Abstract Phase 3

The to-do list contains all the functions that have been planned in the previous phases. Furthermore, you can now also save the to-do list as a pdf. The solution can, in addition to adding and deleting items, mark them as done and change their position in the list to assign a higher priority to certain items. Furthermore, you can search/filter for certain items in the list. Finally, you can save the complete to-do list as a pdf to have it available on other devices, to send the to-do list to others or to print it.

The concept of the solution is to create an uncomplicated to-do list that does not have to do without advanced functions. This is evident in the fact that the basic functions of the to-do list are the addition and removal of entries and these alone are sufficient to define a to-do list. This type of to-do list is not efficient and user friendly. With the advanced functions, the user has more options to better organize themselves and better define their goals.

Before I started the project, I thought about which technique I wanted to use. I decided to do without frontend frameworks, because the use of these would be unnecessarily exaggerated for this project. For the backend I decided to use Python and the web framework Flask. I deliberately did not work with Django, since Django would be unnecessarily exaggerated for this project and I wanted to have a framework without previously generated code, because I wanted to learn useful concepts in web development. For the database I chose SQLite because it is fast and uncomplicated and requires minimal setup.

I first created the basic HTML structure. I then configured the database and created a Flask app. The other functions then followed on from this basic framework.

Adding Items first requires getting the actual text from the HTML form then using this text in a SQL query for adding it to the database and setting its complete status to false since they cannot be completed if they just got recently added to the list. At the end the session gets committed to the database, and the user will be redirected to the main page.

Items can be deleted by clicking on the delete button. This button passes on to the delete route with the primary key of the item that will get deleted. Using this primary key as a filter in the SQL Query the item gets deleted from the database and the session gets committed. At the end the user will be redirected to the main page.

Marking Items as Complete works the same way as for deleting items with the only difference, that instead of getting deleted the complete attribute of the items will be set to true.

In order to separate the completed items from the incomplete items two queries are used in which each contains either the completed or the incomplete items. With the help of the Template Engine Jinja2 it is possible to use loop logic in the HTML file. There are two unordered lists one for the completed items and the other for the incomplete items. Looping through the queries, we can now add every item in the query to the unordered list for the complete or incomplete items.

The functionality of moving the items up or down is implemented with JavaScript. The JavaScript file contains the documentation of the approach and idea of the implementation.

For displaying the search result another unordered list is used along with the query containing the search result. If there is a search request, the query gets rendered to the HTML file along with the complete and incomplete query. Otherwise, only the queries for separating the items get rendered to the HTML file.

Saving the To-do list as a PDF is done by using the html2pdf.js converter. By including the CDN in the HTML file and selecting the items to be converted to PDF by using JavaScript and Element Ids, the to-do list can be saved as a PDF.

This project was very useful for me as this is my first web app with frontend and backend that I have built. I learned how to build a web app with Flask. That means how to initialize a web app, create routes, request handling, connect and working with a database in the web app, working with POST, GET methods, implementing logic in the web app. In addition to that, I learned more about the DOM tree, when I implemented moving the items in the list with JavaScript. Furthermore, I learned how to manage and organize my code using GitHub.