Web Engineering

Technical Report

ToDo-Webapp

By: Marvin Samouelian, Nick Obereiter and Martin Hofsäß

# Table of contents

### Introduction

### Technical requirements and assessment criteria

### Group organization and deliverables

### Planning

### Target definition

### Choice of frameworks

### Division of responsibilities

### Implementation and showcase

### General course of action

### A simple showcase of the ToDo-Webapp

### Routing Structure

### Project conclusion

# Introduction

# Technical requirements and assessment criteria

The task is to design and implement a functional web application, following the MVC (Model View Controller) pattern. While there is free choice of Framework, the web application should at least consist of following components: HTML, CSS, Java Script, Webserver and functional backend logic.

# Group organization and deliverables

The project should be done in teams of 3-5 students and every member of the team should actively work on the development (everyone should code!).

The raw project idea and with team composition has to be registered until 26.01.202023.

The project will be presented on the 02.03.2023 or 16.03.2023 and the project documentation will be due till 23.03.2023.

# Planning

# Target definition

The program should allow different users to login into their existing accounts, create new accounts and recover their password if necessary.

Visibility of content should be connected to authentication status of the user.

The main functionality should allow the user to create ToDo’s with different topics and should be sortable by these. Delete said ToDo’s or edit their parameters such as: Name, due date…

Functionality of the application should be accessible via dedicated pages as well as via a comprehensive dashboard, which also should implement useful evaluation and analysis tools.

Further functionality could comprise of:

Mailing services to Account E-Mail, when a ToDo is about to expire.

The page design should be fully responsive.

Further statistics and services displayed and accessible through the dashboard or new pages.

# Choice of frameworks

As all of us had more or less comprehensive experience in web development, we decided to use technologies and frameworks which we were less or not proficient at all, as to challenge us a bit more.

Thus our decisions fell on React-JS and CSS for the frontend of our web application. For the backend we chose to use Pocketbase, as it would fill in for everything our backend would be comprised of.

Pocketbase functions as the whole backend of our web application in that it supplies a database, an authentication system, file storage and holds our backend logic as a Go framework.

# Division of responsibilities

|  |  |
| --- | --- |
| Login Page | Nick Obereiter |
| Registration Page | Nick Obereiter |
| Reset Password Page | Nick Obereiter |
| Home/Dashboard | Martin Hofsäß / Nick Obereiter |
| Dashboard sub functionality | Martin Hofsäß / Nick Obereiter |
| Create ToDo | Marvin Samouelian |
| ToDo overview Page | Marvin Samouelian |
| ToDo tomorrow Page | Marvin Samouelian |
| ToDo today Page | Marvin Samouelian |
| Topics Page | Nick Obereiter |
| Login/Registration Logic | Marvin Samouelian |
| Database | Marvin Samouelian |
| Backend | Marvin Samouelian |
|  |  |
| Project Documentation | Martin Hofsäß |

# Implementation and showcase

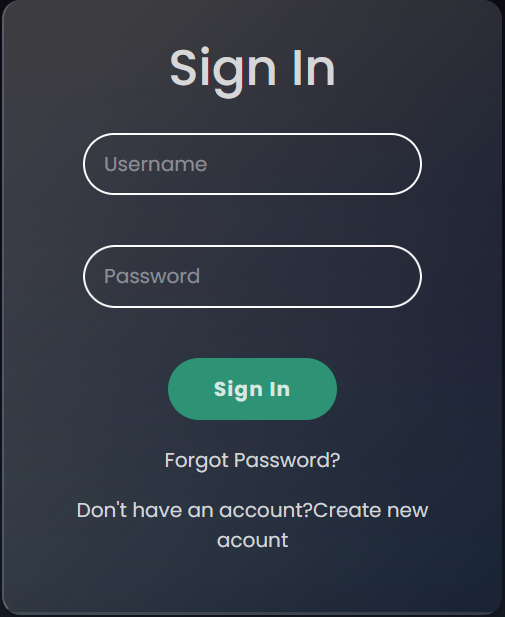
# General course of action

With the scope set for the ToDo-Webapp and the extra challenge added, due to unfamiliar technologies and frameworks, the general idea was to not just use the time during the lecture but to also continue development outside of lectures.

As visible in the table under “2.3 Division of responsibilities” the amount contributed to the project was different for the team members.

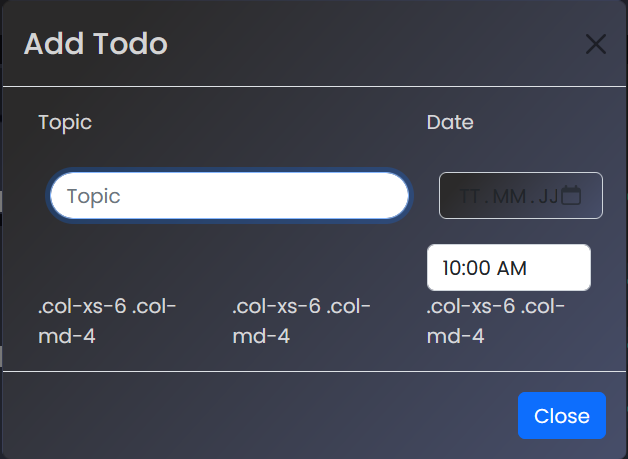
# A simple showcase of the ToDo-Webapp

The user arrives at the login screen and enters his login credentials.

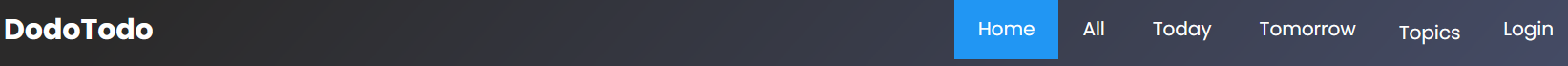


After successfully authenticating he will be greeted on his home-site/dashboard and gets a comprehensive overview of all important information in relation to his existing ToDo’s.

The user decides to create a new ToDo. For this he can use the button in the bottom right of every page.



The Navigation Bar at the top of the screen allows the user to navigate between the different sites.



# Routing structure

