**Informatyka Stosowana gr.1 s.6 2021/22  
Mateusz Birkholz**



Projekt – Lista zadań z wykorzystaniem frameworków React i Flask

Programowanie serwisów sieciowych

# Założenia projektu

Projekt zakłada zbudowanie aplikacji client-server udostępniającej użytkownikom dostęp do listy zadań, możliwość dodawania elementów do listy, usuwania ich, oznaczania czy edytowania. Każdy z elementów na liście posiada swój opis, datę, oraz priorytet. Strona pozwala na rejestrację oraz logowanie użytkowników dzięki czemu każdy użytkownik ma dostęp do swojej własnej listy. Strona komunikuje się z API serwerowym a przesłane do niego dane zostają sprawdzone i zapisane w bazie danych. Front-end aplikacji został przygotowany we frameworku React natomiast server w Pythonie z wykorzystaniem frameworku Flask.

# Opis działania serwera

Server jest prostym API napisanym w Pythonie z wykorzystaniem Flaska. Do obsługi bazy danych wykorzystuje SQLAlchemy w wersji flaskowej.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Do obsługi bazy danych przegotowane zostały 2 klasy User i Todo. Posiadają odpowiednie dla nich pola:

* User
  + Id
  + Name
  + Password
  + Todos – relacja 1 do wielu (lista todo danego uzytkownika)
* Todo
  + Id
  + Text
  + isComplete
  + date
  + prio

Obraz zawierający tekst

Opis wygenerowany automatycznie

Poza polami oba posiadają metodę pozwalająca na przekonwertowanie obiektu do formatu JSON.

API zawiera metody:

* login – na podstawie loginu i hasła
* register – po sprawdzeniu czy dany użytkownik już nie istnieje
* todos – zwraca listę zadań danego użytkownika na postawie jego id
* completeTodo – odznaczenie zadania jako wykonanego lub nie
* deleteTodo – usunięcie zadania z listy
* addTodo – dodanie zadania do listy
* editTodo – edycję zadania z listy

Przykład (editTodo):

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Układ Komponentów



# Opis komponentów

## App

App jest podstawowym kompenentem. Zawiera informacje o zalogowanym użytkowniku na podstawie których definiowana jest jego zawartość. Posiada metody takie jak login czy register przekazywane do komponentów LoginForm i RegisterForm. Funkcje te pozwalają na manipulacje stanami zawierającymi informacje o zalogowanym użytkowniku a także odpowiedzialne są za aktualizowanie informacji o błędach podczas rejestracji lub logowania.

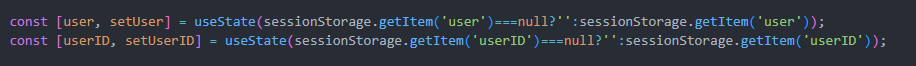
Jeśli użytkownik jest zalogowany zamiast formularzy do logowania i rejestracji wyświetli się komponent TodoList zawierający indywidualną listę zadań użytkownika.

Render komponentu:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Funkcja logowania wysyła zapytanie POST do API serwerowego z zapytaniem o sprawdzenie czy użytkownik o takim haśle i loginie istnieje. Jeśli serwer zwróci status success dane użytkownika zostają zapisane w stanach a strona przechodzi do widoku listy. Jeśli nie wyświetlony zostanie błąd. Strona wykorzystuje również mechanizm sessionStorage dzięki któremu po odświeżeniu strony użytkownik dalej pozostaje zalogowany:



Obraz zawierający tekst

Opis wygenerowany automatycznie

## LoginFrom i RegisterForm

LoginFrom i RegisterForm są prawie tym samym komponentem. Różnią się jednym polem danych, komunikatami błędów oraz wykonywaną przez formularz funkcją. Oba są proste w budowie więc nie ma co tutaj zbyt dużo o nich pisać.

Obraz zawierający tekst

Opis wygenerowany automatycznie

## TodoList

TodoList jest komponentem wyświetlanym po zalogowaniu użytkownika i głównym elementem sterującym widoku, ponieważ to wewnątrz niego znajdują się wszystkie funkcje odpowiadające za komunikowanie się z API serwera. Posiada stan todos który jest listą zadań danego użytkownika. W momencie wywołania dowolnej metody komunikującej się z API serwer po wprowadzeniu ewentualnych zmian w bazie wysyła aktualną listę zadań która jest aktualizowana automatycznie przez mechanizm stanu React. Na początku pobierana jest lista zadań która później w przypadku akcji użytkownika jest aktualizowana:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przykładowa funkcja komunikująca się z API wygląda w następujący sposób:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Najpierw sprawdza czy zadania zostało zmienione lub czy nie zostały wprowadzone puste lub błędne dane. Następnie jeśli wszystko jest w porządku wysyła zapytanie GET do serwera i czeka na odpowiedź. Odpowiedzią jest lista zadań danego użytkownika.

Render komponentu najpierw tworzy obiekt TodoForm a następnie mapuje listę zadań generując obiekt Todo dla każdego zadania. Na końcu dodawany jest przycisk Wyloguj który usuwa informacje o użytkowniku ze stanów oraz z sessionStorage:

Obraz zawierający tekst

Opis wygenerowany automatycznie

## Todo

Todo jest komponentem odpowiadającym jednemu zadaniu. Wyświetla on wszystkie informacje o zadaniu a także zawiera 2 przyciski służące do usunięcia zadaniu lub przejścia do jego edycji. Todo zawiera stan edit który określa w jaki stanie w danej chwili jest zadanie. Jeśli zadanie nie jest edytowane wyświetla się standardowy kafelek z informacjami oraz przyciskami. Jeśli jest w trybie edycji zamiast kafelka generowany jest TodoForm. Dzięki niemu użytkownik mam możliwość prowadzenia innych danych do zadania.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

## TodoForm

TodoForm jest formularzem umożliwiającym dodawanie lub wcześniej omawiane edytowanie zadania. Funkcja którą ma realizować jest mu przekazywana z komponentu nadrzędnego. Zawiera on stany dzięki którym zapamiętuje wartość pól wejściowych. Render różni się jedynie wykonana funkcją oraz klasami CSS w zależności od tego czy zadaniem jest edycja czy dodawania zadania.

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Prezentacja strony

Strona dla użytkownika niezalogowanego:

Obraz zawierający tekst, wewnątrz, monitor, ekran

Opis wygenerowany automatycznie

Błędy przy rejestracji lub logowaniu:

Obraz zawierający tekst, monitor, zrzut ekranu, srebrny

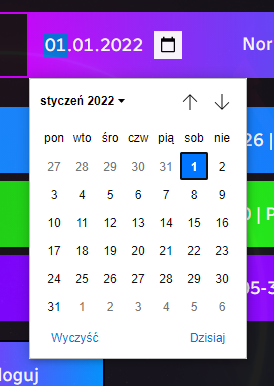
Opis wygenerowany automatycznie

Strona po zalogowaniu:

Obraz zawierający tekst, monitor, wewnątrz, ekran

Opis wygenerowany automatycznie

Wybór daty i priorytetu:



Obraz zawierający tekst, tablica wyników, zrzut ekranu

Opis wygenerowany automatycznie

Edycja pierwszego zadania:

Obraz zawierający tekst, zrzut ekranu, ekran

Opis wygenerowany automatycznie

Zadania odznaczone jako wykonane:

Obraz zawierający tekst, monitor, ekran, zrzut ekranu

Opis wygenerowany automatycznie

Usunięte zadanie:

Obraz zawierający tekst, zrzut ekranu, monitor, ekran

Opis wygenerowany automatycznie