Criterion E: Evaluation

Evaluation of the product

1. The program stores different passwords for different websites under AES-256 encryption
    - ✓ Met. The program uses the AES encryption-JS (Ricmoo, 2018) and Js-scrypt (Garnock-Jones, 2016) libraries to implement AES-256 encryption. The passwords for websites are only stored in Firebase in an encrypted form, thereby ensuring their security.
2. The client finds the program easy-to-use and user-friendly
    - ✓ Met. The client confirmed that the program was easy-to-use (Appendix 2). There are no unnecessary elements such as menu bars, and the application does only the one thing it is supposed to do. Visual icons, popups and tooltips with helpful information complement the application.
3. The client can add new passwords for different websites
    - ✓ Met. In the bottommost row of the password table, there are fields where the client can enter a new password for a new website, and by clicking the plus icon, this password gets added and saved.
4. The client can delete passwords, but is prompted to confirm deleting so that a password doesn't get deleted by accident
    - ✓ Met. When the user clicks on the bin icon, a modal opens asking the user whether he is sure he wants to delete the password.
5. The client can edit passwords for already added websites
    - ✓ Met. By clicking the edit icon, the password field becomes editable, and a save button appears.
6. The client can change the master password
    - ✓ Met. There is a dedicated button which opens a modal with this functionality.
7. Each password has the date that it was last changed displayed next to it
    - ✓ Met. Whenever a password is added or amended, the date that it was last changed will be saved, and will always be displayed next to the password.
8. The program can be used across multiple devices, and the passwords saved on one device will show up on the other
    - ✓ Met. By saving the encrypted passwords in Firebase (i.e. in the cloud), these can be accessed from anywhere.
9. The client can create an account with an email and passphrase
    - ✓ Met. The client creates an account when first starting to use the application, by clicking on the *Create an account* button.
10. The program has an inbuilt password generator
    - ✓ Met. There is a button which when clicked will generate a random string of letters, numbers, and special characters, for the password.

The product works as intended, with all the success criteria being met. The client is very satisfied with the final product (Appendix 2). He was especially impressed with the graphical icons used for the user interface, finding them both intuitive and visually pleasing.

Recommendations for further development

Though immensely satisfied, after using the product for a week, the client had several suggestions for further improvement. He found the interface a bit bland, and would like to be able to customize the background of the main table, as well as the opening screen. Furthermore, there is no information about when he last changed the master password, which he thinks would be helpful to see. Moreover, he has noticed that if using a very long passwords of around 25 characters, its end is not properly hidden from view by the asterisks that normally hide a password unless the view icon is hovered over (Appendix 2).

The advisor suggested to add two-factor authentication, to make the application even more secure.

Word Count: 554

References

Appendix 2

Garnock-Jones, T. (2016). *Js-scrypt: Pure-Javascript Emscripten-compiled scrypt routine*. Retrieved from https://github.io/tonyg/js-scrypt/

Ricmoo. (2018). *AES-JS*. Retrieved from https://github.com/ricmoo/aes-js