Part A: Planning

The Scenario

My client, Oliver Szegedi, is an internet user who uses many different web services requiring separate accounts and logins. He has trouble remembering different passwords for each account, so he has been using the same password for most websites. However, he has recently become worried that this could lead to all his accounts being compromised. He wants to increase security on his accounts, but he doesn't want to remember many different passwords. Therefore, he wants a way to store the different passwords. However, he doesn't want to use an already existing password manager as he does not trust them to keep his passwords secure. Therefore, he asked me for advice. I suggested that I make him a password manager, to which he enthusiastically agreed.

I met with the client and consulted with him (a transcript is available in Appendix 1). We discussed what specific features he would like to see in the password manager. He requested an easy-to-use interface, a random password generator, and the ability to use it on multiple devices, among other things. We decided on a full list of tasks that the product should be able to do, which is shown in the Success Criteria below.

My Computer Science teacher agreed to be my advisor.

Rationale for proposed product

My client has considered writing the passwords down on paper, but this would not be an ideal solution as he would have to type in the password each time, rather than copy-paste it from an application. Furthermore, he could easily lose the paper (since he would have to carry it around to be able to log in away from home on his laptop), or have it stolen. Thus, a complete solution to his problem would be an application that stores his passwords securely: a password manager.

After discussion with my advisor, I decided to develop the product as a single-page application (SPA) that runs in the browser, primarily using Javascript, HTML and CSS. The client requested that the application can run on different computers, and an SPA can easily run on any platform. An SPA improves performance as most elements only need to be loaded once. Moreover, I am already familiar with Javascript, which will make the programming process easier.

I decided to use the Bootstrap framework to improve the design of the application. In addition, Firebase will be used to store the encrypted passwords in the cloud, so that the client can access them from any device. To ensure security, only encrypted data will be transferred to the cloud.

Word Count: 422

Success Criteria

1. The program stores different passwords for different websites under AES-256 encryption
2. The client finds the program easy-to-use and user-friendly
3. The client can add new passwords for different websites

4. The client can delete passwords, but is prompted to confirm deleting so that a password doesn’t get deleted by accident
5. The client can edit passwords for already added websites
6. The client can change the master password
7. Each password has the date that it was last changed displayed next to it, so that the client knows when he should change it
8. The program can be used across multiple devices, and the passwords saved on one device will show up on the other
9. The client can create an account with an email and passphrase
10. The program has an inbuilt password generator