Report: Tech Basics II

Inhaltsverzeichnis

Introduction

Creating the "Queer Security International" GUI application was driven by a vision to establish a leading digital platform that aggregates and shares essential insights about the safety and inclusiveness of nations for the LGBTQ+ community. Developed using Python and Tkinter, my goal was to foster a collaborative environment where users could contribute to and access a shared pool of knowledge regarding queer security globally. This initiative is close to my heart, aiming to provide the LGBTQ+ community with critical information on global queer safety and inclusivity.

Methodology

My approach to developing this application was iterative, relying heavily on feedback from users (I asked friends to use the gui and tell me what could be improved) and my growing understanding of what the community needs. I looked to combine practical coding solutions, insights from Stack Overflow, and collaborative problem-solving strategies we discussed in class to navigate through the challenges encountered.

Design Enhancements

In my quest to enhance user experience, I introduced several design modifications in the latest version of the application. A significant improvement was the addition of a third page designed to offer detailed insights into individual countries. This page showcases a map background of the selected country, shifting the focus from continent-wide ratings to more nuanced country-specific evaluations. This design choice acknowledges the varied experiences within the LGBTQ+ community across different nations, enabling the storage of unique average ratings throughout a user's session.
The application's database has also been expanded to include a wider array of countries, along with more comprehensive information on LGBTQ+ rights. This expansion not only widens the scope of accessible information but also deepens the collective understanding of queer rights internationally.

New Functionalities

Beyond the text review functionality, I introduced several new features to enrich the application further. The review functionality itself was a major milestone, allowing users to submit and view textual reviews on individual countries, thereby sharing their experiences or insights related to LGBTQ+ rights. This feature has significantly contributed to fostering a sense of community and enhancing the information repository with diverse perspectives on each country.

Challenges and Solutions

Adding new functionalities to the existing application is usually very challenging, full of difficulties, and often easier said than done: this task is usually full of fresh and often brave solutions. As I continued to add new functionality to my application, I came across quite a number of challenges that greatly demanded a sense of grit and creativity from me.

Coming up with a rating feature that allows for submission at the same time was also one of the key challenges, allowing the submission of different ratings for the different countries held in the database. I did this using a dictionary, country_reviews, as the key data structure. In doing so, I designed this dictionary to be one that holds lists of rating entries associated with a country, thus allowing the submitted reviews to be stored persistently and without repetition. The solution was motivated by combining some valuable input from Stack Overflow and hints greatly from the classroom code (line 10-11, Class code).

As the project moved forward, the complexity of the code base rose quite a bit, becoming disorganized and chaotic. This was majorly attributed to my starting approach, where every time I added a new page to the application, I would create another function. Seeing how the codebase could use some better organization and be written more cleanly, I went ahead and utilized the function add_new_page we had learned. This means it was a lot easier to add new pages with far less redundancy, hence making the code much more readable.

Creating country-specific pages without cluttering the code presented another substantial challenge. This was one more place where the add_new_page function came in handy in making adding these pages clean and tidy. That, in turn, not only solves the problem of complexity but also improves the project structurally and maintainably.

However, the biggest challenge arose when adding the text review functionality. This called for a fitting design to ensure that users will be able to submit and view the text reviews. I experienced that it has been quite difficult to have the user input captured and displayed correctly and then integrated within the reviews GUI framework already in place. The sought solutions were looked into using thorough searches in Stack Overflow, and coding insights or strategies consulted ChatGPT (line 22-23, ChatGPT). The .get("1.0", "end-1c") method call specifies to get all the text from the first character ("1.0") to the last character minus one ("end-1c"), effectively fetching the entire review content. This is handy in the complexity of trying to roll out a review system that is dynamic and user-friendly.

Limitations and Reflections

While I have made significant strides in enhancing the application, challenges remain, especially in capturing and displaying user inputs effectively within the text review system. Addressing the complexities of integrating a user-friendly review interface required extensive research and consultation with resources like Stack Overflow and ChatGPT.

Conclusion

My journey in developing the "Queer Security International" GUI application has been both challenging and rewarding. It underscores the importance of community-driven development in creating meaningful resources for the LGBTQ+ community. Through an iterative approach, embracing feedback, and a dedication to improving the user experience, I continue to refine the application. My aim is to make it an indispensable tool for advocating global queer safety and inclusivity, a testament to the power of technology in driving social change.