The project was started to create a system for banking, protecting user data and allowing users to easily access the site. The project highlights the process and demanding requirements of cybersecurity, and the efforts done to create these systems. This project highlights the importance of cybersecurity when it comes to banking and sensitive user data. The system was primarily created to protect user information while allowing users to do regular banking activity. The system needed to protect banking information and had to accommodate at least 50,000 people. Additionally, the system needs to be under constant monitoring and needs to connect customers to the site to make bank transactions such as checking account balances, opening/closing accounts, and making deposits and withdrawals. The system required customers to use detailed passwords (including minimum 8 characters, using upper and lower case letters and special symbols).

The project was done using python. Python was used to create basic requirements for the system. Github was used for version management, and most of the system was created through Visual Studio Code for coding and debugging. First, we had a list of things that we wanted the system to achieve. First, we listed down the creation of the system requirements. After defining the requirements, we began working towards the implementation process. We began by creating a password policy. The password was to have a minimum of 8 characters both upper and lower case, with symbols. This was the first part of the coding process. We implement secure password requirements using python. Next, we provide a python script that offers secure user login sessions, at the same time keeping track of their activity. Also, python scripts to track the users for their logging and how many attempts to reach success it would take for them. Using python, we also created functions that users could create checking account balances, making deposits or withdrawing from their accounts and when users would open and close new accounts.

The biggest challenge encountered in this project was ensuring the system could handle large-scale user activity-50,000 concurrent users without performance degradation. Loading the testing revealed bottlenecks in database queries and transition processing. The security vulnerabilities related to the transmission of the data and transactions in the processing area were detected during the testing, The future of iteration must address the issues to prevent the potential issues and the data including the system and others, etc. Security analysis identified as the few critical areas for improvement. Although the password policy was strong, the system could still be vulnerable to brute force attacks and phishing. Multi-factor authentication MFA would significantly reduce the risks of unauthorized access and add an extra layer of protection beyond passwords. That shows that real-time fraud detection mechanisms such as analyzing transactions and patterns and for unusual activity could further enhance security and prevent that from causing more troubleshooting and potential threats.

Our recommendations for the system would be to include a two-factor authentication, since the information is sensitive and security is important. Also, we would include encryption for the sensitive data, and create an outline or section that educates the users about the importance of online safety. Allocate a portion of the system where it would tell the users about the dangers of exposing sensitive information and how it would affect them. Another recommendation would be making the system have regular updates and maintenance. Making the website go through maintenance would improve the user experience and avoid any future issues.

In the end, the project matched its objective of creating a system for users to access a safe banking system. We included a monitoring system for logging user activity, making note of suspicious activity. Whenever it occurs. The system was based in python, we tried to make it as secure as possible. Some challenges faced were making the system able to handle a large number of users. It needed to run a lot of users without disrupting the actual system itself.

The system has various tasks it needs to complete. Such as making sure passwords matched the requirement and wouldn't be too simple. The system also needed to be continuously monitored.It can handle high traffic when the site has a lot of traffic and user activities by creating a logging system. Solution for these tasks would involve improving python use, especially when it comes to using multithreading in Python to further scale the system.The implementation of a detailed password and its strength checks is another solution.

In conclusion, this project created a secure and scalable banking system that met the requirements of protecting sensitive user data while providing a robust platform for the banking activities and the resources. The system was built within the strong security features which are the passwords policies and the secure login and session, and users activities that monitor. Despite the facing challenges related to the scalability and performances under high traffic and solutions such as load balancing and multithreading and efficient database management which were implemented.

Future improvement should focus on strengthening the system with multi factor authentication, real time fraud detentions and enhancing the encryption and protocols, within that it shows that these are the future features which are very important and that needs to be focusing on multi factores and system. Additionally, the ongoing maintenance of regular updates and users' education will ensure the system remains resilient against the emerging threats.

To further explain, the compliance of the regulatory standards such as the GDPR and the PCI and the DSS has to be maintained and the system should continue to adhere to the best of best practices in cybersecurity . In addition to that the security improvements and users experiences and accessibility should be priority to ensure the platform is stable and accessible to a wide range of the users.