

Maryam Aslani

William-Watt Straße 55
21339 Lüneburg
Maryam.Aslani@stud.leuphana.de

Digital Media

Matr.-Nr. 3047078

ZenStudy is a combined study planner and productivity app I developed using Streamlit and Python. The project development began with a clear intention of creating a study aid that was an inspiration of my Tech Basics I project 'Bookexploring'. I didn't want to stick to a simple book exploring app. I wanted to develop an app that has more features, something much more practical and developed. I planned the app not only to allow users to plan tasks, deadlines, and priorities, but also consisted of additional features like an international book explorer, a productivity tracker, a Pomodoro timer, QR Code Generator and even special features on the Navigation page for much better user experience. Development was a structured, iterative process that ranged from specifying the core characteristics through to the creation of Home page, Navigation page, register and login page and session controls, ending up with user interface fine-tuning based on testers' feedback.

My coding was structured and systematic. I initially set up the basic structure of the streamlit app, and developed the main 5 features in the app, Study Planner, International Book Explorer, Productivity Tracker, Pomodoro Timer and QR code Generator. I created 2 other python files in PyCharm named auth.py for user authentication and connecting to MongoDB, and unsplashAPI.py for connecting to Unsplash for image search. Then I initialized the sessions' states in the main script to manage the 3 main pages of the app, Home, Navigation, user Login/Register, Your Account (when logged in), as well as session states in the Navigation page for each feature, and created the Streamlit app with a basic style and design. In my main Python file, I had imported important libraries such as Streamlit for the UI, Pandas for data manipulation, and external APIs for features like image search on Unsplash, listening to music on Spotify, and chatbot functionality by Hugging Face. Every component of the app was developed as an independent section. For example, I created a special "Home" page for users to greet them, a "Navigation" page where all the main tools are stored, and a "Login/ Register" page which handles user authentication. To provide a smooth and secure user experience, I have implemented session management where the users' credentials are

stored temporarily in Streamlit's session state; this has allowed features like the dynamic "Your Account" button which appears after a user logs in.

The authentication system was both a crucial and challenging part of the project to implement. I built functions to log in and register users, ensuring to capture and store appropriately details such as usernames, emails, and passwords. The account page was developed to display the user's details securely—with passwords obscured with dots—and a log out facility that removes the session state. This modular approach not only kept the code organized but also made debugging easy when something did go wrong. As an example, when dealing with the login process, I ensured that once the credentials of a user had been verified, the email from the database was taken and set into the session state, to be displayed on the account page.

Another challenge was the unpredictable nature of scrolling on the Navigation page. Whenever I opened the Navigation page, instead of displaying the top of the page Productivity Hub first, the app would scroll down to the bottom of the page where the embedded Spotify music player was. Then I found out that the problem was Hugging Face and its Chatbot. I made changes to the HuggingFace codes by moving its definition and if-else statements in the correct place, like Music part I used `st.expander` and created a button for HuggingFace, making it appear in the middle of the page after Music Feature, so it was no longer stuck to the bottom of the page and the force scrolling problem fixed.

I received useful user feedback while testing that was instrumental in making enhancements to the app. Users appreciated the overall design and found the interface intuitive and easy to navigate, with users appreciating in particular how multiple productivity apps had been seamlessly integrated. They liked the personal touch provided by the account page, as it allowed them to view their login information in a secure way. However, testers also grumbled about the scrolling issue on the Navigation page, referring to the fact that it disrupted their initial experience of the app. They suggested layout improvements, like color for side bar and finer level adjustments to the UI, like better spacing between elements and texts and making some texts bold. In addition, some users pointed out that the Search Image, Music feature and Hugging Face was a great idea, their placement affected the overall user experience.

In general, the development of ZenStudy was a very practical, valueable, engaging and also challenging process that involved careful planning, iterative coding, and extensive testing. I developed the app by writing in clear, modular language that drew together a range of interactive components into a single source, with the goal of making the process of managing study tasks and

productivity as seamless as possible. Future updates to ZenStudy can include UI layout refinement, performance optimization, and additional features such as dark mode and additional database integration for long-term preservation of user data. Overall, ZenStudy is an example of a committed development process with the aim to deliver an applicable and engaging study app, with plenty of room for refinement and expansion in future updates.