Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

25% FYP Interim Report

Semester

2021 Autumn

PROJECT TITLE: Implementation of Hardware Token for User Authentication

London Met

Internal Supervisor:

Assignment Due Date:

Assignment Submission Date: 15th

Word Count: 3892

Abstract

This report, which is an interim report, serves as an illustration of the mid-term of the development of the customized project "Hardware token for user authentications," which is described in more detail below. Viruses and malware are among the most common risks that internet users must deal with. Spyware is one of the most common threats that internet users are exposed to. Cybercriminals may steal your personal information and follow your movements without your awareness if they use spyware on your computer. Alternatively, they may choose to trade or sell their knowledge to other cyber-threat operators in order to make money. As a result, our project intends to address such issues by developing a security hardware token that protects people's personal data and information from these attacks.

Table of Contents

Table of Figures

Table of Tables

1. Introduction:

1.1. Topic Introduction

As the present world is entirely dependent on the internet, IT professionals are constantly looking for new ways to propel accelerated innovations. However, technology comes at a cost, as everything has flaws. In the light of highly publicized information ruptures, IT proficient are continually looking for security endeavors (precautions). There have been numerous cases of cyber security disasters at various enterprises caused by spyware. Basically, it is software that installs itself on your computer and begins secretly monitoring your online activity without your knowledge or permission. Any software can be classified as spyware if it is downloaded without the user's authorization (Kaspersky, 2021).

One of the most prevalent risks to internet users is spyware. It basically operates by transmitting personal information such as your name, address, browsing history, preferences, or downloads via your internet connections (Gillis, 2021). Credit card numbers, bank account numbers, and passwords are among the most common types of personal data that spyware collects. As a result, people's personal information and data are potentially vulnerable to cyber-attacks, and none of their data is locally secure.

Implementing strong security protocols, such as hardware tokens, is the most effective way to avoid spyware. A hardware token is a physical device used for strong authentication into a system. It is generally used in secure environments. Its objective is to add an extra layer of protection by requiring two-factor authentication (HIDEEZ, 2021). A number of businesses, including gambling platforms and banks, are presently using this method of securing their data and systems.

1.2. Problem Scenario

Since its modest origins, spyware has seen a fast development, much like other dangerous applications. Viruses and malware are among the most prevalent dangers to internet users. Spyware is a sort of virus that aims to remain undetected while surreptitiously recording data and tracking your internet activity. It can monitor and duplicate anything you do. Some software might secretly activate cameras and microphones to spy on you.

Using spyware, cybercriminals may steal your personal information and track your movements without your knowledge. Alternatively, they might trade or sell their information to other cyber threats operators for profit. Cybercriminals are well aware that your IT systems are a goldmine of data (Savvy, 2021). The following are examples of crimes that may be committed as a result of data theft:

- Theft of one's identity
- Bank fraud and other forms of financial fraud
- Theft of passwords and login information
- Taking credit card information
- Data and keystrokes are being recorded.
- Planting various sorts of malware on the victim's device

Malwarebytes, “which categorizes spyware as one of two types of “stalkerware-type detections” in its 2021 report, observed that consumer spyware detections rose 24% to 2.43 million in 2020 (up from 1.96 million in 2019). On the other hand, the business spyware detections increased 51% from 291,525 in 2019 to 440,368 detections in 2020” (Savvy, 2021).

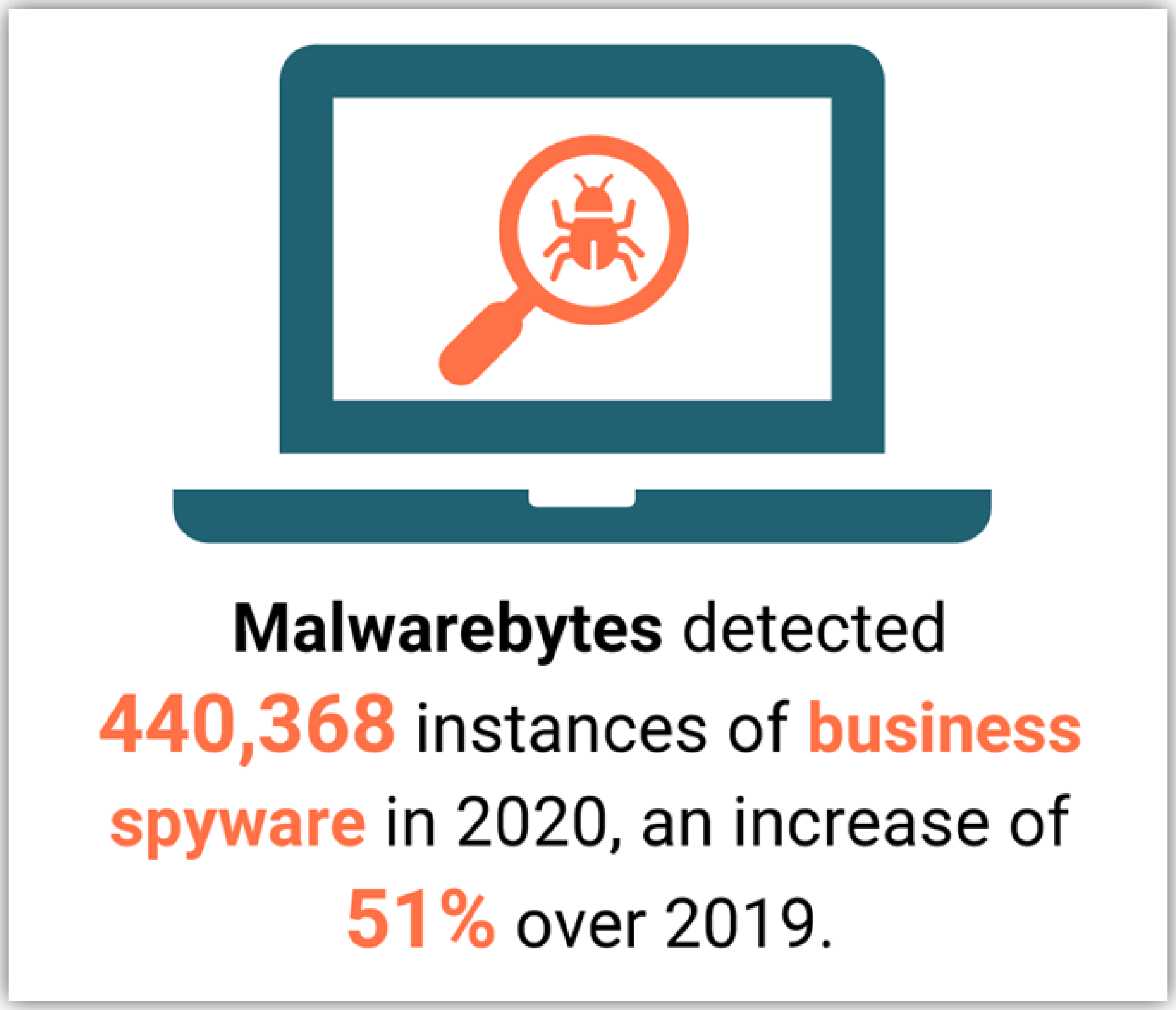

Malwarebytes detected
440,368 instances of business
spyware in 2020, an increase of
51% over 2019.

Figure 1: Malwarebytes 2021 State of Malware Report

Between January 1 and June 30, 2020, Malwarebytes detected an increase of 780 per cent in “Monitor” app detections (the other form of stalkerware Malwarebytes detects) (Savvy, 2021).

“Worse, the consequences of cyberattacks continue to grow, with digital incidents now costing businesses of all sizes $200,000 on average, according to insurance carrier Hiscox. Sixty per cent go out of business within six months of being victimized.”

1.3. The Project as a Solution

In general, a security hardware token is created to secure people's personal data and information from these threats. Its objective is to add an extra layer of protection by requiring two-factor authentication. An OTP code will be integrated here; the numbers are normally six-digit codes that expire every 30 seconds. This token doesn’t have Bluetooth, SIM, Wi-Fi, internet connection systems, satellites, GPS, and has no way of interrupting the signals to access the code from the device.

Only the device memory can produce a secret key for this device since it has a clock system and a unique ID similar to a MAC address. A clock is used by the hardware token once it has been synchronized with the server. To begin, the token must be connected to the authentication server by the user or administrator. Both the server and token utilize a time synchronization technique to construct a six-digit code using a secret key that delivers the same result on both the security token and the server at the same time.

As a result, each time we attempt to log in, the security token devices display a new six-digit pin that varies every 30 seconds. As a result, the code created by the security token must be input in real-time, and the server validates the token at the same time. Following the implementation of this system, the user's personal data and information will be safeguarded against different assaults, such as spyware attacks.

1.4. Aim and Objectives

1.4.1. Aim

Hardware token devices, which are specialized portable devices used to confirm a user's identification and prevent unwanted access, will be used in this project. When using a hardware token system, six-digit codes will be included that expire every 30 seconds, providing an additional level of protection. Furthermore, the project intends to enhance the user’s password, data, and, most crucially, account balance.

1.4.2. Objectives

All the major objectives of the projects are:

- Minimize the chance of a person or organization being subjected to an invasion of their data privacy.
- Creating the algorithm needed to generate the 6-digit pins in the token device and on the server-side.
- Analyze the risk factors.
- Multifactor authentication research.
- Clock synchronization is used to establish a link between the server and the token device.
- Identify software-based authentication solutions and their features.
- Examine several OTP systems.
- Establish a secure login mechanism.
- Managing resources allocated to the development of the hardware token
- Integrating the method established in both server-side and hardware tokens.

1.5. Structure of the Report

1.5.1. Background

The report's background ties together the overall concept of the project, explaining the client's needs. This section contains research material for comparable projects, as well as a comparison table outlining the key features and components of the project "Implementation of hardware token for user authentication."

1.5.2. Development

The word development implies that this section covers all of the project's development stages. This section summarizes the methods used to kick-start the development. The stages of the project's development are segmented and shown in a work breakdown structure and the project's estimated time management is displayed in a Gantt Chart as a timeline.

1.5.3. Analysis of Progress

This section of the report presents the research and development progress of the project. In this part, a progress table is created which shows the tasks completed till date and tasks to be completed in the date expected. In my case, this section explains all the shortcomings faced beyond the expectations presented in the proposal’s Gantt chart. A progressive solution is represented as an action plan to recover the time loss to accomplish the project within the time frame.

1.5.4. Future Work

The last part of this documentation is the future work. In this part, the development and documentation phases that are planned yet to be completed are included. Some of the adoptions plan to accomplish the project in the predicted timeframe is defined in this part.

2. Background

2.1. Client's Description and Requirements

2.1.1. Client's Name and Description

- **Name of the Client:** Shrawan Shrestha
- **Description of the client:**

The client of the project is Mr. Shrawan Shrestha. He is currently employed as an ITS Lead and IT manager at United Mission to Nepal, which is based in Thapathali, Kathmandu, Nepal. Mr. Shrawan Shrestha has consented to serve as the project's client since he believes this project would be very beneficial in his line of work. He is willing to work with the ideas and to meet the conditions that have been set out.

2.1.2. Client's Requirements

- Hardware token is needed.
- Web application is needed where multi factors authentications must be generated during the log in credentials.
- Low Cost
- Easy to use

2.2. Understanding the Project

2.2.1. Project Elaboration

The project uses the Arduino Nano and RTC Module DS3231. Multiple features of hardware token authentication are controlled by an Arduino Nano and DS3231 module. The Real-Time Clock (RTC) Module DS3231 is used to record the real-time clock, which aids in the synchronization of the clock with the backend. The Arduino Nano, which is utilized in the project, contributes to the running of the algorithm, which is responsible for generating a new TOTP every 30 seconds. Arduino Nano and the RTC module DS3231 are used to produce TOTPs once they have been generated in order to login to the system. To put it another way, this method contributes to the enhancement of system security.

2.2.2. Project Deliverables

The project's primary target audience consists of offices and company professionals who are in charge of the major systems where the login credentials are utilized. As the title suggests, the project is primarily targeted at businesses, as we have seen lot of the companies have suffered reputational damage as a result of data breaches caused by spyware. As a result, one of the most significant effects is the addition of an additional layer of security to the systems of those companies or offices.

2.3. Similar Projects Review

2.3.1. Project 1: RSA SecurID Hardware

RSA SecurID, a multi-factor authentication technology developed by RSA Data Security, protects apps and websites. Organizations and individuals are protected by the RSA SecurID hardware token's ability to verify login credentials and the token's PIN code. It's now possible to buy RSA SecurID equipment tokens that produce and display unique codes every 60 seconds (RSA, 2015).

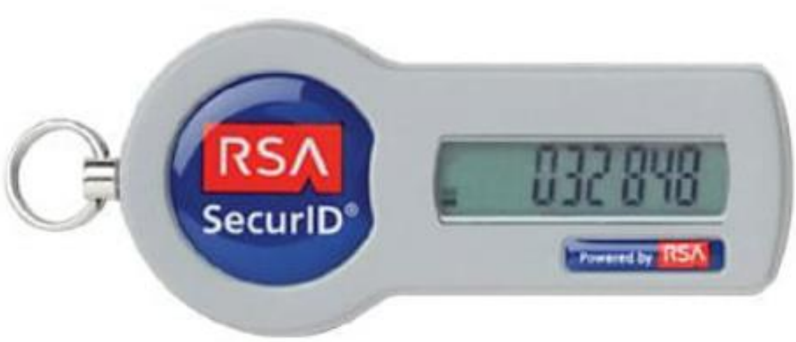


Figure 2:RSA SecurID Hardware Token

Users are asked for their password whenever they try to get access to a secure service. The SecurID system's PIN and the code produced by the client's authenticator token are used to create the password for each session. Session codes are generated every 60 seconds by tapping on their RSA SecurID device. The codes are then examined and verified by the RSA Authentication Manager program. No access is granted until an RSA SecurID token's current state and the client's input are compared.

Token 700 and 800 are offered in two models: RSA SecurID Hardware Token 700 and 800. The RSA SecurID 800, on the other hand, is a USB device that stores login credentials, functions as a master key for various authentication methods, and eliminates the need for the client to enter their code. The RSA SecurID 700 key fob, which can be attached to any key ring and carried in a client's pocket or small bag, will be the basis of our solution (RSA, 2015).

2.3.2. Project 2: Google Authenticator

Google Authenticator is a mobile security application based on two-factor authentication (2FA) that helps to verify user identities before granting them access to websites and services (Wigmore, 2014).

Google Authenticator is used to authenticate users using two-step verification based on Time-based One-Time Passwords (TOTP) and HMAC-based One-Time Passwords (HOTP). TOTP is a one-time password generation technique that uses a shared secret key and the current time to generate a one-time password. HTOP is a one-time password generation technique that makes use of the HMAC algorithm (Lodha, 2018).

For two-factor authentication (2FA), Google Authenticator serves as a security tool. September 20, 2010 was the day it was released by Google. Google Authenticator is a Java- and Objective-C-based app for Android and iOS that employs two methods to verify identity: HMAC-based (HOTP) and Time-based HOTP (TOTP). Using the HMAC algorithm, HOTP generates a one-time password (Hash-based Message Authentication Code). Passcodes are generated using a secret key and the current time, as defined in the RFC 6238 standard for TOTP, which creates a six-digit passcode. Every 30 seconds, a new passcode is generated (Lodha, 2018)

Authenticator works with any website or service that accepts two-factor authentication. As with the bulk of web-based two-factor authentication methods, the system integrates knowledge and control elements. To access websites or web-based services, the user enters his or her regular username and password, followed by a one-time passcode (OTP) generated by their device upon login. This combination assures that the same person who inputs login information on the site is also the owner of the device on which the Google Authenticator software was downloaded (Wigmore, 2014).

Learned how to establish secure sign-ins about any web-based or digital system from the Google Authenticator for our project. I studied Google Authenticator's process and came up with a program that produces a one-time password good for 30 seconds and then automatically generates another entirely different password. In the case that the hardware token is lost or destroyed, we have implemented google-authenticator as a backup token generator. In contrast to Google Authenticator's software-based approach, we concentrate on hardware-based technology (Wigmore, 2014).

2.3.3. Project 3: Arduino TOTP Generator

Author: Shiv

When the button is pressed, this project creates a new code every 30 seconds using a pre-shared key and the current time (which is kept track of using the real-time clock module) and displays it on the display. The most typical use case would be two-factor authentication using Time-based One Time Password (TOTP) and HMAC-based One Time Password (HOTP).

TOTP is an algorithm for generating a one-time password using a shared secret key and the current time. HTOP is a one-time password generation technique that use the HMAC algorithm. TOTP technology is already used by Google, Microsoft, and Steam for two-factor authentication (Shiv2132, 2020).

2.3.4. Time-Based One-Time Passwords with an Arduino

Author: Mike

In addition to providing a platform for token generation, OATH is an open standard authentication method that offers a higher level of security than a password alone. Companies that supply hardware-based dongles for remote logging in are often using the TOTP method. But even if it has been hacked, passwords alone do not provide enough protection. Even while you may be attempting to conceal industrial or government secrets by designing it around an Arduino, we think you are just interested in learning.

One button and a screen are all that's needed to get started with the hardware setup. When it comes to keeping accurate time, this device contains an oscillator made of crystal (as long as it remains powered). Every thirty seconds, a new token will be sent out in this project (Mike, 2012).

2.4. Comparison Table

S.N	Features	Project 1	Project 2	Project 3	Project 4	My Project
1.	Microcontroller used	RSA SecurID	✗	Arduino Nano	Arduino Uno	Arduino Nano
2.	LCD Display	✓	✗	✗	✓	✓
3.	Six-digit OTP	✓	✓	✗	✓	✓
4.	Web Application	✓	✓	✗	✗	✓
5.	Additional Hashing Algorithm	✓	✓	✗	✗	✓
6.	Multiple Propose	✓	✓	✗	✗	✗
7.	Additional Hardware Requirement	✓	✗	✓	✓	✓
8.	Easy to use	✓	✓	✓	✓	✓

Table 1: Project comparison table.

2.5. Analysis and Conclusion of the Comparison

Project 3 and Project 4 have a significant influence on my ongoing project. Because project 3 lacks certain features, I got help from project 4. Web applications are not accessible in Project 4 thus a web application was created using reference material from Project 1. Additionally, projects 3 and 4 do not have an additional hashing algorithm, therefore for the security enhancement, I included the same hashing algorithm as in projects 1 and 2.

So, essentially, I used projects 3 and 4 as the base for my project, and then I added the remaining features from projects 1 and 2. In this manner, I accomplished my assignment.

3. Development

2.5. Considered Methodologies for Project Management

2.5.1. Waterfall Methodology

A waterfall is a process that happens in a certain order. It's also heavily reliant on requirements. You must be confident of the project's needs before proceeding. There is no turning back once a project is started. There are various stages to the Waterfall method. Begin by collecting requirements, designing a solution, putting it into action, and fixing any defects. This approach is self-contained; you must complete each step before proceeding to the next (Cohen, 2019).

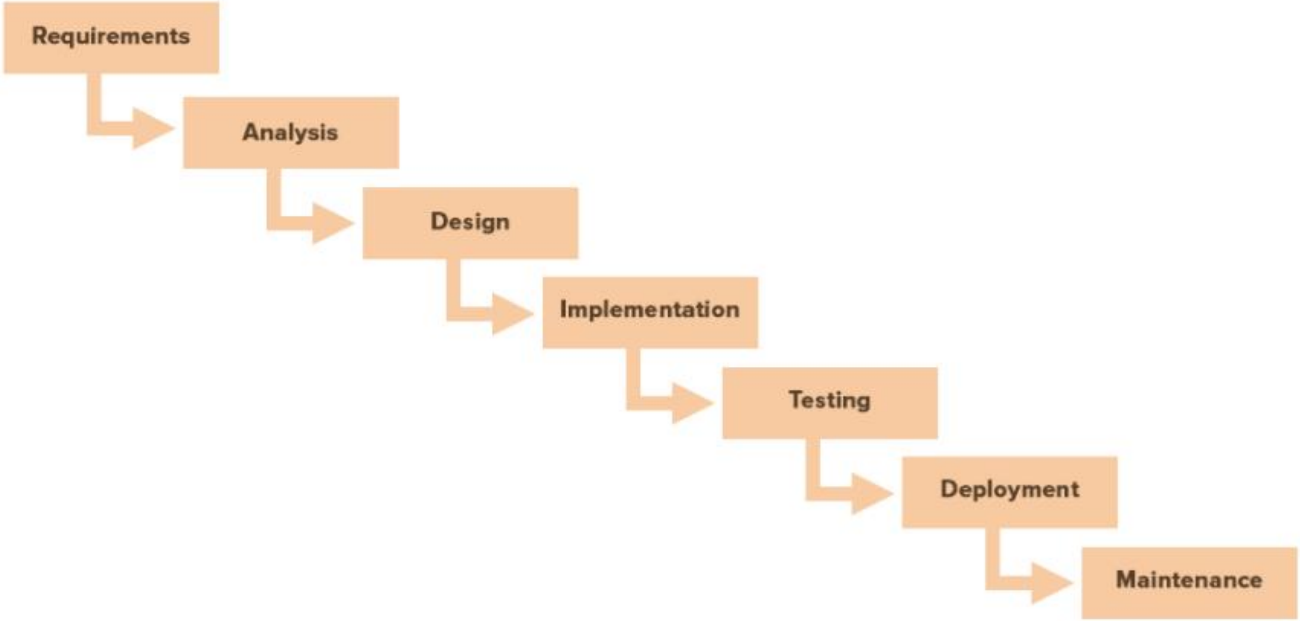


Figure 3: Waterfall Methodology (Cohen, 2019).

2.5.2. Hybrid Methodology

When using the Hybrid technique, you blend both the Waterfall and Agile methodologies. Waterfall and Agile are combined into a system that may be used on several projects at the same time. With the help of Waterfall and Agile methods, the hybrid method is created. If you have a low fund, you may use this method. Even if a number of tests are proposed, The final product design has been decided upon (Cohen, 2019).

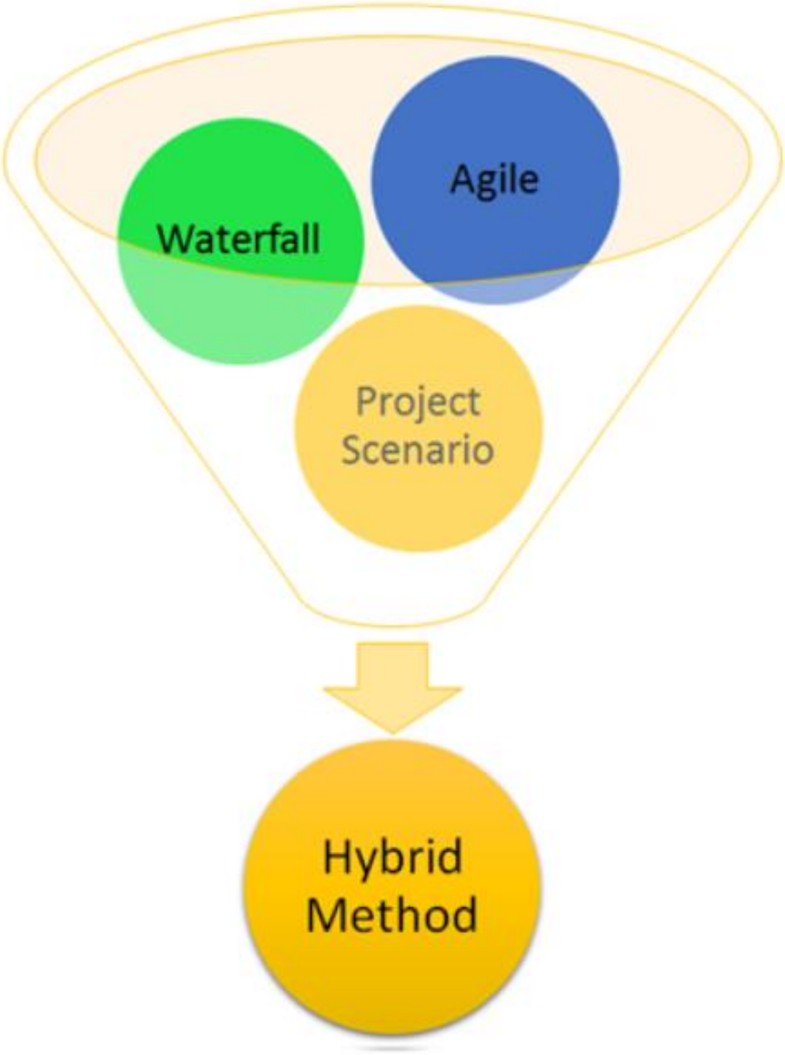


Figure 4: The Hybrid Methodology (Karuna, 2015).

2.5.3. Prototype Methodology

Before real software development can start, a functional prototype of the system must be made. This is how the prototype model works. A prototype is a small version of the system that you can try out before you make the real thing. When compared to real software, a prototype is often a very basic version of the real thing, with limited functional capabilities, poor reliability, and inefficient performance. Many times, the customer only has a vague idea of what the software is supposed to do. Prototypes can be used when there isn't enough information about what the system needs to do, how it needs to do it, or what it needs to produce (JavaTpoint, 2021).

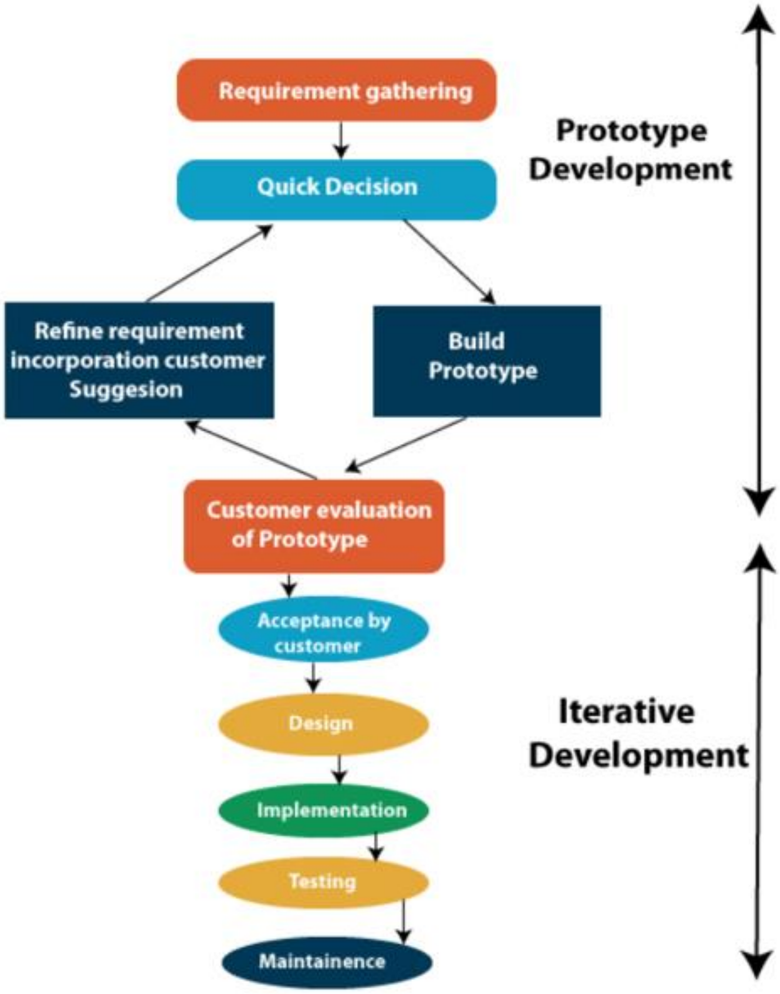


Figure 5: Prototype Methodology (JavaTpoint, 2021).

2.6. Selected Methodology

2.6.1. Agile Methodology

Most often, agile techniques can be found in the field of computer software development. It relies on cross-functional teams and their customers to work together to identify problems and produce solutions (Asmo, 2018).Agile Methodology is a people-focused, results-focused approach to software development that respects our rapidly changing world. It's centered around adaptive planning, self-organization, and short delivery times. It's flexible, fast, and aims for continuous improvements in quality, using tools like Scrum and extreme Programming (Altvater, 2017).

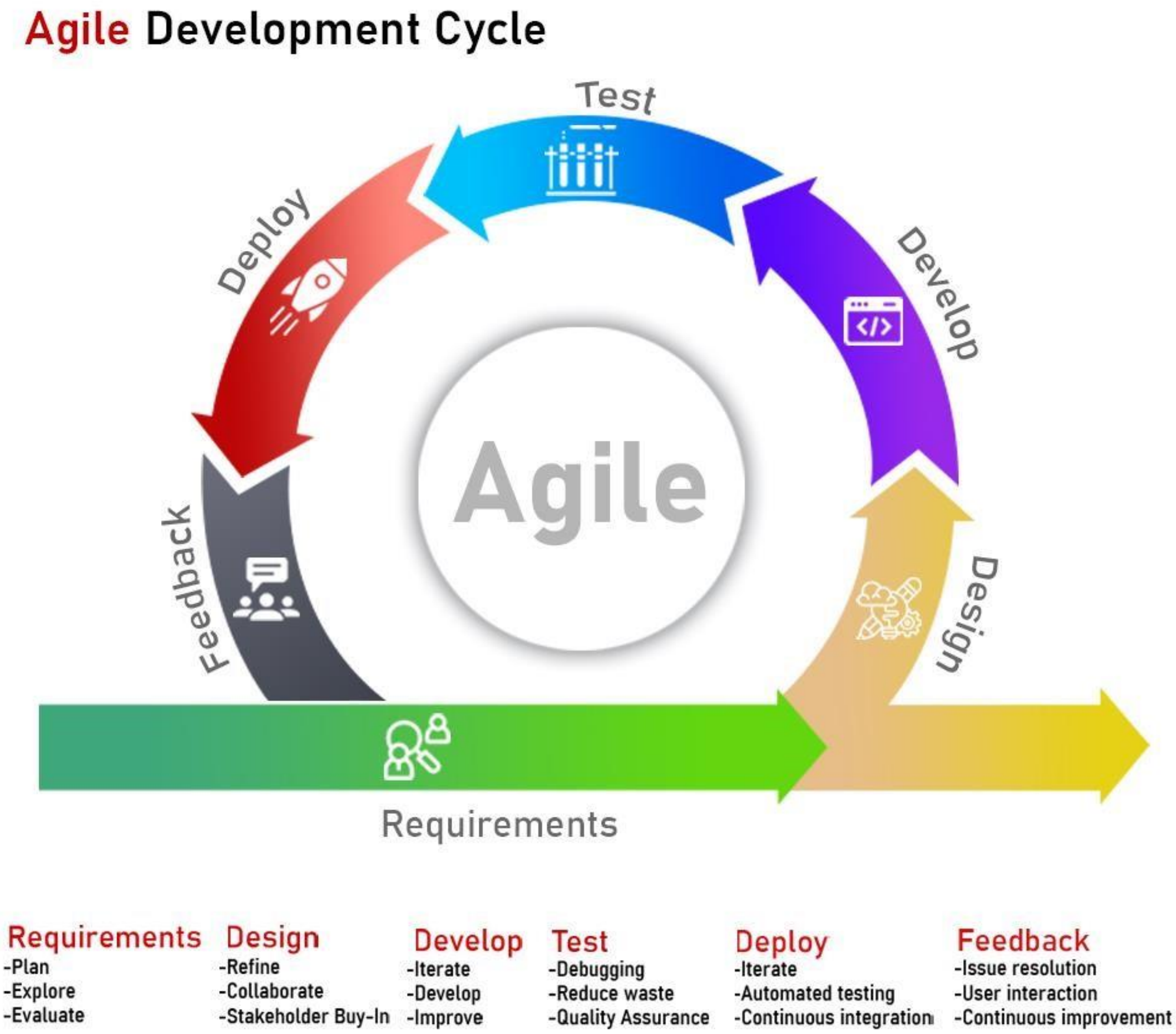


Figure 6: Agile Methodology

- **Benefits of having an agile methodology for project management**

The "Hardware Token for User Authentication" project is heavily influenced by the Agile Methodology. A more involved and agile mindset is closely linked to the advantages of Agile. All in all, it provides exactly what the consumer wants and when they want it. As a result, less time is wasted developing in the incorrect direction, and the system as a whole respond to changes more quickly.

- **Advantages of Agile Methodology**
- Rapid, ongoing, and beneficial software development is the key to customer pleasure.
- On a frequent basis, the customer, developer, and product owner engage to emphasize results rather than procedures and tools.
- The product is rapidly created and provided on a regular basis (weeks rather than months.)
- A face-to-face conversation is the best form of communication.
- It maintained a constant focus on both technological and design excellence.
- The business people and the developers work together every day.
- Constantly adapting to changes in the environment.
- Late adjustments in criteria are appreciated, even if they are minor (javatpoint, 2020).

2.7. Work Breakdown Structure

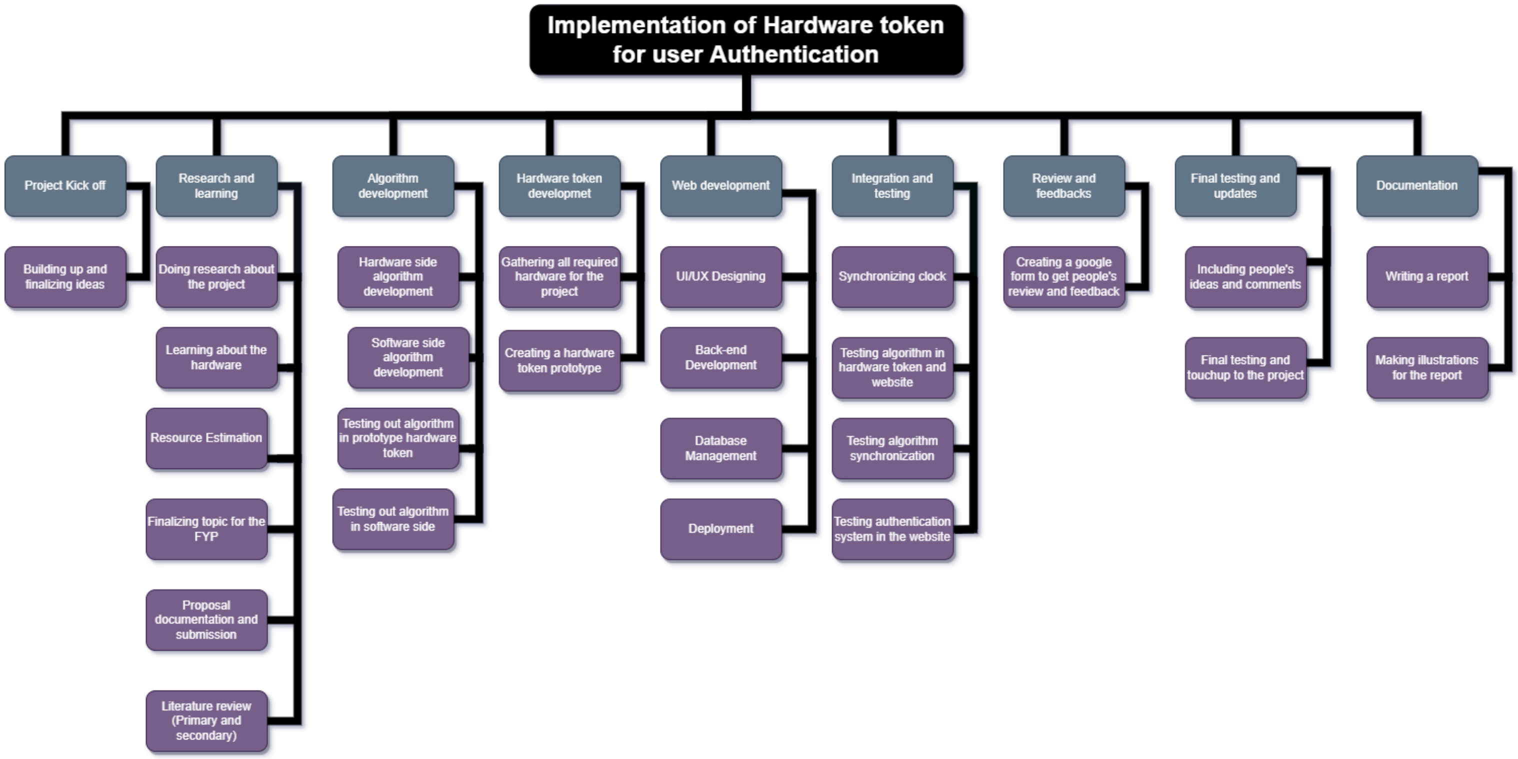
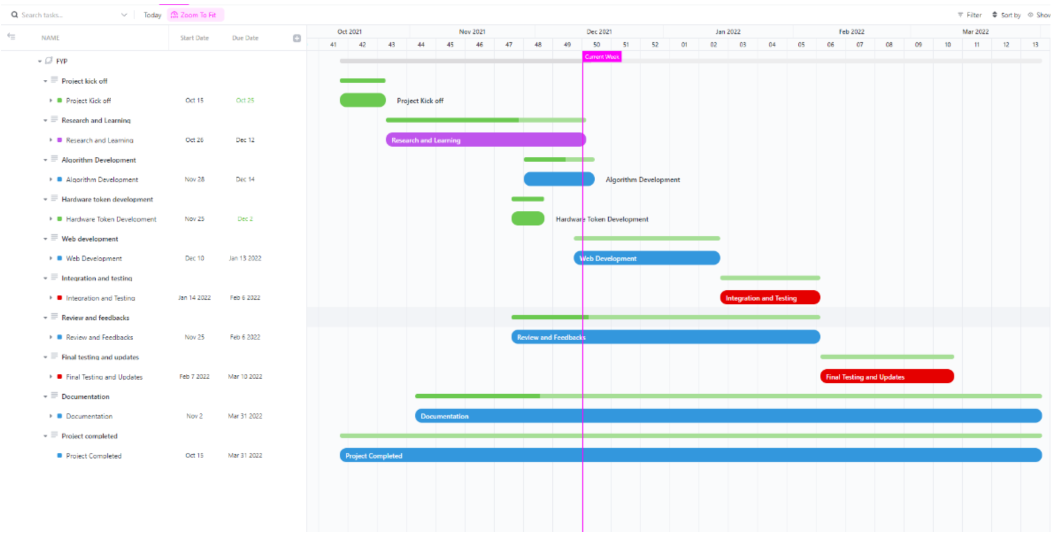

Implementation of Hardware token for user Authentication
Project Kick off
Building up and finalizing ideas
Research and learning
Doing research about the project
Learning about the hardware
Resource Estimation
Finalizing topic for the FYP
Proposal documentation and submission
Literature review (Primary and secondary)
Algorithm development
Hardware side algorithm development
Software side algorithm development
Testing out algorithm in prototype hardware token
Testing out algorithm in software side
Hardware token developmet
Gathering all required hardware for the project
Creating a hardware token prototype
Web development
UI/UX Designing
Back-end Development
Database Management
Deployment
Integration and testing
Synchronizing clock
Testing algorithm in hardware token and website
Testing algorithm synchronization
Testing authentication system in the website
Review and feedbacks
Creating a google form to get people's review and feedback
Final testing and updates
Including people's ideas and comments
Final testing and touchup to the project
Documentation
Writing a report
Making illustrations for the report

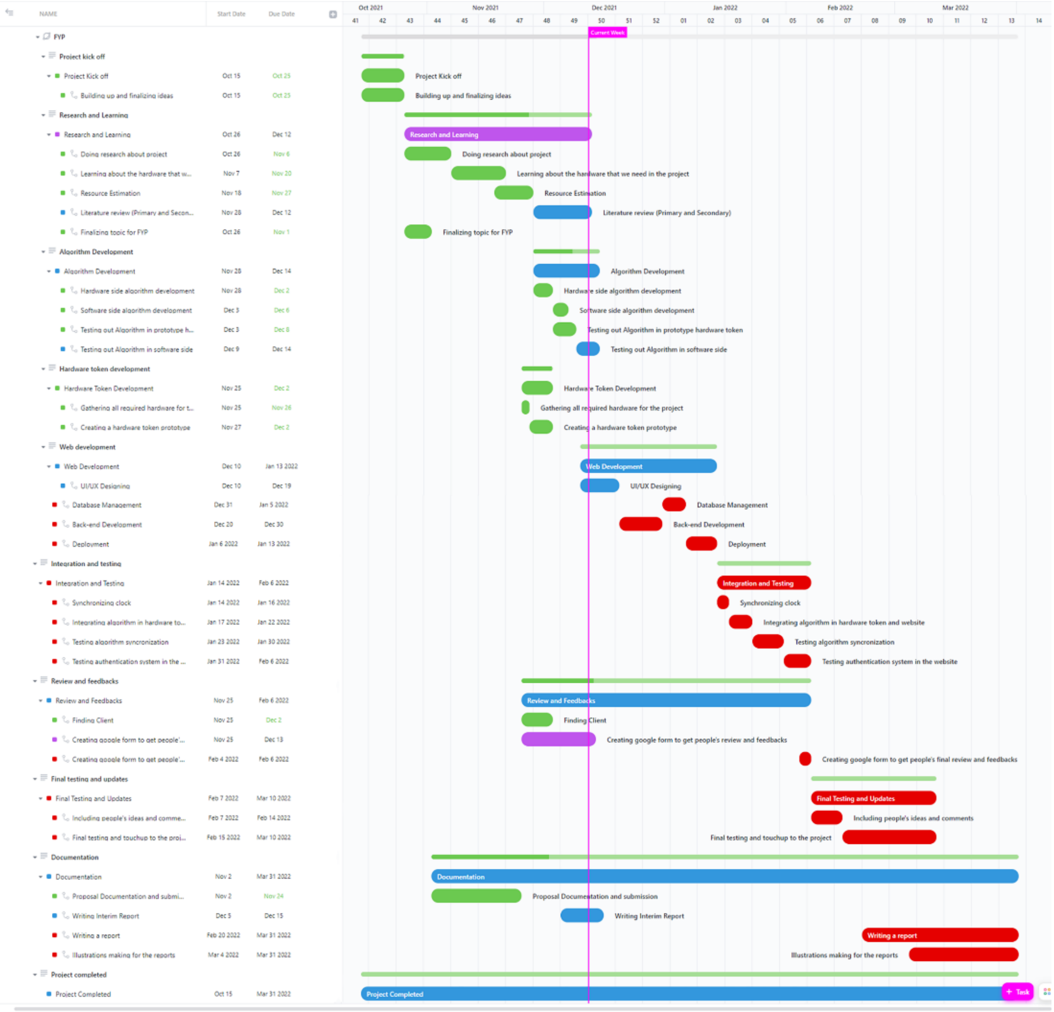
Figure 7: Work Breakdown of the project

2.8. Gantt Chart



NAME	Start Date	Due Date
FYP		
Project kick off		
Project Kick off	Oct 15	Oct 25
Research and Learning		
Research and Learning	Oct 26	Dec 12
Algorithm Development		
Algorithm Development	Nov 28	Dec 14
Hardware token development		
Hardware Token Development	Nov 25	Dec 2
Web development		
Web Development	Dec 10	Jan 13 2022
Integration and testing		
Integration and Testing	Jan 14 2022	Feb 6 2022
Review and feedbacks		
Review and Feedbacks	Nov 25	Feb 6 2022
Final testing and updates		
Final Testing and Updates	Feb 7 2022	Mar 10 2022
Documentation		
Documentation	Nov 2	Mar 31 2022
Project completed		
Project Completed	Oct 15	Mar 31 2022

Figure 8 : Gantt Chart of the Project



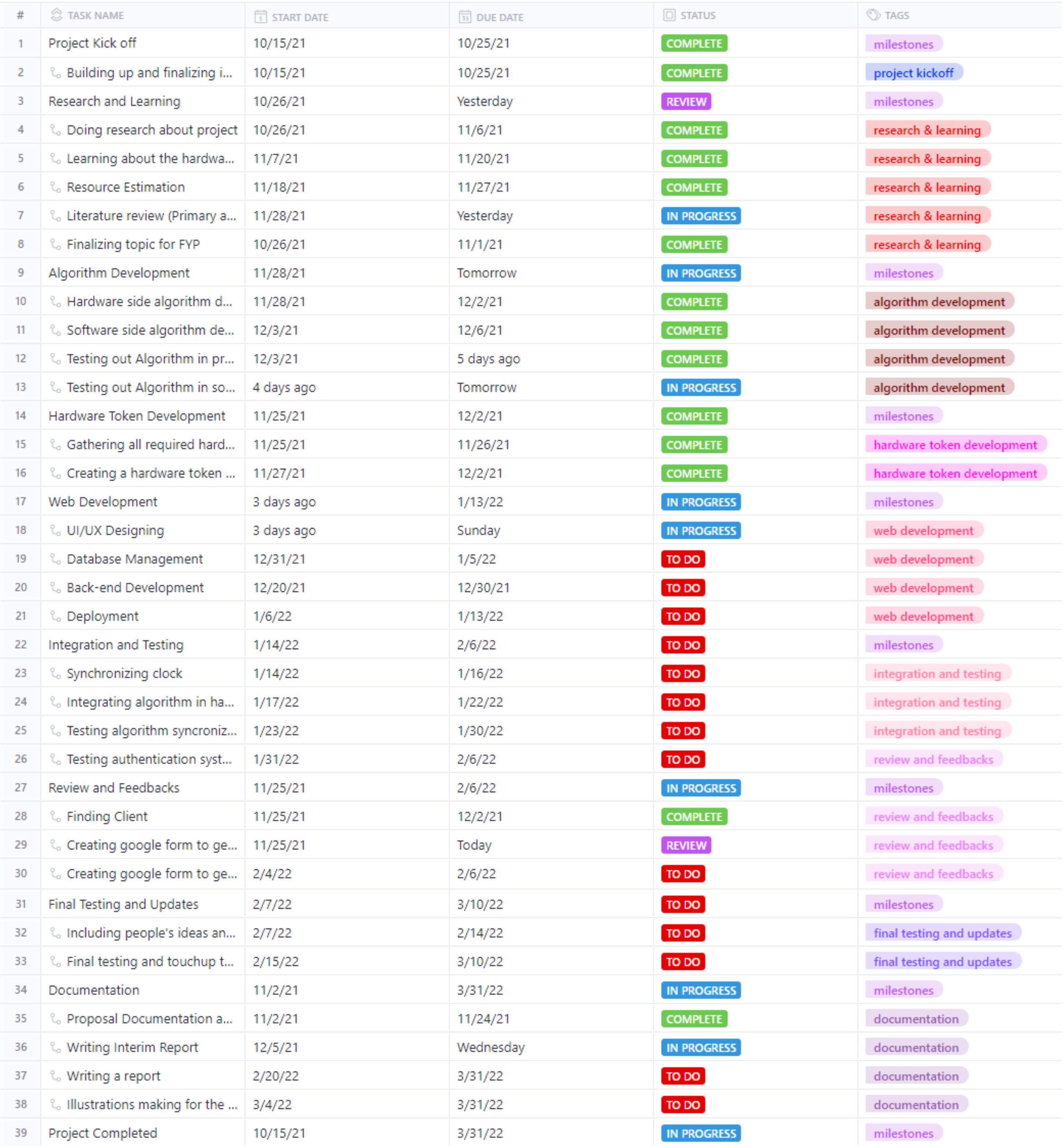
NAME	Start Date	Due Date
FYP		
Project kick off		
Project Kick off	Oct 15	Oct 25
Building up and finalizing ideas	Oct 15	Oct 25
Research and Learning		
Research and Learning	Oct 26	Dec 12
Doing research about project	Oct 26	Nov 6
Learning about the hardware that w...	Nov 7	Nov 20
Resource Estimation	Nov 18	Nov 27
Literature review (Primary and Secon...	Nov 28	Dec 12
Finalizing topic for FYP	Oct 26	Nov 1
Algorithm Development		
Algorithm Development	Nov 28	Dec 14
Hardware side algorithm development	Nov 28	Dec 2
Software side algorithm development	Dec 3	Dec 6
Testing out Algorithm in prototype h...	Dec 3	Dec 8
Testing out Algorithm in software side	Dec 9	Dec 14
Hardware token development		
Hardware Token Development	Nov 25	Dec 2
Gathering all required hardware for t...	Nov 25	Nov 26
Creating a hardware token prototype	Nov 27	Dec 2
Web development		
Web Development	Dec 10	Jan 13 2022
UI/UX Designing	Dec 10	Dec 19
Database Management	Dec 31	Jan 5 2022
Back-end Development	Dec 20	Dec 30
Deployment	Jan 6 2022	Jan 13 2022
Integration and testing		
Integration and Testing	Jan 14 2022	Feb 6 2022
Synchronizing clock	Jan 14 2022	Jan 16 2022
Integrating algorithm in hardware to...	Jan 17 2022	Jan 22 2022
Testing algorithm syncronization	Jan 23 2022	Jan 30 2022
Testing authentication system in the ...	Jan 31 2022	Feb 6 2022
Review and feedbacks		
Review and Feedbacks	Nov 25	Feb 6 2022
Finding Client	Nov 25	Dec 2
Creating google form to get people'...	Nov 25	Dec 13
Creating google form to get people'...	Feb 4 2022	Feb 6 2022
Final testing and updates		
Final Testing and Updates	Feb 7 2022	Mar 10 2022
Including people's ideas and comme...	Feb 7 2022	Feb 14 2022
Final testing and touchup to the proj...	Feb 15 2022	Mar 10 2022
Documentation		
Documentation	Nov 2	Mar 31 2022
Proposal Documentation and submi...	Nov 2	Nov 24
Writing Interim Report	Dec 5	Dec 15
Writing a report	Feb 20 2022	Mar 31 2022
Illustrations making for the reports	Mar 4 2022	Mar 31 2022
Project completed		
Project Completed	Oct 15	Mar 31 2022

Figure 9:Detail Gantt Chart of the project

3. Analysis of Progress

3.1. Progress Table

Table 2: Progress Table of the Progress



#	TASK NAME	START DATE	DUE DATE	STATUS	TAGS
1	Project Kick off	10/15/21	10/25/21	COMPLETE	milestones
2	Building up and finalizing i...	10/15/21	10/25/21	COMPLETE	project kickoff
3	Research and Learning	10/26/21	Yesterday	REVIEW	milestones
4	Doing research about project	10/26/21	11/6/21	COMPLETE	research & learning
5	Learning about the hardwa...	11/7/21	11/20/21	COMPLETE	research & learning
6	Resource Estimation	11/18/21	11/27/21	COMPLETE	research & learning
7	Literature review (Primary a...	11/28/21	Yesterday	IN PROGRESS	research & learning
8	Finalizing topic for FYP	10/26/21	11/1/21	COMPLETE	research & learning
9	Algorithm Development	11/28/21	Tomorrow	IN PROGRESS	milestones
10	Hardware side algorithm d...	11/28/21	12/2/21	COMPLETE	algorithm development
11	Software side algorithm de...	12/3/21	12/6/21	COMPLETE	algorithm development
12	Testing out Algorithm in pr...	12/3/21	5 days ago	COMPLETE	algorithm development
13	Testing out Algorithm in so...	4 days ago	Tomorrow	IN PROGRESS	algorithm development
14	Hardware Token Development	11/25/21	12/2/21	COMPLETE	milestones
15	Gathering all required hard...	11/25/21	11/26/21	COMPLETE	hardware token development
16	Creating a hardware token ...	11/27/21	12/2/21	COMPLETE	hardware token development
17	Web Development	3 days ago	1/13/22	IN PROGRESS	milestones
18	UI/UX Designing	3 days ago	Sunday	IN PROGRESS	web development
19	Database Management	12/31/21	1/5/22	TO DO	web development
20	Back-end Development	12/20/21	12/30/21	TO DO	web development
21	Deployment	1/6/22	1/13/22	TO DO	web development
22	Integration and Testing	1/14/22	2/6/22	TO DO	milestones
23	Synchronizing clock	1/14/22	1/16/22	TO DO	integration and testing
24	Integrating algorithm in ha...	1/17/22	1/22/22	TO DO	integration and testing
25	Testing algorithm syncroniz...	1/23/22	1/30/22	TO DO	integration and testing
26	Testing authentication syst...	1/31/22	2/6/22	TO DO	review and feedbacks
27	Review and Feedbacks	11/25/21	2/6/22	IN PROGRESS	milestones
28	Finding Client	11/25/21	12/2/21	COMPLETE	review and feedbacks
29	Creating google form to ge...	11/25/21	Today	REVIEW	review and feedbacks
30	Creating google form to ge...	2/4/22	2/6/22	TO DO	review and feedbacks
31	Final Testing and Updates	2/7/22	3/10/22	TO DO	milestones
32	Including people's ideas an...	2/7/22	2/14/22	TO DO	final testing and updates
33	Final testing and touchup t...	2/15/22	3/10/22	TO DO	final testing and updates
34	Documentation	11/2/21	3/31/22	IN PROGRESS	milestones
35	Proposal Documentation a...	11/2/21	11/24/21	COMPLETE	documentation
36	Writing Interim Report	12/5/21	Wednesday	IN PROGRESS	documentation
37	Writing a report	2/20/22	3/31/22	TO DO	documentation
38	Illustrations making for the ...	3/4/22	3/31/22	TO DO	documentation
39	Project Completed	10/15/21	3/31/22	IN PROGRESS	milestones

Figure 10: Progress Table of the project

3.2. Progress Review

3.2.1. Project Plan, Design, and Requirements

The process of project management has been handed over to the principles of Agile Methodology. The Gantt chart has been followed for the entire process handling of the project. In the initial phase, the process of topic selection and feasibility study was carried out for making sure if the topic is viable or not. The survey was conducted among 36 people to collect the data then a client was selected.

It helped in having an idea for the appropriate features and usability to add up in the project. Based on the requirements the cost estimation was done and most of the hardware component was purchased to start with the development work. In addition to that, research on similar projects and tools and technique was done to have an in-depth. **(Resource Requirements: Appendix 1)**

3.2.2. Progress Timeline

The project has progressed in accordance with the time estimates provided in the Gantt chart. Prior to creating the Gantt chart, the project's research and learning about the hardware tokens were completed. Following that, algorithm development started, with both hardware and software side algorithm development accomplished. Before developing the method, the algorithm was tested on prototype hardware tokens and on the software side. All of the necessary hardware for this project was collected during hardware token development, and the hardware token prototype was constructed. According to the Gantt Chart given in the interim report, the UI/UX design could not be finished on time, despite the fact that everything else indicated was done on time. The algorithm development has been accomplished to this point.

3.2.3. Action Plan

The project's tasks will be completed in accordance with the updated Gantt chart created for the interim report. Requirements, design, and development were accomplished in accordance with the Gantt chart and the selected methodology. The development work has been completed. When this interim report is submitted, the next phases will be completed.

4. Future Work

4.1. Phases to Complete

4.1.1. Web Development

In this phase, we are presently working on UI/UX design. The front-end work will be finished after the UI/UX design is finalized. As a next step, JS will be used to create the web application, which will have both a front-end and a back-end. After the back-end and front-end are completed, GraphQL (a query language) will be utilized to interact between the front-end and back-end. POSTGRESQL will now be used as a database for web development, and Heroku will be used as a database as a service. As a GraphQL interface, the Apollo Client and Server will be used. Finally, it will be deployed in virtual reality when the structure coding is completed. In this way, we will reach a significant milestone in our development process. This is the beginning of the running phase. This phase will take 33 days if all goes according to plan.

4.1.2. Integration and Testing

Getting the hardware token and server site clocks in sync is step one in this phase. A test will then be run to see whether the algorithm is functioning in the web application once the hardware token and web application are integrated. The hardware token will be used to test the authentication system if all goes as planned. According to the Gantt chart, this phase will continue from January 14 to February 6.

4.1.3. Final Testing and Updates

The third stage of the project involves doing the web application test to see whether the problem still exists or not. A final touchup will be done if the defect is discovered to be minimal or eliminated. Finally, it involves gathering the results of the survey's participants' opinions. From there, we may use the people's feedback to help us improve our application. The project will be completed at that point. The project will be completed at that time.

4.1.4. Documentation

This is the last step, in which the prototype that has been thoroughly tested is put into action on the real-world environment of the customer. The final documentation is also completed at this phase of the project's development. As a result, the project's final report will be completed and sent.

5. Conclusion

To summarize what's been said earlier, this report demonstrates that the client was chosen, the project plan was created, and the development processes are detailed in the report. The development process shows the hardware token prototype development, back-end token generating algorithm, hardware token algorithms, and login UI design are demonstrated in this report. Although the project appeared to be somewhat complex and messy at times, overcoming the milestones was an incentive to do more and more.

6. References

Altvater, A. (2017, September 17). *What is Agile Methodology? How It Works, Best Practices, Tools – Stackify*. Retrieved December 10, 2021, from Stackify: https://stackify.com/agile-methodology/

Asmo. (2018, March 2). *Zenkit*. Retrieved November 21, 2021, from Agile Methodology: An Overview - Zenkit.

Cohen, E. (2019, July 1). *Workamajig blog*. Retrieved December 11, 2021, from The Definitive Guide to Project Management Methodologies: https://www.workamajig.com/blog/project-management-methodologies

Gillis, A. S. (2021, July 01). *What is Spyware?* Retrieved November 23, 2021, from SearchSecurity: https://www.techtarget.com/searchsecurity/definition/spyware

HIDEEZ. (2021, February 17). *Hideez*. Retrieved 11 22, 2021, from What is a Hardware Token? Hard Tokens vs. Soft Tokens &ndash; Hideez: https://hideez.com/blogs/news/hardware-tokens

javatpoint. (2020). *Advantage of Agile Methodology* | *Disadvantage of Agile Methodology - Javat*. Retrieved December 13, 2021, from www.javatpoint.com: https://www.javatpoint.com/advantage-and-disadvantage-of-agile-methodology

JavaTpoint. (2021). *www.javatpoint.com*. Retrieved December 9, 2021, from Prototype Model (Software Engineering) - javatpoint: https://www.javatpoint.com/software-engineering-prototype-model

Karuna, V. (2015, October 10). *Agile + Waterfall = Hybrid Method; when does it have the right application?* Retrieved December 14, 2021, from Agile Gnostic: https://agilegnostic.wordpress.com/2015/10/10/agile-waterfall-hybrid-method-when-does-it-have-the-right-application/

Kaspersky. (2021, July 28). *What is Spyware? Protect Yourself from Spyware* | *Kaspersky*. Retrieved November 23, 2021, from usa.kaspersky.com: https://usa.kaspersky.com/resource-center/threats/spyware

Lodha, T. (2018, April 17). *Google Authenticator and how it works?* Retrieved December 14, 2021, from Medium: https://medium.com/@tilaklodha/google-authenticator-and-how-it-works-2933a4ece8c2

Mike. (2012, July 11). *Time-based One-Time Passwords With An Arduino*. Retrieved December 14, 2021, from Hackaday: https://hackaday.com/2012/07/11/time-based-one-time-passwords-with-an-arduino/?fbclid=IwAR2LihAenl2N4B1Wv7f3i64oWV4ZuWSbGO1V6V6XUP03J7gVgUaum3NSygM

RSA. (2015). RSA SECURID®. USA.

Savvy. (2021, June 02). *Savvy Security*. Retrieved November 22, 2021, from What Is Spyware? A Look at Spyware Examples & Types: https://cheapsslsecurity.com/blog/what-is-spyware-a-look-at-spyware-examples-types/

Shiv2132. (2020, January 09). *Arduino TOTP Generator : 3 Steps - Instructables*. Retrieved December 12, 2021, from Instructables: https://www.instructables.com/Arduino-TOTP-Generator/?fbclid=IwAR1fVrYEa1KnJViFEcpFo4NONqAxhpCllkxaM7bae56UqPOfxpOVfMW99_w

Wigmore, I. (2014, December 1). *What is Google Authenticator? - Definition from WhatIs.com*. Retrieved December 12, 2021, from SearchSecurity: https://www.techtarget.com/searchsecurity/definition/Google-Authenticator

8. Appendix

8.1. Appendix 1: Resource Requirements

This project is built on the hardware and software components listed below. To address the requirements of the client, following resources are gathered:

8.1.1. Software Components

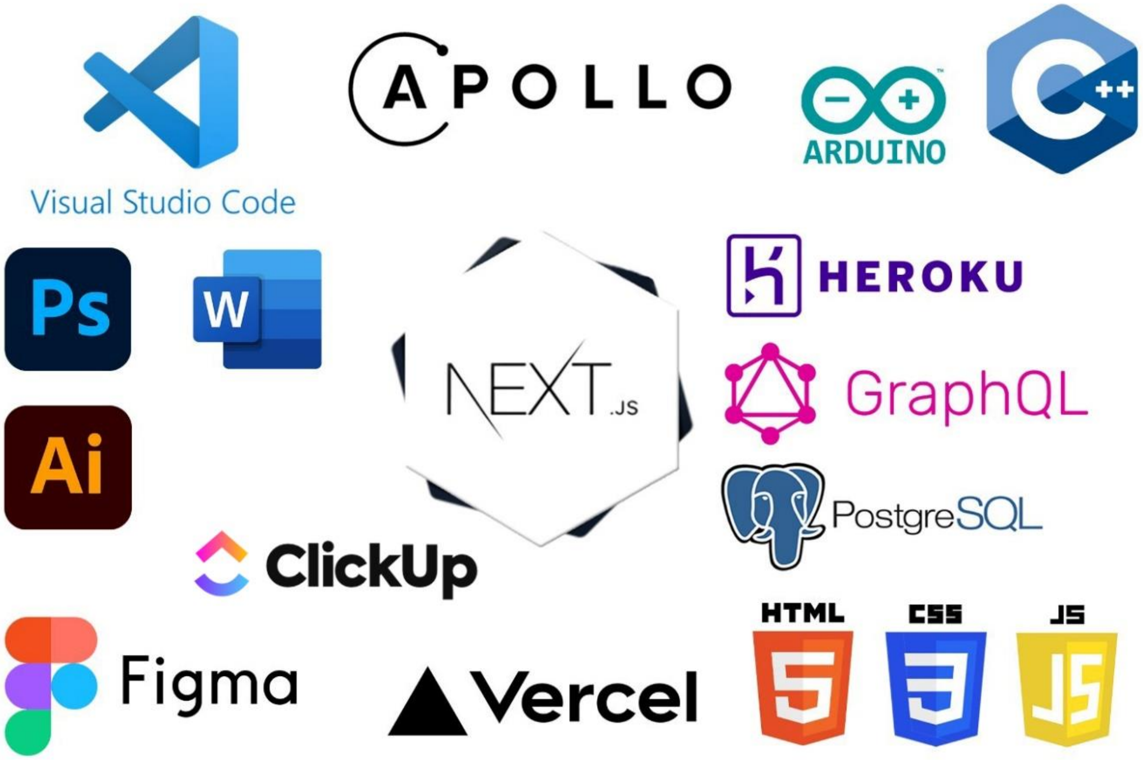


Figure 11: Software and languages required in this project

- **Arduino IDE:** The software for Arduino is called Arduino IDE (Integrated Development Environment). It's a text editor with a variety of capabilities, similar to a notepad. It's used to write code, compile it to see if there are any issues, then upload the code to the Arduino.
- **Visual Studio Code:** Visual Studio Code is a compact code editor that includes features for debugging, task execution, and version management. It seeks to provide developers just the tools they need for a rapid code-build-debug cycle, leaving more sophisticated processes to full-featured IDEs like Visual Studio IDE.

- **C++:** C++ is a general-purpose, case-sensitive, free-form programming language that supports both object-oriented and procedural programming styles. C++ is used to construct an algorithm for generating TOTP for Arduino-powered hardware tokens.
- **NEXT JS**: NEXT JS is used to create both the front-end and back-end of the web application.
- **POSTGRESQL:** POSTGRESQL is a relational database management system.
- **GraphQL:** GraphQL is a query language that is used to interact between the front-end and back-end of a website.
- **Heroku:** Heroku is used as a database as a service (DBaaS).
- **Apollo Client and Server:** Apollo Client and Server are used to interact with GraphQL.
- **Vercel:** Vercel is utilized to deliver our project's web application.

8.1.2. Hardware Components

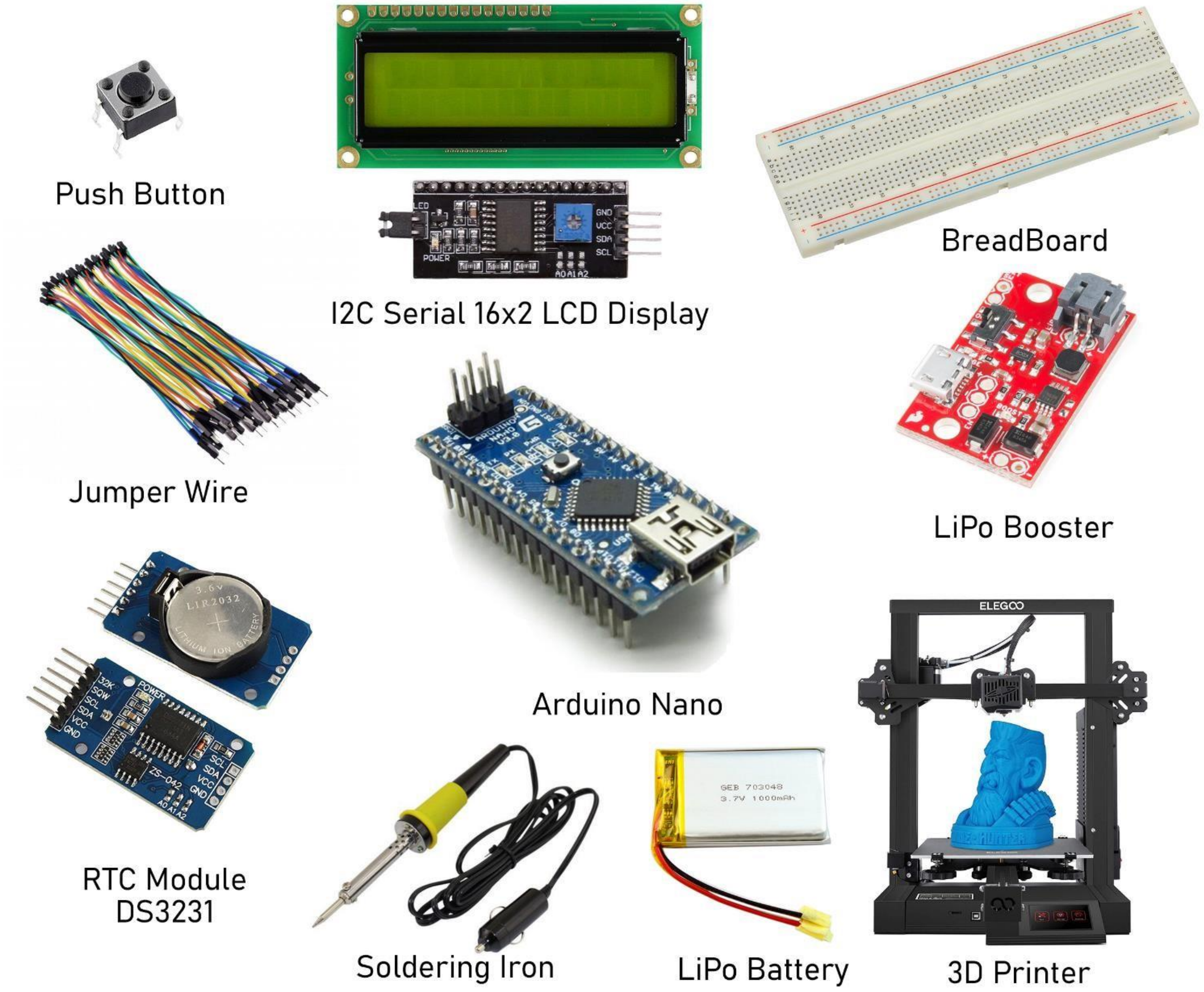


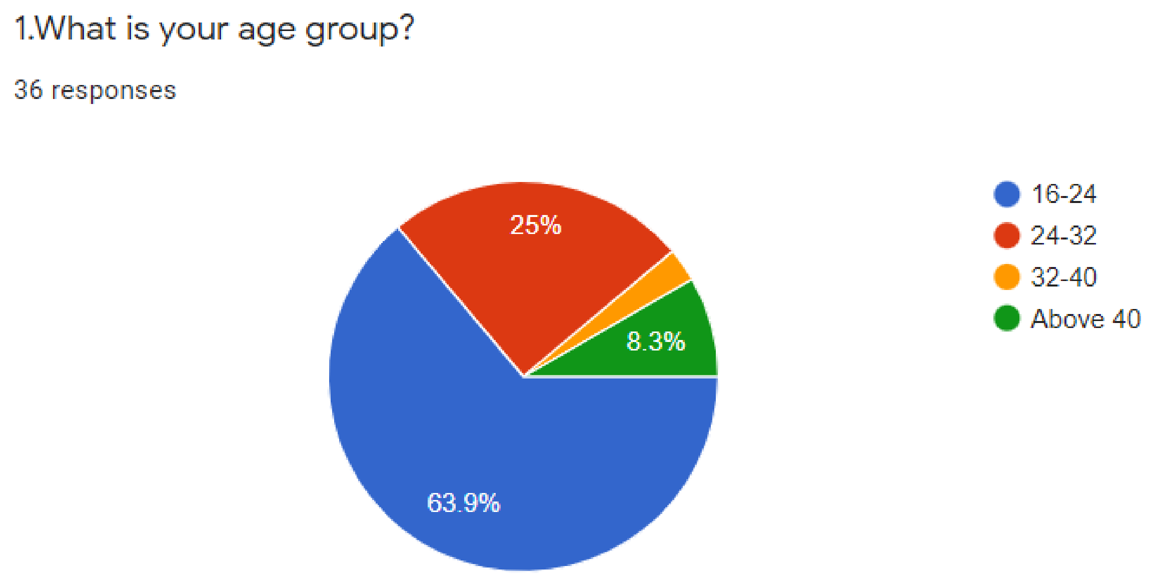
Figure 12: Required Hardware in this Project

- **Arduino Nano:** Our main computing device is an Arduino Nano, which processes algorithms to generate TOTP. The size and mobility of the Arduino Nano are the major reasons for its selection. Based on the ATmega328, the Arduino Nano is a compact, comprehensive, and breadboard-friendly board (Arduino Nano 3.x). It offers a lot of the same features as the Arduino Duemilanove, but it comes in a different packaging. It just has a DC power connector and uses a Mini-B USB cable rather than a conventional one.

- **Breadboard:** A breadboard is used to rapidly construct and test circuits prior to completing any design. The breadboard has several holes for inserting circuit components such as integrated circuits and resistors.
- **RTC Module DS3231:** The DS3231 RTC module is a low-cost, exceptionally accurate I2C real-time clock (RTC) with an inbuilt temperature-compensated crystal oscillator (TCXO) and crystal. When the main power to the machine is interrupted, the device integrates a battery input and maintains precise timekeeping.
- **I2C Serial 16x2 LCD Display:** This 12C 16x2 LCD display is used to show the OTP and the time remaining till the OTP is changed.
- **Push Button:** A push-button (sometimes written pushbutton) is a basic switch mechanism used to control some component of a machine or process. Buttons are frequently constructed of a hard substance, such as plastic or metal. Push buttons are basic switches that regulate the power of a machine or device. To switch on our display, we simply used a push button.
- **Jumper Wires:** Jumper wires are simply wires with connector pins on both ends. They can be used to connect two points to each other without having to solder them to each other. A lot of prototyping tools use jumper wires to make it easy to change a circuit as needed. We used three different types of jumper wires in this project they are: Male to male Jumper Wire, Male to Female Jumper Wire and Female to Female Jumper Wire.
- **Lipo Booster:** The LiPo Booster is a device that accepts a 3.3V input from the LiPo battery and outputs a configurable voltage of 3.3V and 5V, since our display requires 5V.
- **LiPo Battery:** The LiPo Booster is a device that accepts a 3.3V input from the LiPo battery and outputs either 3.3V and 5V, since our display requires 5V.

8.2. Appendix 2: Survey Findings

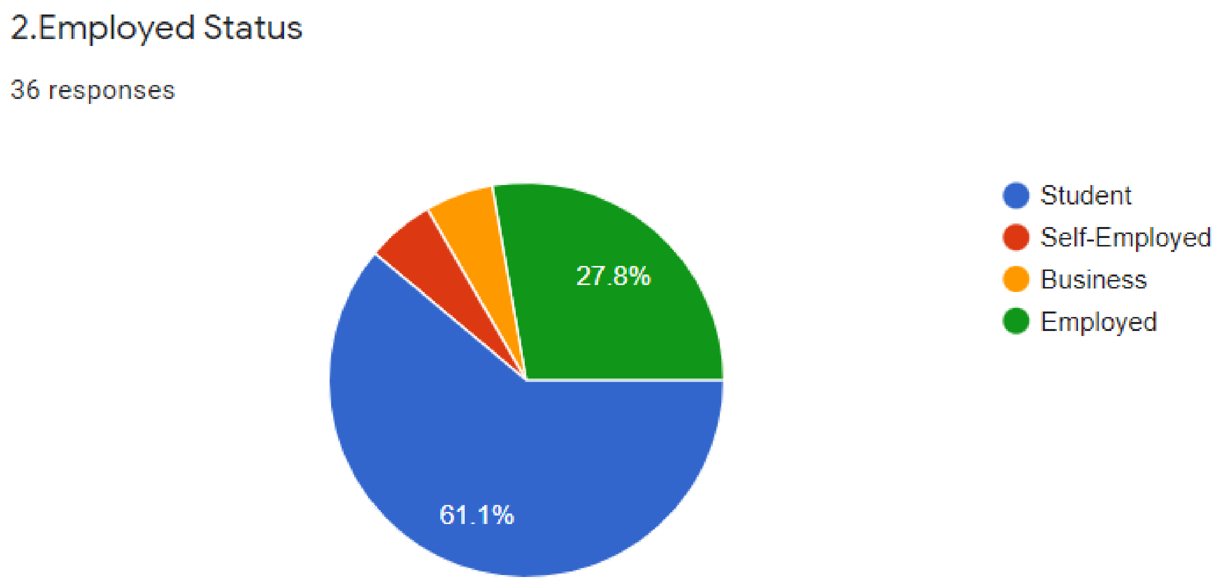
The survey was done among 36 participants who represented a diverse range of ages, jobs, and genders.



1.What is your age group?

36 responses

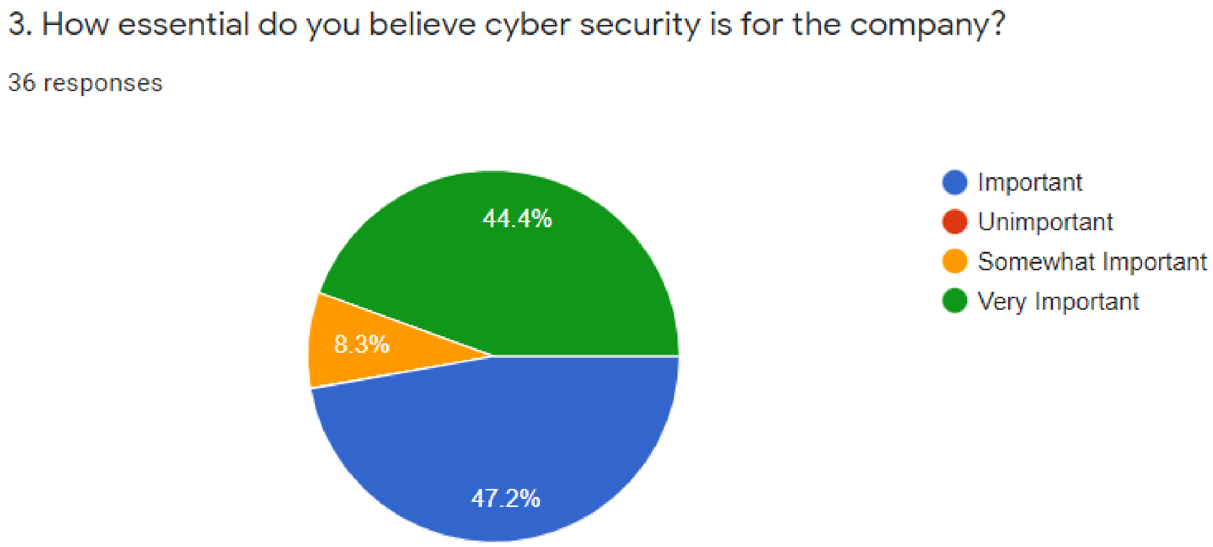
Figure 13: Survey Question no.1



2.Employed Status

36 responses

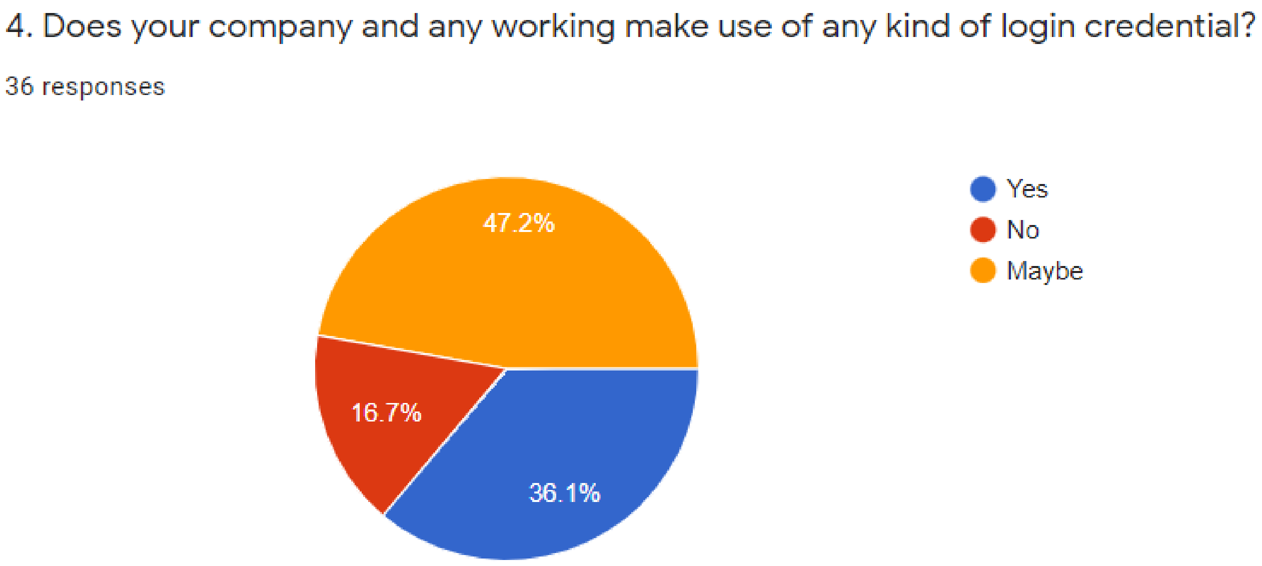
Figure 14: : Survey Question no.2



3. How essential do you believe cyber security is for the company?

36 responses

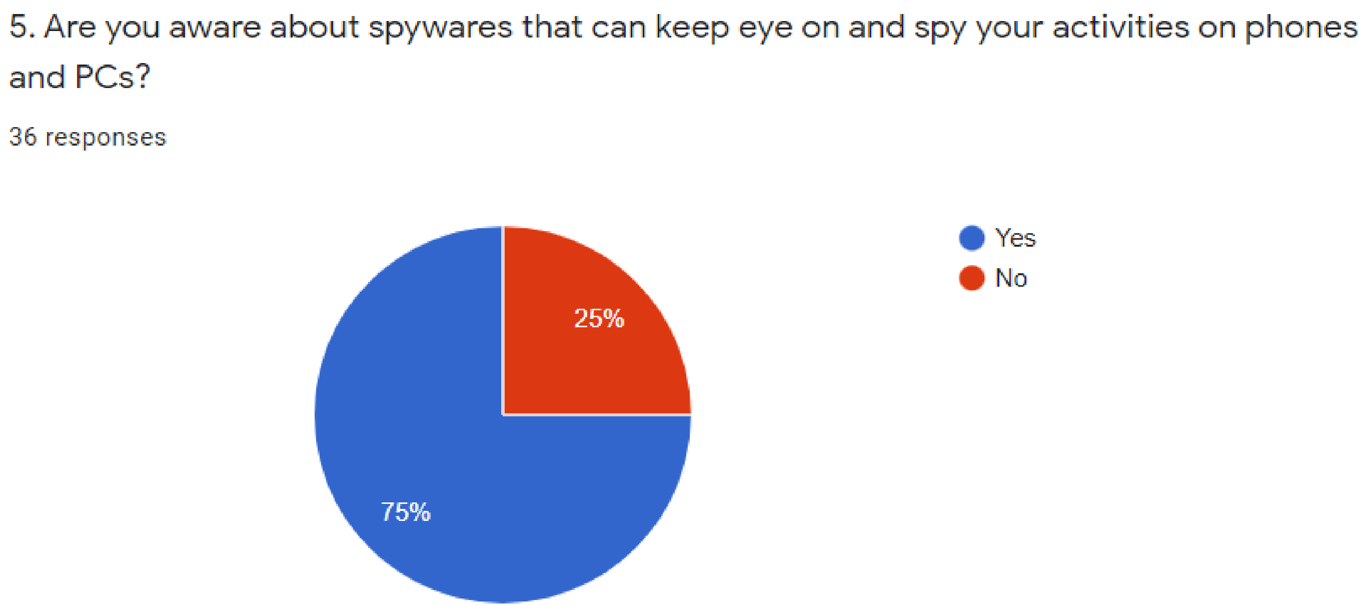
Figure 15: : Survey Question no.3



4. Does your company and any working make use of any kind of login credential?

36 responses

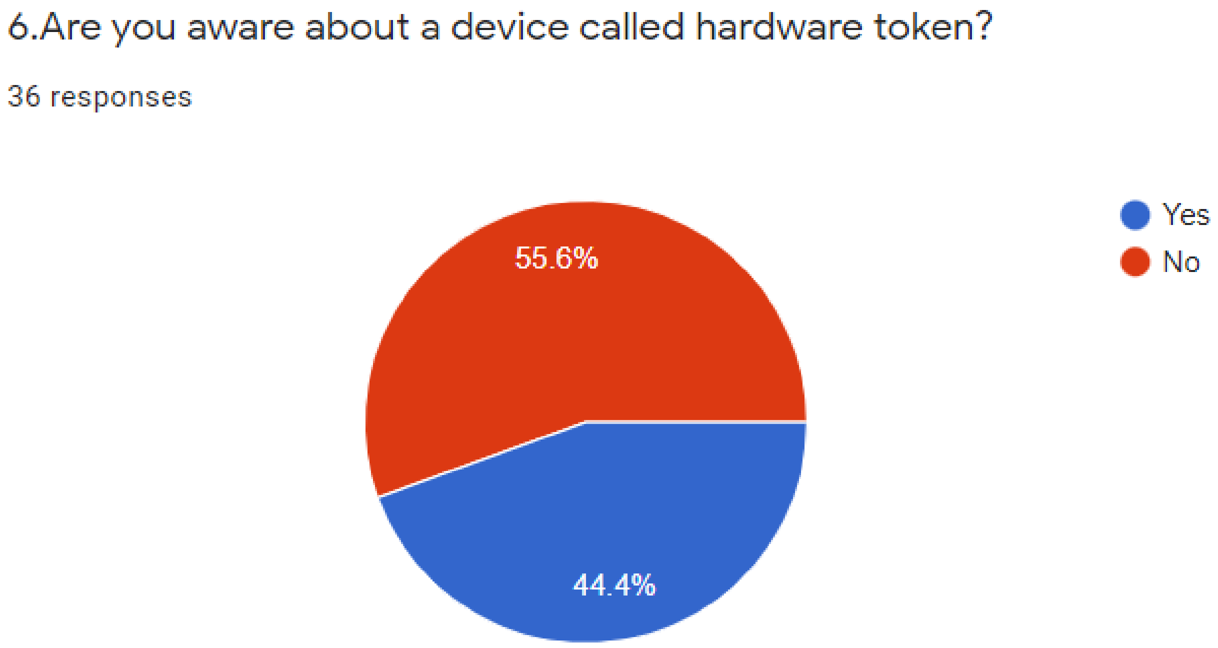
Figure 16: : Survey Question no.4



5. Are you aware about spywares that can keep eye on and spy your activities on phones and PCs?

36 responses

Figure 17: : Survey Question no.5



6.Are you aware about a device called hardware token?

36 responses

Figure 18: : Survey Question no.6

7.Are you inclined to utilize an additional security authentication device to enhance the security of your company or any field authorization?

36 responses

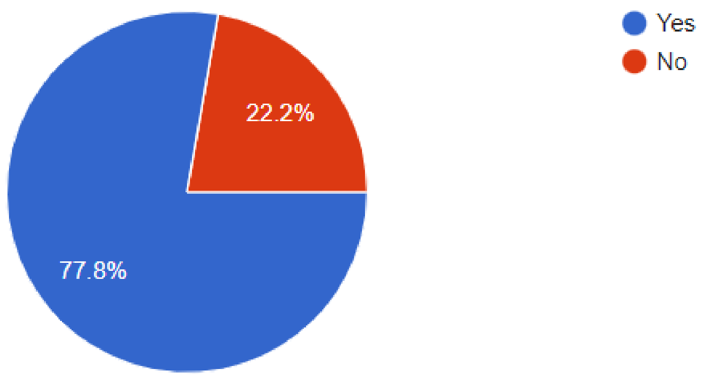


Figure 19: : Survey Question no.7

8.Will you carry a credit/debit size token if it will provide additional security during authorization?

36 responses

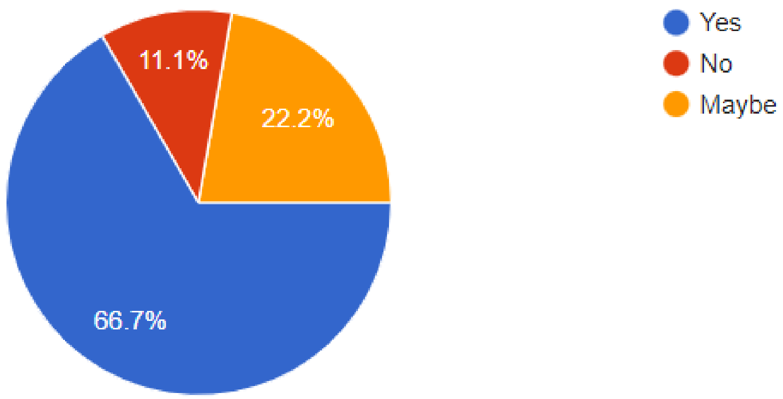


Figure 20: : Survey Question no.8

8.3. Appendix: Milestones Overview

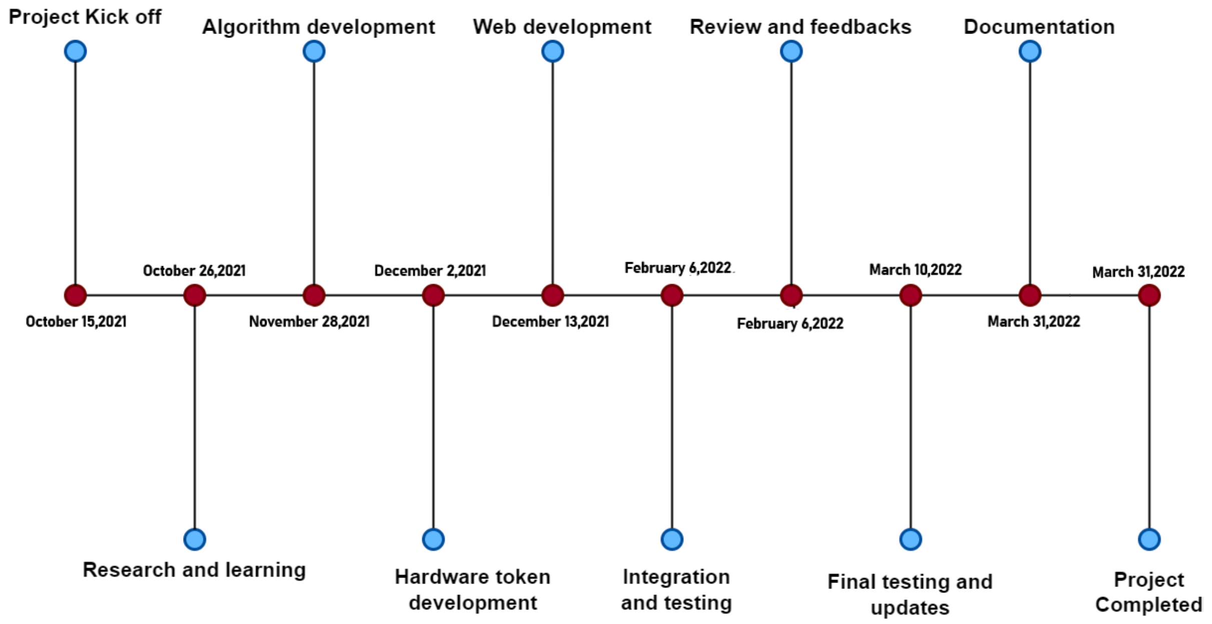
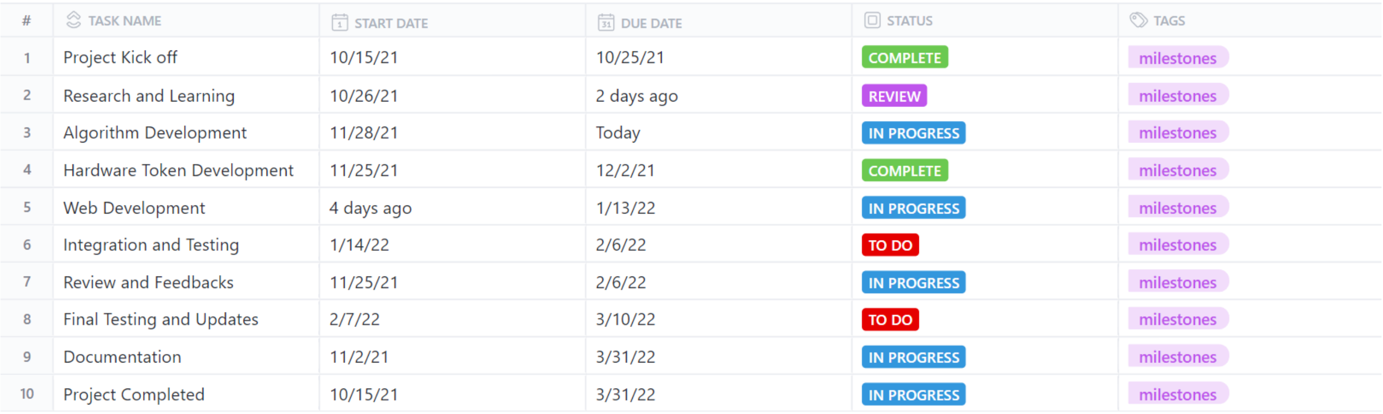


Figure 21: Milestones of the project

Table 3: Milestone Overview Table

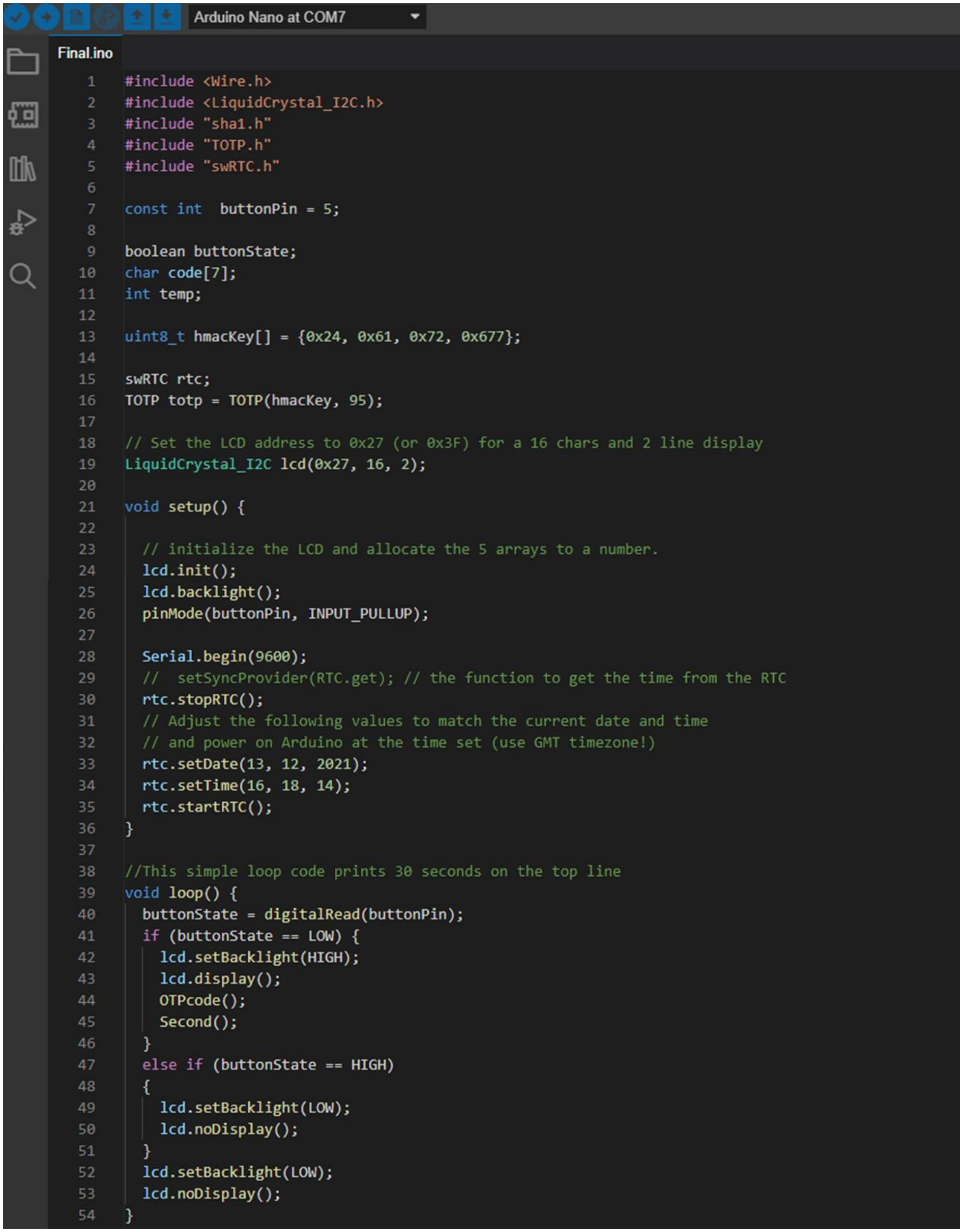


#	TASK NAME	START DATE	DUE DATE	STATUS	TAGS
1	Project Kick off	10/15/21	10/25/21	COMPLETE	milestones
2	Research and Learning	10/26/21	2 days ago	REVIEW	milestones
3	Algorithm Development	11/28/21	Today	IN PROGRESS	milestones
4	Hardware Token Development	11/25/21	12/2/21	COMPLETE	milestones
5	Web Development	4 days ago	1/13/22	IN PROGRESS	milestones
6	Integration and Testing	1/14/22	2/6/22	TO DO	milestones
7	Review and Feedbacks	11/25/21	2/6/22	IN PROGRESS	milestones
8	Final Testing and Updates	2/7/22	3/10/22	TO DO	milestones
9	Documentation	11/2/21	3/31/22	IN PROGRESS	milestones
10	Project Completed	10/15/21	3/31/22	IN PROGRESS	milestones

Figure 22: Milestones Overview Table

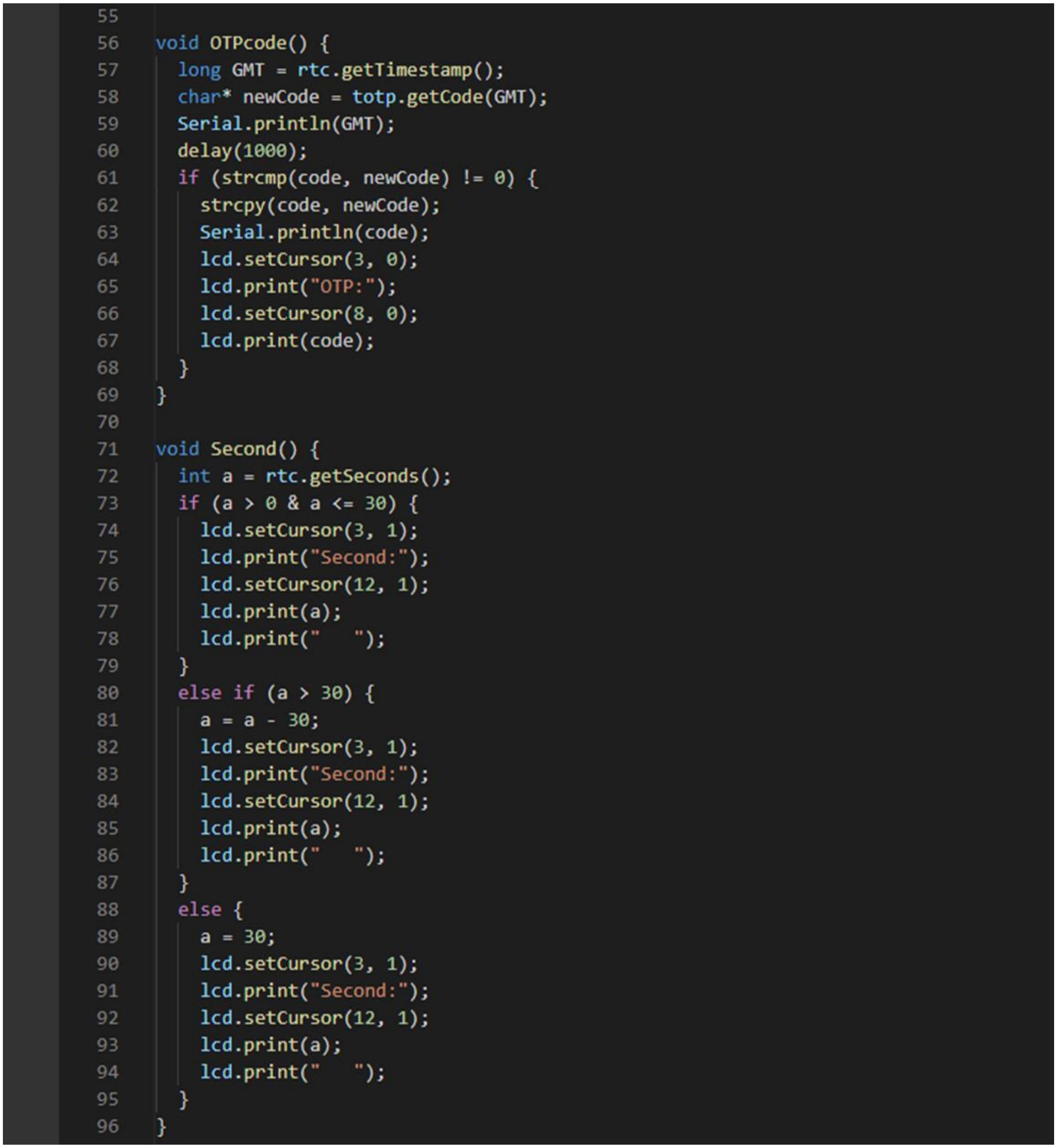
8.4. Appendix 3: Development

8.4.1. Hardware Algorithm



```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "sha1.h"
#include "TOTP.h"
#include "swRTC.h"

const int  buttonPin = 5;

boolean buttonState;
char code[7];
int temp;

uint8_t hmacKey[] = {0x24, 0x61, 0x72, 0x677};

swRTC rtc;
TOTP totp = TOTP(hmacKey, 95);

// Set the LCD address to 0x27 (or 0x3F) for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {

  // initialize the LCD and allocate the 5 arrays to a number.
  lcd.init();
  lcd.backlight();
  pinMode(buttonPin, INPUT_PULLUP);

  Serial.begin(9600);
  //  setSyncProvider(RTC.get); // the function to get the time from the RTC
  rtc.stopRTC();
  // Adjust the following values to match the current date and time
  // and power on Arduino at the time set (use GMT timezone!)
  rtc.setDate(13, 12, 2021);
  rtc.setTime(16, 18, 14);
  rtc.startRTC();
}

//This simple loop code prints 30 seconds on the top line
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == LOW) {
    lcd.setBacklight(HIGH);
    lcd.display();
    OTPcode();
    Second();
  }
  else if (buttonState == HIGH)
  {
    lcd.setBacklight(LOW);
    lcd.noDisplay();
  }
  lcd.setBacklight(LOW);
  lcd.noDisplay();
}
```

Figure 23: Hardware Token Algorithm I



```

void OTPcode() {
  long GMT = rtc.getTimestamp();
  char* newCode = totp.getCode(GMT);
  Serial.println(GMT);
  delay(1000);
  if (strcmp(code, newCode) != 0) {
    strcpy(code, newCode);
    Serial.println(code);
    lcd.setCursor(3, 0);
    lcd.print("OTP:");
    lcd.setCursor(8, 0);
    lcd.print(code);
  }
}

void Second() {
  int a = rtc.getSeconds();
  if (a > 0 & a <= 30) {
    lcd.setCursor(3, 1);
    lcd.print("Second:");
    lcd.setCursor(12, 1);
    lcd.print(a);
    lcd.print("   ");
  }
  else if (a > 30) {
    a = a - 30;
    lcd.setCursor(3, 1);
    lcd.print("Second:");
    lcd.setCursor(12, 1);
    lcd.print(a);
    lcd.print("   ");
  }
  else {
    a = 30;
    lcd.setCursor(3, 1);
    lcd.print("Second:");
    lcd.setCursor(12, 1);
    lcd.print(a);
    lcd.print("   ");
  }
}
```

Figure 24: Hardware Token Algorithm II

8.4.2. Hardware Algorithm Prototype

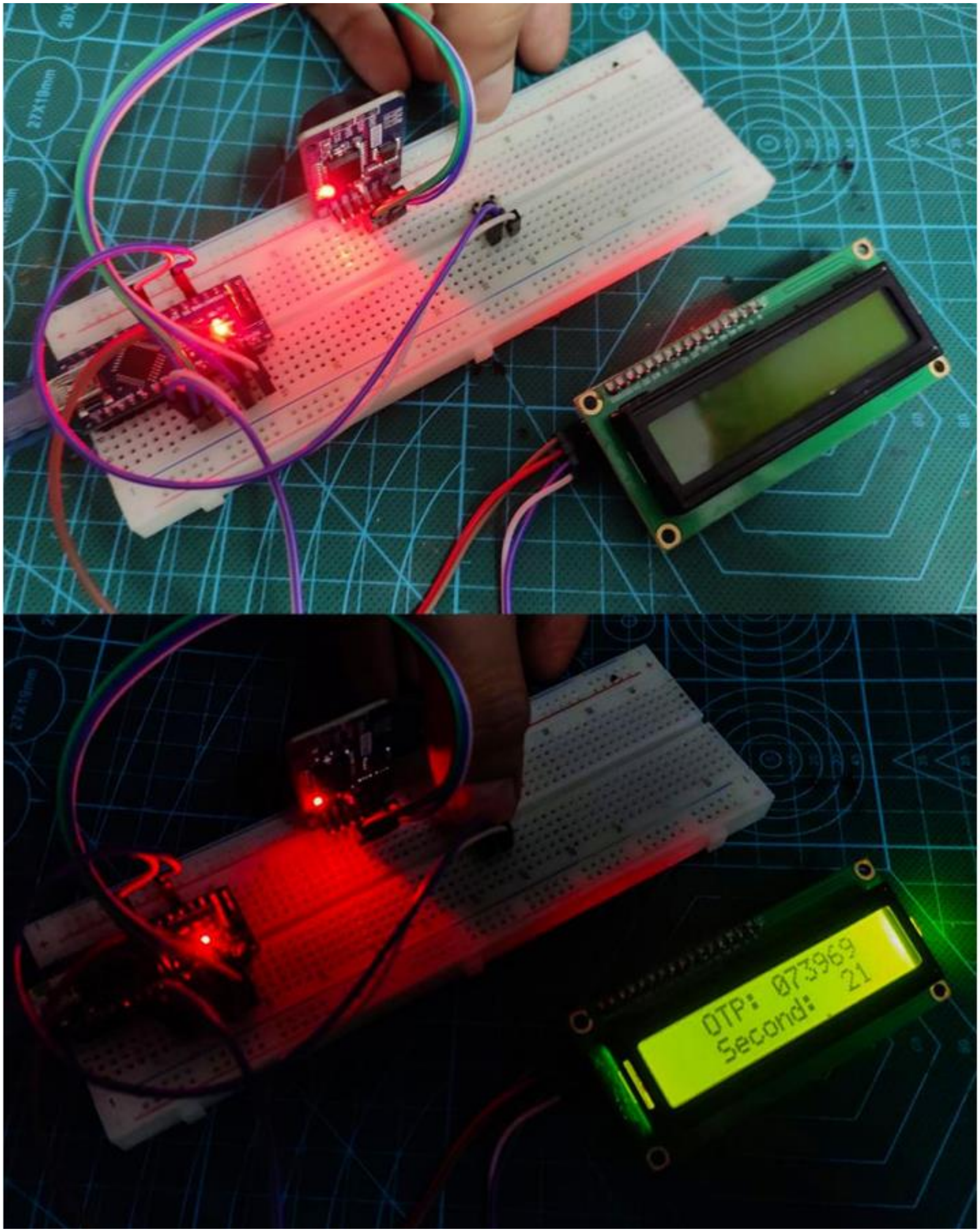
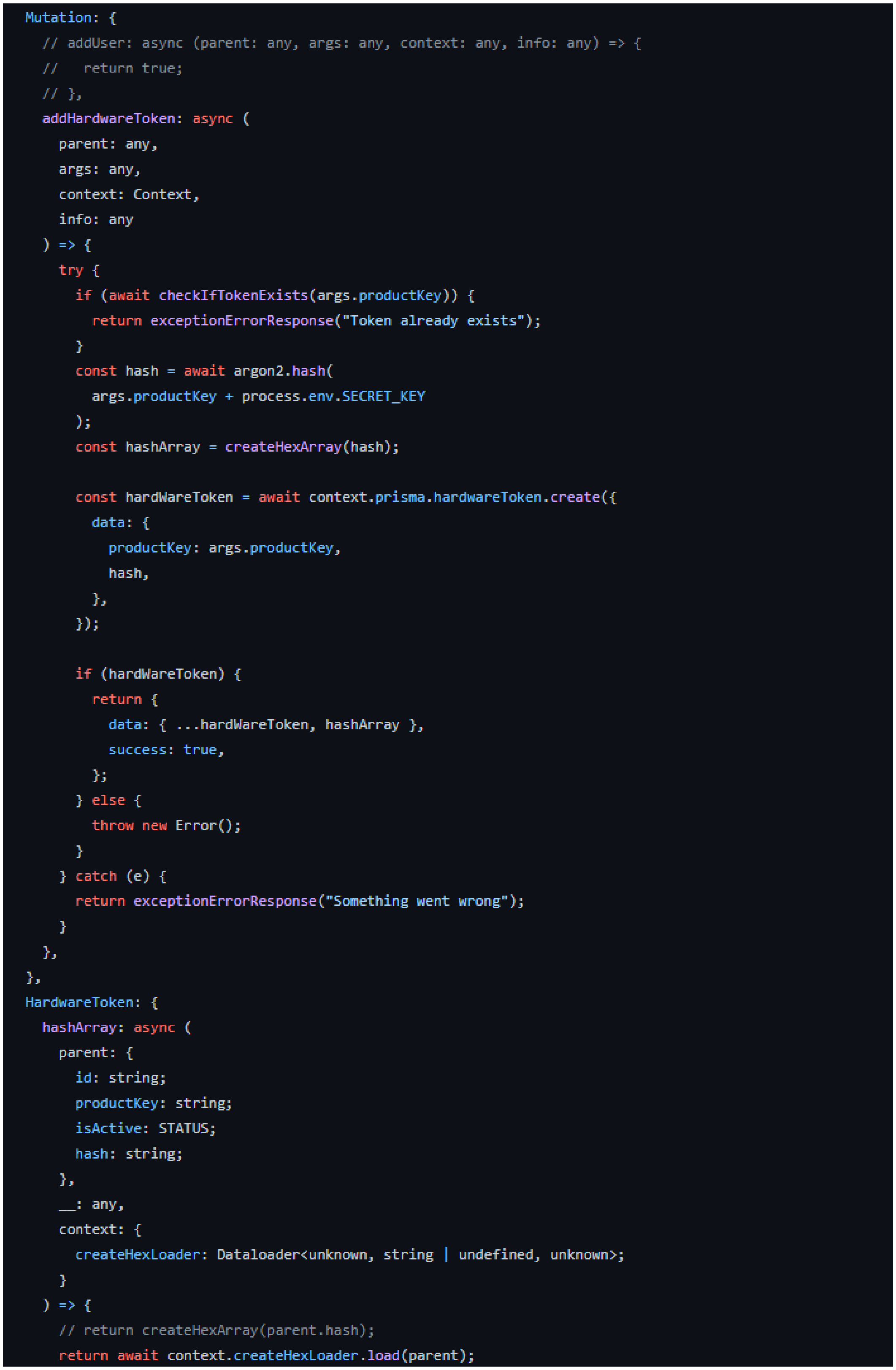


Figure 25: Hardware Token Prototype

8.4.3. Hardware Algorithm



```
Mutation: {
  // addUser: async (parent: any, args: any, context: any, info: any) => {
  //   return true;
  // },
  addHardwareToken: async (
    parent: any,
    args: any,
    context: Context,
    info: any
  ) => {
    try {
      if (await checkIfTokenExists(args.productKey)) {
        return exceptionErrorResponse("Token already exists");
      }
      const hash = await argon2.hash(
        args.productKey + process.env.SECRET_KEY
      );
      const hashArray = createHexArray(hash);

      const hardWareToken = await context.prisma.hardwareToken.create({
        data: {
          productKey: args.productKey,
          hash,
        },
      });

      if (hardWareToken) {
        return {
          data: { ...hardWareToken, hashArray },
          success: true,
        };
      } else {
        throw new Error();
      }
    } catch (e) {
      return exceptionErrorResponse("Something went wrong");
    }
  },
},
HardwareToken: {
  hashArray: async (
    parent: {
      id: string;
      productKey: string;
      isActive: STATUS;
      hash: string;
    },
    __: any,
    context: {
      createHexLoader: Dataloader<unknown, string | undefined, unknown>;
    }
  ) => {
    // return createHexArray(parent.hash);
    return await context.createHexLoader.load(parent);
```

Figure 26: Back-end token generating algorithm

8.4.4. Login UI Design

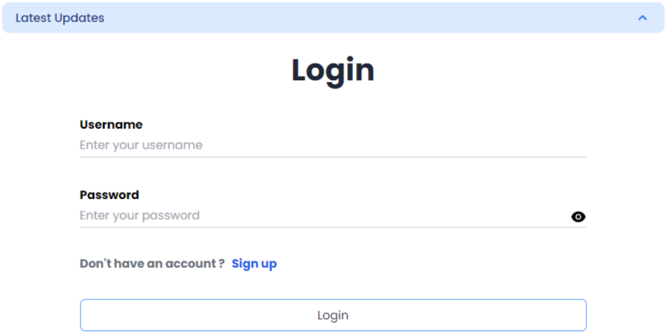


Figure 27: Login UI Design

8.5. Agreement Letter