PuzzlePortal: An Interactive Application for Puzzle Fans

Wintersemester 2024/25
Leuphana University Germany
Tech Basics 2
Lecturer: Sarah Haq
Submitted by Helena Zühlke
Handed in on: 28.02.2025

Inhaltsverzeichnis

Introduction

In Tech Basics 1, I learned how to code in Python and built my own application, *PuzzlePortal*. Over the last semester, I worked on improving my application's usability and design and implemented it into "Streamlit". Now, the app is web-based, which makes it more user-friendly and modern. The purpose of *PuzzlePortal* is still the same: to give puzzle fans a place to share recommendations, discover new puzzles and find inspiration for new hobbies. With the new version, users can not only enjoy a cleaner design and choose different themes but also receive better puzzle recommendations, write reviews, and read information about the puzzling history and the person behind the idea.

Methodology

The biggest change in this version of *PuzzlePortal* is the switch from *Tkinter* to *Streamlit*. Many functions are similar to what I did with *Tkinter*, such as if-elif-else statements and the definition of pages or lists. However, *Streamlit* made it much easier for me to implement additional features, such as containers, customizable color themes the user can choose from or the ability to display images from online URLs. I also used *Pandas* to store and organize user data, while *RapidFuzz* and *FuzzyWuzzy* helped me improve puzzle recommendations by comparing words and finding the best match. These improvements make the app more user-friendly, flexible, and well-structured. Lastly, the toast function allowed me to add small pop-up messages, making the website feel more interactive.

Design

Since I switched to *Streamlit*, the design of *PuzzlePortal* has changed a lot. Instead of a GUI, it is now a web-based application. One of the biggest improvements is the introduction of color themes, which not only update the background and banner but also adjust the buttons, sidebar, and text. The navigation is now easier, as the pages are better organized within the sidebar. The texts and layouts have also been improved, especially through the usage of containers. All buttons, text fields, and images are clearer, which creates a better overall user experience.

Limitations

Even though I made a lot of improvements, some limitations still exist. There is still no option for users to chat with each other in real time. Also, the puzzle recommendations use simple word matching instead of advanced AI, which limits their accuracy.

User Feedback

For a variety of user Feedback, I tested the app with friends and family members and the feedback was very positive:

- **User Friendliness:** The testers agreed that the app was easy to use and navigate through. Even family members who less familiar with electronic devices, found the navigation and interface intuitive. Especially the columns on the inspiration page, which were marked with thin borders, were very useful for a better overview.
- **Preference Page:** The users said they enjoyed the preference page because it was fun clicking through the questions and seeing how well the preferences matched the suggested puzzle. Some told me they played around with the feature to get a 100% match, and some tried to get the lowest matching percentage possible.
- **Problems:** During the first test trial, some users reported a few bugs, such as image display errors or the app running slowly due to something running in the background. However, after I adjusted a few things, everything worked smoothly, and no one reported any bugs.
- **Broader Engagement:** Because I tested the app with a diverse group, including people who don't regularly do puzzles, some suggested adding more interactive features to make the website engaging for a broader audience.

Conclusion

Since I have started coding about a year ago, I have made huge progress. In the beginning I was very skeptical and would have never imagined to be able to program my own web-based application. Now I have completed *PuzzlePortal* based on my knowledge from Tech Basics 1 and 2. Especially transitioning from *Tkinter* to *Streamlit* has been a big learning experience. The app is now web-based and offers a cleaner design, better user interaction, and new features like personalized puzzle recommendations and reviews. Technologies like *Pandas* and *RapidFuzz* helped improve data handling and puzzle matching, while *Streamlit* made it easier to create a user-friendly interface.

The user feedback was mostly positive, especially regarding the navigation and the preference page. However, there are still some limitations, such as the lack of a real-time chat and advanced AI recommendations. This project has taught me a lot about app development, problem-solving, and user-centered design