**Proposal**

Student Name: Carlos R. Ocasio Rodriguez

Degree and Major: Bachelor of Science in Cybersecurity

Project Advisor Name: Prof. Henderson

Expected Graduation Date: May 2022

**Problem Statement**

The management of passwords is one of the most important aspects of Cybersecurity and it is imperative that users implement and practice safe password management practices. In order to ensure safe password management practices users must create strong and unique passwords for every one of their accounts. User should then be able to log into all of their accounts in an efficient and effortless manner. Unfortunately, most users have far too many online accounts to remember a unique password for each account. A survey conducted by Nate Lord (2020) indicates that the average user has over 90 different online accounts. There is a limit to how many passwords a user can remember, and this survey revealed how the majority users struggle with the number of passwords they manage, which usually leads to users reusing the same passwords. A Google Online Security Survey (2019) found that 52 percent of users reuse the same password for multiple accounts. The practice of reusing passwords leaves users vulnerable to intrusion attacks, which can result in data loss, monetary loss and more. A report conducted by IBM (2020) found that the average cost of a data breach is $3.68 million, while a data breach in the healthcare industry is estimated to cost around $7.13million. Furthermore, IBM found that data breaches in the United States are among the costliest averaging around $8.64 million.

By using a password manager users can easily manage unique passwords for their accounts while eliminating all the difficulties that are associated with complex passwords. With a password manager users can store all of their passwords in a secured vault that can be accessed with a “master” password. This eliminates the need to remember multiple passwords and greatly reduces the chances of having multiple accounts compromised. The management of passwords is one of the most important and overlooked factors of information security. In order eliminate the difficulties that come with password management practices it is recommended that user utilize a password management application in order to protect their accounts.

**Project Description**

The project will be a password manager application for Windows computers used to securely store and manage log in information for various accounts. It will use a “master” password to provide access into the user’s account with all of his/her passwords. The application will have connectivity to a database that will be used to securely store the user’s information. In order to protect the information in the database the application will use the well-established AES encryption and SHA256 hashing algorithms. Additionally, the application will be able to generate unique and complex passwords for every account. Users will also be able to enter their own passwords and the application will guide them on how to improve and strengthen the complexity of their passwords. Lastly, the program will have features that allows users to reset their “master” password if they forget it.

**Proposed Implementation Languages**

The password manager application will be developed by using the latest versions of Python (version 3.9.7) and MySQL (version 8).

**Libraries, packages, development kits**

The password manager application will use the following libraries:

* Hashlib
* cryptography.fernet
* msvcrt
* mysql.connector
* os
* sys
* getpass
* random

***Note: This information is tentative and will be updated as the program develops.***

Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |
| --- |
|  |

**Additional Software/Equipment Needed**

A Windows operating system is required, as well as Python (version 3.9.7) and a MySQL (version 8) database. See below for instructions on how to download and setup both Python and MySQL.

**MySQL Installation Guide:**

These steps will guide you through the installation and setup of the MySQL Community Server for the required database of this application.

1. Visit <https://dev.mysql.com/downloads/mysql/>
2. Look for the “MySQL (MSI) Installer for Windows” and click on the “Go to Download Page” button (You can also use this link: <https://dev.mysql.com/downloads/windows/installer/8.0.html>)
3. There will be two Windows MSI Installer options here. Please find the “(mysql-installer-community-8.0.27.0.msi)” option and press the download button.
4. You will be redirected to a screen that asks you to login or sign up, but this is not needed. Scroll down to the bottom of the page and click on the “[No thanks, just start my download.](https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-8.0.27.0.msi)” link.
5. A download prompt will be displayed. Click on the “Save File” button. This will start the download of the file.
6. Wait for the download to finish.
7. Once it has downloaded, look for the “mysql-installer-community-8.0.26.0” (version number might change) file and run it. The file is typically located in your downloads folder.
8. After double clicking the file, it will begin preparing the installation and it will then ask you for permission to install the application. Click the “yes” button.
9. It may ask you “Do you want to allow this app to make changes to your pc?”. Click on the “yes” button. (Skip this step if you did not see this alert)
10. The MySQL installation windows will now be displayed, and it will ask you to read over the License Agreement. After doing so, press the “I accept the license terms” check box and press the “Next” button.
11. On the “Choosing a Setup Type” screen select the “Developer Default” option and press next.
12. A screen may alert you that some product requirements have not been satisfied and it will ask you if you wish to continue. Select yes. (Skip this step if you did not see this alert)
13. On this installation screen you can see all of the products that will be installed. When you are ready select the “Execute” button. This will begin installing all of the products.
14. Wait for the Installation to finish.
15. After the download has finished you will see a green checkmark on all of the products downloaded. If everything downloaded successfully press the “Next” button.
16. On the “Product Configuration” screen you will be able to see the configuration for all of your downloaded products. Once you have reviewed this information press the “Next” button.
17. On the “Type and Networking” screen select the “Standalone MySQL Server/Classic MySQL Replication” option and press the “Next” button.
18. On this screen leave the “Config Type” as “Development Machine”. Make sure that the “TCP/IP” checkbox and “Open Firewall port for network access” checkbox are selected. Lastly, the “Port Number” is 3306. After confirming these options press the “Next” button.
19. On the Accounts and Roles screen enter a strong password on the “MySQL Root Password” fields (**make sure you remember this password**). Then press the “Add User” button to create a user account.
20. A screen will appear and ask you for username information. Enter a username for your account (**make sure you remember this username**). Then enter a strong password for your account (**make sure you remember this password**). Once this is done press the “Ok” button. After creating your user account press on the “Next” button.
21. On the “Windows Service” screen leave all settings as default and press on the “Next” button.
22. On the “Plugins and Extensions” screen leave all settings as default and press on the “Next” button.
23. On the “Apply Configuration” screen you will see an overview of the configurations. Once you are ready press the “Execute” button and the listed changes will be applied. Once this is done press the “Finish” button.
24. On the “Product Configuration” Screen you will see that the MySQL Server is completed. Press the “Next” button to continue.
25. On the “MySQL Router Configuration” screen you will leave all settings as default and press on the “Next” button.
26. On the “Product Configuration” screen you will review the configuration and press on the “Next” button.
27. On the Connect To Server screen, select the “MySQL Server 8.0.27.0 (version may be different for you) text. Then enter press the “Check” button to test the connection. Once the connection is established and you see a “Connection Successful” message press the “Next” button.
28. On the apply configurations, press the “Execute” button. Once all the configurations are finished press the “Finish” button.
29. On the “Product Configuration” screen you will review the configuration and press on the “Next” button.
30. You will see the installation complete screen. Press the “Finish” button.
31. You have successfully installed the MySQL database.

**To use with MySQL Workbench:**

1. Go to start and search for “workbench”
2. Open the MySQL Workbench
3. Once the application opens, double click the “Local instance” below the “MySQL Connections”.
4. A screen will appear, and you will be prompted to enter the password for the database. After entering the password, press the “OK”.
5. This is your database serve.

**Personal Motivation**

The practice of securing networks, systems and other digital infrastructures are an essential aspect to information systems and cybersecurity. As a Cybersecurity major, I strive to broaden these previously mentioned technical and functional skills. The development of this application will provide me with a deeper understanding of various hashing and encryption practices that are vital to the field of cybersecurity. It will also strengthen my problem solving and debugging skills that are necessary as a programmer. The program will also require an understanding of the various methods that attackers used to exploit passwords. This will expand my computer analysis skills and provides me with additional resources and solutions that can be used to thwart malicious attacks. By creating this application and developing the previously mentioned skills I will be able to set myself apart from the competitions and I will be able to demonstrate my expertise as a cybersecurity professional.

**Outline of Future Research Efforts**

The project will be completed by utilizing the previously attained skills during my academic development at CSU. The first phase of the project involves researching and testing of various features. This is important as it will reveal what the requirements for the applicating should be and it will determine what features are not feasible for the scope of this project. The second phase involves identifying requirements and developing the requirements document. At this stage the specifics of the project will be described and a document with the requirements will be drafted. This document will identify the main functions of the program and will establish the scope of the project. After this document is created some research will have to be conducted in order to begin the prototype construction phase. The research will determine how the design of the program should be implemented. The research will involve surveys, observations and studying similar applications. After this is done a prototype will be constructed along with all of its documentation. After conducting some additional research, a high-fidelity prototype will be constructed. The design of the final product will be constructed based on the feed-back from this high-fidelity prototype. The final phase involves testing the functionality and security of the application.

A functional password manager application will be presented along with all of the relevant documented information. The documents include Requirement’s document, Prototype documents, Final Prototype documents, Test cases document.

**Requirements**

Requirement ID Number: 1 Requirement Type: 1, 2 (Functional, Look and Feel)

Description: The application will have a command-line-based user interface.

Rationale: An interface will be needed so that users can interact with the application in an efficient and simple manner. A broader audience will be able to use the application if a command-line-based user interface is implemented.  
Fit Criterion: Users must be able to use all features and accomplish all tasks by using the command-line-based interface.

Priority: Very High  
Dependencies: None

Requirement ID Number: 2 Requirement Type: 1, 6 (Functional, Security)

Description: A login section must be implemented.

Rationale: The login section is needed to authenticate users and grants them access to their data.   
Fit Criterion: When presented with valid credentials the application must authenticate users and grant them access to their data. If presented with invalid credentials, the application will display an error message and will not grant them access to the data. Users will not be able to get past the login screen without valid credentials.

Priority: Very High  
Dependencies: None

Requirement ID Number: 3 Requirement Type: 1, 6 (Functional, Security)

Description: The application must have connectivity to a database.

Rationale: The database provides storage and security features.

Fit Criterion: The application must be able to save data to the database. The data will be lost or inaccessible if there is no database connectivity.

Priority: Very High  
Dependencies: None

Requirement ID Number: 4 Requirement Type: 6 (Security)

Description: The application must store all the data in the database.

Rationale: The database provides storage and security features. It also provides features against data redundancy and reduces inconsistencies. All these features are important for managing log-in credentials.

Fit Criterion: The application must be able to save data to the database. This data must then be accessible to authorized users only. An error message will alert the user if the data was not saved to the database.

Priority: Very High  
Dependencies: The application must have connectivity to the database for this requirement to be fulfilled.

Requirement ID Number: 5 Requirement Type: 1, 6 (Functional, Security)

Description: A “Create New Account” section must be implemented.

Rationale: Users need to be able to create an account to use the application. Therefore, a “create new account” section must be implemented.

Fit Criterion: The “create new account” section must allow users to create a master username and password. If correctly done the username and password will be stored in the database. These credentials can then be used to log in to the application. Users cannot use the application without creating an account.

Priority: Very High  
Dependencies: A “Login” section and the database must be implemented before this requirement can be fulfilled.

Requirement ID Number: 6 Requirement Type: 6 (Security)

Description: Application will lock-out an account after five failed log-in attempts and the password will be reset.

Rationale: Unauthorized users may attempt to guess the login credentials of a user. To prevent these brute force attacks a lock-out feature will be implemented. User’s will need to check their emails to reset their password.

Fit Criterion: After five failed log-in attempts the application will place a lock on the account and the user will not be able to log in until a password reset is completed for the account. The lock-out will be lifted once the password reset has been completed.

Priority: Very High  
Dependencies: The “Login” section and the “Create New Account” section must be implemented before this requirement can be fulfilled.

Requirement ID Number: 7 Requirement Type: 6 (Security)

Description: All data stored and used by the application must only be accessible to authorized users.

Rationale: The data stored and used by the application must only be visible to authorized users. To prevent a data breach, registered users will only have access to their data. Users that are not registered will not be able to use the application and will not have access to any data.

Fit Criterion: Registered users must only be able to see their data, while unregistered users will not be able to use the application.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, and database sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 8 Requirement Type: 6 (Security)

Description: A SHA256 hash will be used as a log-in authentication method.

Rationale: In order to improve security, the username and password along with a hashing algorithm will be used to authenticate the user.

Fit Criterion: SHA256 will be used to hash the username and password provided. If the username and password provided match the login credential stored in the database, the user will be authenticated, and access will be granted. If the hash does not match the login credential stored in the database access to the application will be denied.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, and database sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 9 Requirement Type: 3, 6 (Usability, Security)

Description: The application must be able to generate random passwords.

Rationale: Users may need help creating secure passwords. The application will provide them with the option of generating a strong random password for their desired account.

Fit Criterion: A random password must be generated and displayed to the user after opting for this feature. An error will alert the user if the random password was not generated.

Priority: Medium  
Dependencies: None

Requirement ID Number: 10 Requirement Type: 3, 6 (Usability, Security)

Description: The application will allow users to create their own password.

Rationale: Users may not like having randomly generated passwords. They may want to use their own passwords.

Fit Criterion: Users must be able to enter their own passwords. A confirmation message will alert the user that their password was accepted and saved.

Priority: High  
Dependencies: The “Login”, “Create New Account”, and database sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 11 Requirement Type: 3, 6 (Usability, Security)

Description: A new algorithm will be created in order to detect the strength of a password.

Rationale: Users may want to know if their password is strong enough or if it is weak. This can help them get a better understanding of what a strong password is composed of. It will also increase the overall security of the accounts.

Fit Criterion: A message will alert the user with the strength of their password. Additional suggestions will be displayed to increase the strength of the password. An error will be displayed if the strength cannot be determined.

Priority: High  
Dependencies: The “Login”, and “Create New Account” sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 12 Requirement Type: 3 (Usability)

Description: Simple controls must be implemented on the front end to send SQL queries to the database.

Rationale: Users do not want to create SQL queries for every task while using the application.

Fit Criterion: Queries will be created in the back end and the users must be able to utilize them on the front end. Users must be able to view, add, edit, and delete records on the database from the front end. A confirmation message will be displayed after a successful query.

Priority: High  
Dependencies: The “Login”, and “Create New Account” sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 13 Requirement Type: 3 (Usability)

Description: User data stored in the database will be encrypted by using fernet (AES) encryption.

Rationale: Encrypting the data prevents unauthorized users from using it. Encryption increases the security of the application and protects the data of the users.

Fit Criterion: No plaintext must be visible in the database. If plaint text is found in the database this requirement has not been fulfilled. The information in the database must be decrypted before it can be used. If an encryption attempt fails an error message will be displayed to the screen.

Priority: Very High  
Dependencies: The database must be implemented before this requirement can be fulfilled.

Requirement ID Number: 14 Requirement Type: 3, 6 (Usability, Security)

Description: Registered users must be able to retrieve their unencrypted data from the database.

Rationale: Users need to be able to see their saved login credentials. They may want to view them before changing or deleting the credential.

Fit Criterion: Users must be able to select a specific account to retrieve their unencrypted password. If the data retrieved data is encrypted this requirement has not been fulfilled. If the data cannot be retrieved or if it cannot be unencrypted an error message will be displayed to the user.

Priority: High  
Dependencies: The database and the encryption algorithm must be implemented before this requirement can be fulfilled.

Requirement ID Number: 15 Requirement Type: 3 (Usability)

Description: Passwords retrieved from the database must be copied to the user’s clipboard.

Rationale: This is done so that the user can immediately use the password as it is automatically placed in their clipboard. After retrieving the password users can paste it into their desired field.

Fit Criterion: After retrieving the data the user must be able to use the “paste” command and the password will be “pasted” into their desired field. An error message will be displayed if the password was not placed in the clipboard. On the other hand, a confirmation message will be displayed if the password was placed into the user’s clipboard.

Priority: Low  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 16 Requirement Type: 3 (Usability)

Description: The application must guide and direct users on how to accomplish all tasks.

Rationale: Users may not understand how to use the application. Clear and concise instructions will help the user navigate the application.

Fit Criterion: Prompts and alerts will guide the user on how to accomplish their task. The success of the direction will be based on task completion time. The average task completion time should be under five minutes.

Priority: Medium  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 17 Requirement Type: 3 (Usability)

Description: The users must be able to go back to the previous step at any point of the process.

Rational: The users may decide they wants to change a specific thing on a previous screen and may not want to start over the whole process.

Fit Criterion: A back option must be present in every part of the application. Once the option has been selected an alert message will prompt the user to confirm the action. The user will be taken back to the previous screen if they confirm the action.

Priority: Medium  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 18 Requirement Type: 6 (Security)

Description: The users must be able to log out of the application at any point of the process.

Rational: The users need to be able to log out of the application at any point of the process for security reasons. This may prevent unauthorized users from gaining access to their account.

Fit Criterion: Once the option has been selected an alert message will prompt the user to confirm the action. The user will be logged out of the application if they confirm the action.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 19 Requirement Type: 3 (Usability)

Description: The users must be able to create, view, edit, and delete records from their account.

Rational: Users need to have full control over their data. They may want to create new records or delete old ones.

Fit Criterion: A confirmation message will be displayed if the action selected was completed successfully. On the other hand, if one of these actions cannot be completed an error message will be displayed to the user.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 20 Requirement Type: 6 (Security)

Description: Passwords that are considered too weak will not be accepted.

Rational: Short passwords or passwords that are considered too weak will not be accepted by the application as this is a security risk.

Fit Criterion: A message will alert the user if their password is too weak. They will also not be able to save their password and will be asked to come up with a new one. Once a strong enough password has been entered a confirmation message will be displayed to the screen.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, password strength detector, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 21 Requirement Type: 6 (Security)

Description: Input validation must be performed throughout every section of the application.

Rational: Users may enter invalid data accidentally or purposefully and thus every input must be properly validated and sanitized to prevent a security risk. All prompts will be validated and will include exceptions to handle all scenarios.

Fit Criterion: All invalid inputs will produce an error message that will be displayed to the screen.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, password strength detector, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 22 Requirement Type: 3 (Usability)

Description: Error messages must provide guidance on how to complete the task.

Rational: Users may not understand why they can’t proceed and may need help fixing their mistakes.

Fit Criterion: The error messages must provide clear and concise instruction on how to complete the current task. The success of the direction will be based on task completion time. The average task completion time should be under ten minutes.

Priority: Low  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 23 Requirement Type: 6 (Security)

Description: The application must have measure in place to prevent SQL injection attacks.

Rational: Unauthorized users may attempt to use an SQL injections to get access to the data.

Fit Criterion: Inserting SQL injections will result in an invalid input error.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 24 Requirement Type: 6 (Security)

Description: The average user should be able to retrieve a password in under three minutes.

Rational: Users typically do not want to use application that take to long to complete a task.

Fit Criterion: Users must be able to retrieve a password in under three minutes.

Priority: Medium  
Dependencies: The “Login”, “Create New Account, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 25 Requirement Type: 6 (Security)

Description: The application must have emailing capabilities in order to reset passwords.

Rational: The application may need to send emails to the users. To reset passwords, the application will require emailing capabilities.

Fit Criterion: Application must be able to send emails to the users. Once the email has been sent a confirmation message will be displayed to the screen.

Priority: Medium  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 26 Requirement Type: 6 (Security)

Description: Security questions stored in the database can be used to reset the master password.

Rational: Users may forget their password and may need additional ways of resetting their passwords other than email.

Fit Criterion: Answering the three security questions correctly will allow the user to reset their password. Answering one of the questions incorrectly will place the account in a lock-out state.

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, database, emailing and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 27 Requirement Type: 6 (Security)

Description: Exiting the application in the middle of data entry will display a warning and no data will be saved to the database.

Rational: User’s may need to exit the application in the middle of data input. This requirement will prevent the application from saving invalid or incomplete data into the database. The warning will inform the user of the data lost.

Fit Criterion: An alert message will be displayed if the user tries to close the application in the middle of data input. The user will have to acknowledge the alert before exiting the application.

Priority: High  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

Requirement ID Number: 27 Requirement Type: 6 (Security)

Description: Clipboard will be cleared after XXXXX seconds.

Rational:

Fit Criterion:

Priority: Very High  
Dependencies: The “Login”, “Create New Account”, database and encryption sections must be implemented before this requirement can be fulfilled.

**Additional Requirements:**

**Securing Database (Security)**

In order to secure the database and the information within it, access into the database will be limited and will require a password to authenticate. Ideally only the root user will have control and access to the database. Other accounts will need well defined rules for database access, Views access, and data storing procedures based on needs and the role of the individuals. To protect the information within the database, all the data will be encrypted before being stored in the database. In order to prevent unauthorized connections into the database, access will be limited to local host. Additionally, in order to prevent SQL injections, all queries will be defined within the application, and then the parameters will be passed on to the query after being sanitized. This will validate the user input and prevent users from creating their own queries.

**Clipboard Vulnerability (Security)**

All password manager applications utilize the Windows clipboard to allow users to copy and paste a specified account password. This is a convenient feature as it prevents users from having to type long and complicated passwords, but it can also be exploited if the computer was infected with malware. When the password is copied into the clipboard this data is temporarily stored in the memory of the computer. The longer the password kept in the clipboard the longer it will be stored in memory. This creates a window of opportunity for an attacker to access your system and retrieve the password from the clipboard or the memory of the computer. To prevent this from happening password managers flush the clipboard after a certain amount of time has passed. A feature like this will be created and implemented in this password manager. The application will alert the user that the password will be wiped from the clipboard after thirty seconds of retrieving the password. The user may retrieve the password again if he needs it after the first thirty seconds. In addition to clearing the clipboard after thirty seconds, it will also be cleared when the user performs any other task in the application, when the user logs out, and lastly when the user exits the application. This feature will not completely prevent this vulnerability, but it will greatly reduce the probability of it happening and it is the industry standard.

**Application User Interface (Appearance)**















































**Proposed Schedule**

1. First Week (Sept 20 – Sept 26): During this week the research for the Requirements document will be conducted.
2. Second Week (Sept 27 – Oct 3): During this week a draft of the Requirements document will be submitted, and a final version of the document will be created based on the feedback.
3. Third Week (Oct 4 – Oct 10): During this week the Requirements document will be completed and submitted to the advisor. Lastly, research will be conducted for the prototype design of the application. The process of creating the front end of the application will commence and this will conclude the CSCI 497 course.
4. Fourth Week (Oct 11 – Oct 17) The front end of the application will be completed this week.
5. Fifth Week (Oct 18 – Oct 24) The database will be created and linked with the program during this week.
6. Sixth Week (Oct 25 – Oct 31) The database work will be completed this week and the encryption section of the project will commence.
7. Seventh Week (Nov 1 – Nov 7) The encryption algorithm will be completed this week and the hashing and authentication work will commence.
8. Eight Week (Nov 8 – Nov 14) The Hashing and Authentication functions will be completed this week. Any other remaining features will be completed this week.
9. Ninth Week (Nov 15 – Nov 21) Input validation and testing will commence this week. SQL injection prevention methods will be implemented this week.
10. Tenth Week (Nov 22 – Nov 28) This week will be used to test the application.
11. Eleventh Week (Nov 29 – Dec 6) Last minute changes and testing will be performed this week.
12. Twelfth Week (Dec 7 – Dec 13) Final’s week

***Note: This is a rough schedule, and it will be updated accordingly after receiving additional guidance.***

**References**

Google. (2019, February). *Online Security Survey*. Google. Retrieved September 18, 2021, from https://services.google.com/fh/files/blogs/google\_security\_infographic.pdf.

Greig, J. (2020, July 29). *IBM finds cyberattacks costing companies nearly $4 million per breach*. TechRepublic. Retrieved September 18, 2021, from https://www.techrepublic.com/article/ibm-finds-cyberattacks-costing-companies-nearly-4-million-per-breach/.

Lord, N. (2020, September 29). *Uncovering password habits: Are users' password security habits improving? (infographic)*. Digital Guardian. Retrieved September 18, 2021, from https://digitalguardian.com/blog/uncovering-password-habits-are-users-password-security-habits-improving-infographic.