2025

FITLISTIC
HOLISTIC FITNESS

Fitlistic V2 - Project Report

TECH BASICS II (STREAM B), SARAH HAQ
FINN KÜNZEL | 4000323

LEUPHANA UNIVERSITY | February 28, 2025

Introduction

With the idea in mind to create an app that creates a bridge between physical fitness and mental wellbeing, the prototype of the Fitlistic application was created. While this prototype offered a clear view of the possibilities and the structure of an app of this kind, it was still missing the function to make it a fully working product.

This report shows how the prototype developed with the Python library Tkinter was transformed into a fully working Streamlit application. It will shed light on the design process, the limitations encountered, and how most of them were overcome in developing this fully functioning product.

Development Process

The development started with a close look at the prototype and a deeper dive into what can be reused from the prototype and what things need to be redesigned or added to help transform it into a fully functioning application. Design-wise, the prototype offered a well-thought-out color palette and structure that could be reused for the development of the new MVP version of the app. The green and blue accents from the logo combined with a clean white look created a pleasant atmosphere around the topic of fitness and wellbeing. Feature-wise, the prototype offered a view into what functionality the app should offer. It should have the possibility of tracking and monitoring a wellbeing score. The chance to complete detailed holistic workouts and a reminder function to create reminders for workouts.

While these things demonstrate the potential of the Fitlistic app, the actual application needed more functions to achieve the state of a usable app. The following things needed to be added. First the ability to register and select goals that affect the app experience. Then the reminders need to be saved in the app. The possibility of chatting with an AI about holistic fitness is also a neat feature that should be added. A workout creator that creates a tailored holistic fitness plan makes the app a real fitness app and An actual wellbeing score and chart to display it that is generated from the user logging his wellbeing is making it possible for the user to track his wellbeing.

After the to-dos were clear, the development began by doing the basics. In this case, a MongoDB database was set up and connected to the Streamlit Code. With the database set up, the next focus was migrating the existing application into Streamlit. This part went smoothly, and Streamlit’s built-in theme configuration options made creating the prototype's light blue and light green style easy. The next step was the register and login process. For the registration process, the option to enter the height and weight and choose from holistic fitness goals was implemented. Afterward, the function to check if a user is logged in before entering a page was implemented. For this purpose, an annotation pattern was used. From here on it was time to fill the existing pages with real function.

At first the Homepage was added with the possibility to log the wellbeing and displaying it in a chart after five entries were made. From here on the next part was to add a new page for the AI coach. With the knowledge from the sessions and a look at the OpenAI documentation the development of the AI coach went smoothly. While the AI coach should offer general help there also should be a non-AI workout creator that creates a useful holistic fitness plan. The next part was to add different exercises to the MongoDB so a workout creator can use these added exercises to create a user specific holistic fitness plan. For the generation ChatGPT was used to generate exercise Json documents that could be imported into the database. Then the workout creator could be implemented. Here the exercises from the database collection are fetched and are combined with a workout that is part of a seven-day plan that gets generated. The generated workout plan gets also saved as a collection per user and can then be shown on the Exercise page. The part of generating the right workout based on the exercise collections and then mapping it to a useful seven-day plan and then saving this plan to another collection was time consuming but also helped to get a great understanding of how to work with MongoDB in python Here the user can complete the days and see the instructions to the selected exercises from each day again. After this was done the reminder function was programmed to create reminders and save them in the database. The last thing that was added was the option to change details about the profile on a specific profile page.

Limitations and Solutions

During the development process, I faced different challenges that required different solutions. The first questions came up while working on the registration process, thinking about a good validation for the login and registration form that hinders the submission and also giving helpful feedback on the fields where the errors are. This topic could be cleared while asking for support in class and getting the hint to look again at the GitHub code from the lesson where we created a registration form together [1].

While looking for a solution to hide the option to make an image fullscreen, I stumbled across the Streamlit discussion platform, where a user named AvratanuBiswas explained how CSS styles can manipulate the Streamlit data [2]. This is something I used to hide the fullscreen option and later, for example, to style the sidebar and hide the login and register pages from it after a user logged in.

For the authentication and state management after the user logged in, I used ChatGPT to help me build an auth_helper file that checks if a user is logged in and, therefore, is allowed to visit a page behind the login [3].

After searching on Google for components, I stumbled across a user-developed Streamlit component developed for that. The user Tian developed a component that can be used to rate out of five or with emojis [4]. His example app was an easy way to integrate his component into my app and then connect the entries to the database. To display these entries nicely, I tried different things with the st.plotly_chart integration until I found a nice-looking implementation [5]. The only problem here was that entries from the same day made the graph look weird, so I created a check that validates if the user has at least five entries for his wellbeing score.

To get the AI Coach working I looked at the OpenAI API Python documentation and could easily set it up with the Streamlit Chat component [6]. The only limitation I encountered was how to make it possible that the AI is actually visibly writing. There is a loading screen after the user asks something, and then the AI shows the complete answer.

The most time-consuming part that took a lot of testing and debugging was handling all the fetching from the collections and storing things like the wellbeing scores, the generated workouts, or the reminders into the collections. The MongoDB documentation regarding their Python implementation was helpful [7].

User Feedback (Usability, Design and Accessibility)

Throughout development, I shared the app with several of my family and friends, who provided valuable insights on the aspect's usability, design, and accessibility. One recurring issue they noticed was the incorrect tab navigation on the registration page, which hindered keyboard-only users from cycling through form fields in a predictable order. I addressed this by adjusting the column structure of my form, thereby improving access for users relying on keyboard navigation. This also ensured that my app was more accessible.

My family and friends also highlighted how the plus and minus buttons for adjusting weight and height in the registration area caused the application to re-render prematurely, resulting in an inconvenient user experience. Investigating Streamlit's dynamic UI features, I resolved this by consolidating the input fields and preventing immediate re-execution until all user inputs had been entered.

On the design side, they recommended incorporating more emojis for added personality and suggested embedding images to break up text-only sections. These changes boosted the overall visual appeal of the interface, making it more engaging and encouraging.

Additionally, they proposed integrating hover-based notes within the wellbeing chart, allowing users to view personal notes without cluttering the interface—an update that balanced visual clarity with informative detail. All in all, the feedback from external testers helped polish the app and get a better view of how users see it.

Internal and External References

Looking back into things we covered in class

[1] S. Haq, "Tech Basics Two," GitHub. Internet: https://github.com/shaq31415926/tech-basics/tree/main/tech_basics_two
Accessed: Feb. 27, 2025.

Hiding things in Streamlit through CSS – Answer by AvratanuBiswas

[2] AvratanuBiswas, "Hide Fullscreen Option when displaying images using st.image," Streamlit Discuss. Internet: https://discuss.streamlit.io/t/hide-fullscreen-option-when-displaying-images-using-st-image/19792
Accessed: Feb. 27, 2025.

Listing different exercises with image and text (Lines 41-54 & 294-310)

[3] OpenAI, "Response generated by ChatGPT (GPT-o1) on how to check if a user is authenticated to visit a page in Streamlit," OpenAI. Internet: https://www.openai.com/chatgpt
Accessed: Feb. 27, 2025.

Rating Component in Streamlit – Component by Tian

[4] Tian, "New Component: Star Ratings!" Streamlit Discuss. Internet
https://discuss.streamlit.io/t/new-component-star-ratings/36829
Accessed: Feb. 27, 2025.

Displaying the trend of the wellbeing score in a nice chart

[5] Snowflake Inc, "st.plotly_chart " Streamlit Documentation. Internet
https://docs.streamlit.io/develop/api-reference/charts/st.plotly_chart
Accessed: Feb. 27, 2025.

Looking into how to use OpenAI API

[6] OpenAI, "Developer quickstart" OpenAI Platform. Internet:
https://platform.openai.com/docs/quickstart
Accessed: Feb. 27, 2025.

Working with MongoDB in Python

[7] MongoDB, Inc., "How to Use Python with MongoDB" MongoDB Docs. Internet:
https://www.mongodb.com/resources/languages/python
Accessed: Feb. 27, 2025.