Criterion E: Evaluation

1. Evaluation of the product

After delivering the program to the client, I scheduled a meeting to discuss the Success Criteria and gather feedback on possible improvements (outlined in Appendix B).

Success Criteria	Met?	Summary/feedback
1. Add new and update existing students	Yes	In the current version, the user cannot change a student's class. Nevertheless, the client changes primarily portfolios' URLs so that wasn't a problem. "*This worked*".
2. Add new and update existing classes	Yes	No complaints from the client. "*Yes, is does work*"
3. Open one or more students' portfolios	Yes	Works, however, choosing multiple students could be more straightforward without holding "Ctrl" key – because the client "*sometimes confuse*"
4. Choose either descending or ascending alphabetical ordering of students that are displayed	Yes	Works properly. Client described as "*very helpful*"
5. Sort students by either surname or first name	Yes	Works properly. No complains from the client.
6. Restrict student are displayed to a chosen class	Yes	Works properly. Helpful as client doesn't like having "*all students all together*"
7. Search for students by their surname case-insensitively and without the need to type the whole surname	Yes	No complaints from the client. "*Yes, it does work*"
8. Delete individual and multiple students'	Yes	Deleting works, although the client "*do(es) not wish*" to delete any entire class (see 2.2)

records as well as entire classes		
9. Make the app resizable	Yes	Works, however, the client doesn’t not use it extensively because now she "*can open multiple portfolios at the same time*"
10. Enable the change of theme – either black or white	Yes	Works. The client uses the dark theme, but admits that white theme "*may be helpful*" in the summer
11. Validate data before insertion	Yes	No complaints from the client. "*Yeah, it does work*"

2. Recommendations for the future developments of the product

2.1 Add multiple students automatically

The client highlighted that adding multiple students was "*quite a challenge*" due to the required waiting time for URL validation. To expedite this process, a new mode could be implemented to automatically fetch data from an Excel spreadsheet. This mode would add students, generate a report detailing successful additions, and flag any failures.

2.2 Separate current classes from alumni classes

The client expressed that she prefers not to delete alumni classes as their portfolios serve "as examples for younger students." However, the growing number of classes over time could potentially slow down the lookup process. To solve this problem, current and alumni classes could be separated, each with distinct functionalities (as alumni classes are less likely to change). To implement this, a switcher is needed to toggle between the views of current and alumni classes.

2.3 Fetch data while updating

The client mentioned that she had "typed the surname incorrectly once or twice while changing the portfolio" because she needed to re-type all data. To reduce the likelihood of such mistakes, it is suggested to implement the functionality to fetch the existing data of a student to be updated so that the user only needs to correct the necessary fields.

Word count: 338