**Implementacja**

### **Rejestracja**

Autoryzacja wymagana jest ze względu na konieczność posiadania konta w celu korzystania z aplikacji. Aby utworzyć konto należy podać unikatowy adres e-mail, unikatową nazwę użytkownika oraz hasło. Klient wysyła żądanie typu POST do serwera, a następnie tworzony jest nowy obiekt *User* w bazie danych.

module.exports.register = (req, res) => {

const user = new User();

user.name = req.body.name;

user.email = req.body.email;

user.setPassword(req.body.password);

user.save(() => {

const token = user.generateJwt();

res.status(200);

res.json({

token: token

});

});

};

Hasło przetwarzane jest za pomocą funkcji wyprowadzania klucza PBKDF2 z zastosowaniem algorytmu SHA512 oraz 1000 iteracji. Wygenerowany ciąg znaków dodatkowo wzmocniony jest poprzez dodanie soli, czyli losowego, 16 bajtowego ciągu znaków.

userSchema.methods.setPassword = function(password) {

this.salt = crypto.randomBytes(16).toString('hex');

this.hash = crypto

.pbkdf2Sync(password, this.salt, 1000, 64, 'sha512')

.toString('hex');

};

Pomyślne utworzenie użytkownika zwraca kod odpowiedzi http 200 oraz następuje automatyczne zalogowanie użytkownika.

### **Logowanie**

Logowanie oraz utrzymanie sesji użytkownika odbywa się z wykorzystaniem tokenu JWT. Pierwszym krokiem jest zweryfikowanie danych wprowadzonych przez użytkownika – adresu email oraz hasła. Hasło przetwarzane jest wcześniej opisaną funkcją, a otrzymany hasz porównywany jest z uprzednio utworzonym haszem przy rejestracji. Jeżeli ciągi znaków są takie same zostaje wygenerowany token JWT, który przechowywany jest w *localStorage* przeglądarki. Wylogowanie użytkownika jest jednoznaczne z usunięciem tokenu z pamięci. Token może również zostać usunięty samoistnie po upłynięciu 7 dni od utworzenia.

**Zarządzanie zadaniami**

Po zalogowaniu użytkownik zostaje przekierowany na stronę z zadaniami. Jeżeli użytkownik nie należy do żadnej grupy, na stronie zostanie wyświetlony stosowny komunikat przekierowujący go do strony przeznaczonej do zarządzania grupami użytkownika. Użytkownik przypisany do grupy zobaczy pełną funkcjonalność strony na której może dodawać, edytować oraz usuwać zadania przeznaczone dla każdej z jego grup. Ponadto, po rozwinięciu paska zadania użytkownik może przeczytać oraz edytować opis wybranego zadania. Po wykonaniu akcji użytkownik jest informowany o sukcesie lub niepowodzeniu za pomocą snackbarów.

Zapytania do backendu obsługiwane są przez moduł “axios”.

Wyświetlenie zadań obsługuje zapytanie GET.

Utworzenie zadania obsługuje zapytanie POST.

Edycja zadania obsługuje zapytanie PUT.

Usunięcie zadania obsługuje zapytanie DELETE.

**Zarządzanie grupami**

Po zalogowaniu użytkownik ma możliwość przejścia na stronę zarządzania swoimi grupami, gdzie może przejrzeć, edytować i usuwać grupy do których należy oraz dodawać nowe. Każda akcja po ukończeniu informuje użytkownika o sukcesie lub niepowodzeniu poprzez snackbary.

Wszystkie zapytania do backendu odbywają się za pomocą modułu “axios”

Wyświetlenie grup wywołuje wysłanie zapytania typu GET.

Utworzenie grupy powoduje wysłanie zapytania typu POST.

Edycja grupy wywołuje wysłanie zapytania typu PUT.

Usunięcie grupy wywołuje wysłanie zapytania typu DELETE.

**Testowanie**

Aplikacja była testowana manualnie przed każdym mergem nowego brancha do mastera przez inną osobę z zespołu niż implementująca nową funkcjonalność. Wykorzystana została również technika *code review* w celu wykrycia błędów oraz rozwoju dobrej jakości kodu.

**Podsumowanie**

Prace nad aplikacją Task Manager do zarządzania zadaniami przebiegły sprawnie. Zadania w harmonogramie zostały wykonane, a ponad to zaimplementowany został system obsługi błędów oraz informowania użytkownika o stanie aplikacji w formie *snackbarów* pojawiających się przy wykonywanych akcjach. Ze względu na ograniczony czas zostały zaimplementowane jedynie kluczowe funkcjonalności, natomiast aplikacja posiada potencjał do dalszego rozwoju. Kwestie, które warto byłoby rozważyć w przyszłości to między innymi:

* Funkcjonalność dodawania użytkowników do znajomych
* Zapytania z prośbą o dołączenie do zespołu
* System powiadomień