved 14 March 2022, from https://www.lifewire.com/definition-of-sniffer-817996#:~:text=Network%20sniffing%20is%20the%20use,the%20appropriate%20software%20or%20firmware.

Practical Cryptography

Practical Cryptography. (2022). Retrieved 14 March 2022, from http://practicalcryptography.com/ciphers/simple-substitution-cipher/