Leuphana University Lüneburg

App Development

Prof. Sarah Haq

Winter Semester 2023/2024

Application Development

Mind Arena: edutainment challenges app

Alexander Kozhin

alexander.kozhin@stud.leuphana.de

Digital Media

Matriculation Nr. 3047924

Methodology

This report contains information about the 'Mind Arena' application. This is an edutainment (meaning: education and entertainment) application where users can browse the themed selection lists of influential artworks, such as movies and books.

The first phase of the development included defining the application's core objectives and features. As the primary programming tool, Python was used in the development process due to its readability and extensive libraries. Streamlit was used to create the graphical user interface of the app and to deploy the final project. Additional library pymongo was utilized to handle connection to the database. As a database, MondoDB was used. The MVP was presented to a group of 5 testers aged 25-45 years.

Application design

In this section the description of the application design will be presented.

On the **Welcome** page of the application users can either login or register, and the username and password input boxes are validated (it is checked whether both of them are filled). For the login procedure, user authentication with the remote MongoDB database is implemented. For the registration process, it is checked in the database that the entered username is not occupied yet.

Logged in users can access the **Home** page, which greets them with their username, 'getting started' section, and active challenges.

On the left of the app, users can see the sidebar with a navigation menu, containing the following buttons: **Home**, **Curated Books**, **Curated movies**, **Logout**.

On **Curated Books** and the **Curated Movies** pages users can view 3 themed selections of artworks, presented as carousels. Each selection contains 3 items, displayed as cards. Additionally, for navigation purposes, each card has an arrow. One of the key features of the app are the '**challenge yourself**' buttons, which allow adding books and movies to the 'Active challenges' list.

Once the challenges had been added, users can see the deadlines for the challenge: 7 days for movies, and 30 days for books, as well as the **'remove'** buttons on the Home page to remove the unwanted challenges.

Overall, the application has 2 pages, 3 tabs, and several essential features.

Limitations

Due to the time constraints and the complexity of certain web development concepts, the app has several limitations:

- Passwords are not stored securely, but just as plain text, which is a big security concern
- Several entities are stored statically – this includes movies selections, books selections, and active challenges. Because of that, the challenges do not save after logging out or leaving the page. However, since the Mongo database is already setup and being used, this can be easily fixed in the future. MongoDB was used because I could not figure out the Deta setup.
- The application was developed and tested on a machine operated by macOS. Therefore, additional testing for Windows operating system is required.

References

For certain function in the code, additional guidance was needed. In particular, for the setup of the database, the Streamlit and MongoDB documentation was used. For some other problems that I encountered in the development process, I used platforms like ChatGPT and Stackoverflow, in particular, for the functions create_carousel (app.py file, line 49), and the display_item function (app.py file, line 30).

1. Streamlit. (n.d.). Connect Streamlit to MongoDB. Retrieved 10.03.24 from https://docs.streamlit.io/knowledge-base/tutorials/databases/mongodb
2. MongoDB. (n.d.). Connection String URI Format. Retrieved 10.03.24 from https://www.mongodb.com/docs/guides/atlas/connection-string/
3. OpenAI. (n.d.). ChatGPT. Retrieved 10.03.24 from https://chat.openai.com/
4. Stack Overflow. (n.d.). Retrieved 10.03.24 from https://stackoverflow.com/