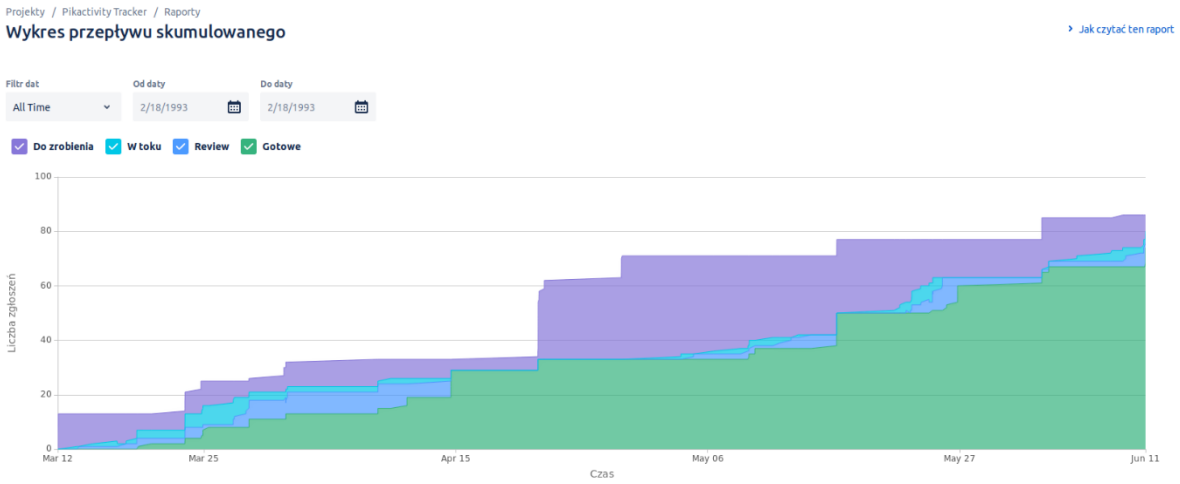
PATRYK KARBOWNIK
ANIELA KOSEK
ALEKSANDRA OKRUTNY
ZUZANNA SANTOROWSKA

PIKACTIVITY TRACKER – DOKUMENTACJA KOŃCOWA

RAPORTY

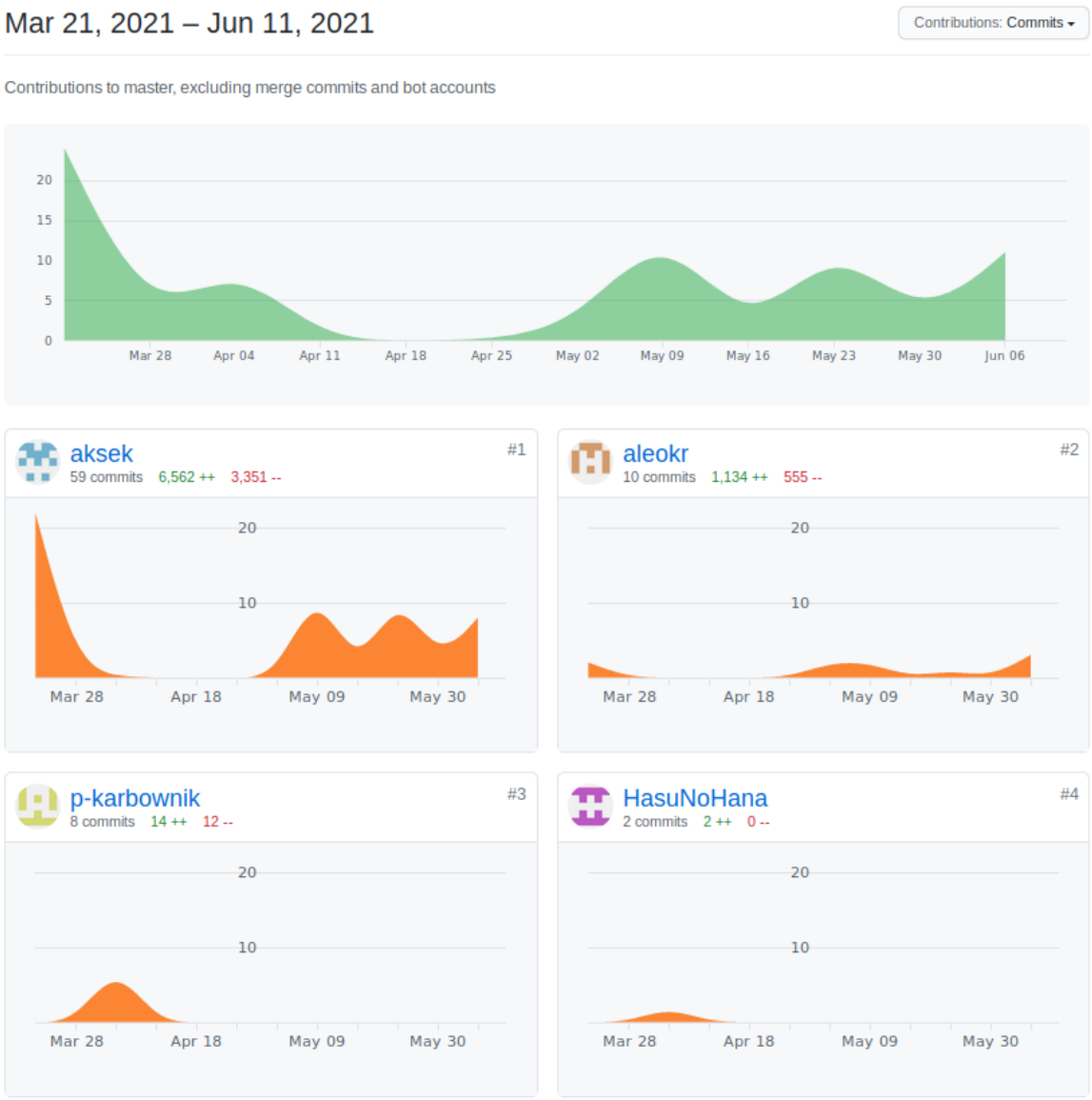


CZAS SPĘDZONY NAD ZADANIAMI [PEŁNE DNI]

Członek zespołu	Czas spędzony
Aleksandra Okrutny	9 dni
Aniela Kosek	10 dni
Patryk Karbownik	9,5 dnia
Zuzanna Santorowska	9 dni

Statystyki liczby commitów są zaburzone przez kod generowany automatycznie oraz commity wykonane bez użytku odpowiednich kont github.

WEBSITE



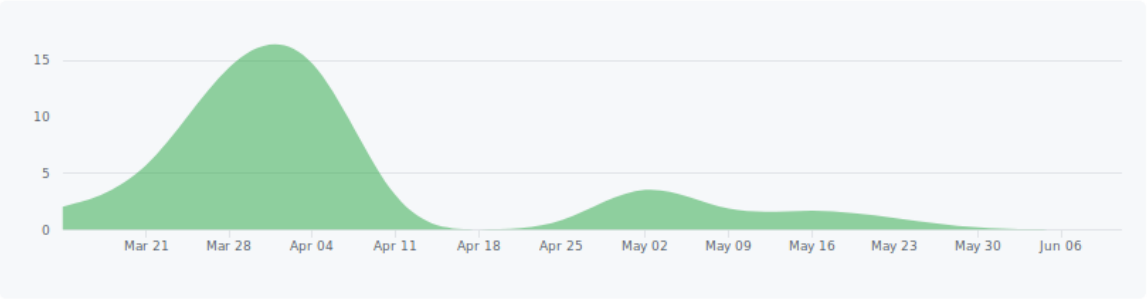

```
=============================== Coverage summary ===============================
Statements   : 16.96% ( 29/171 )
Branches     : 7.14% ( 3/42 )
Functions    : 8.7% ( 6/69 )
Lines        : 17.58% ( 29/165 )
================================================================================
```

LIBRARY

Mar 14, 2021 – Jun 11, 2021

Contributions: Commits

Contributions to main, excluding merge commits and bot accounts



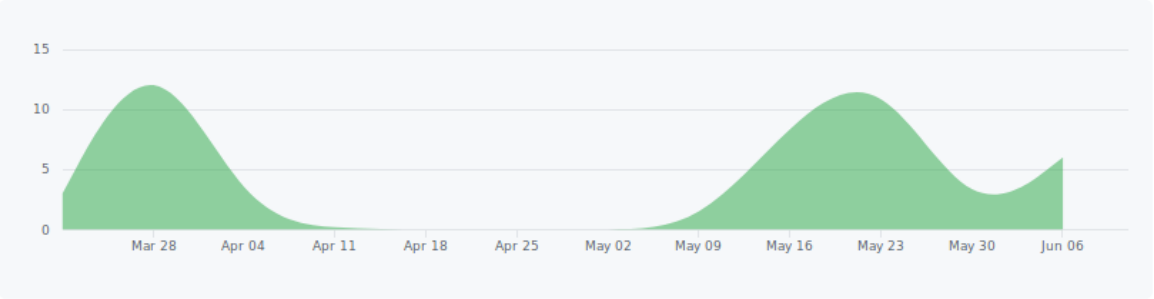

```
------------|---------|----------|---------|---------|-------------------
File        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------|---------|----------|---------|---------|-------------------
All files   |   72.73 |       50 |      80 |   76.19 |
 Event.ts   |   54.55 |       50 |     100 |      60 | 6-13
 Tracker.ts |   90.91 |      100 |      75 |   90.91 | 13
------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.95 s
Ran all test suites.
```

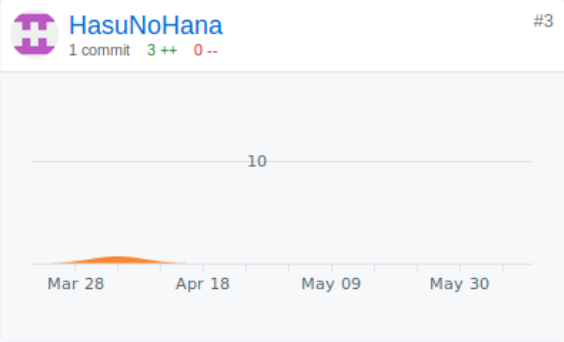
SERVER

Mar 21, 2021 – Jun 11, 2021

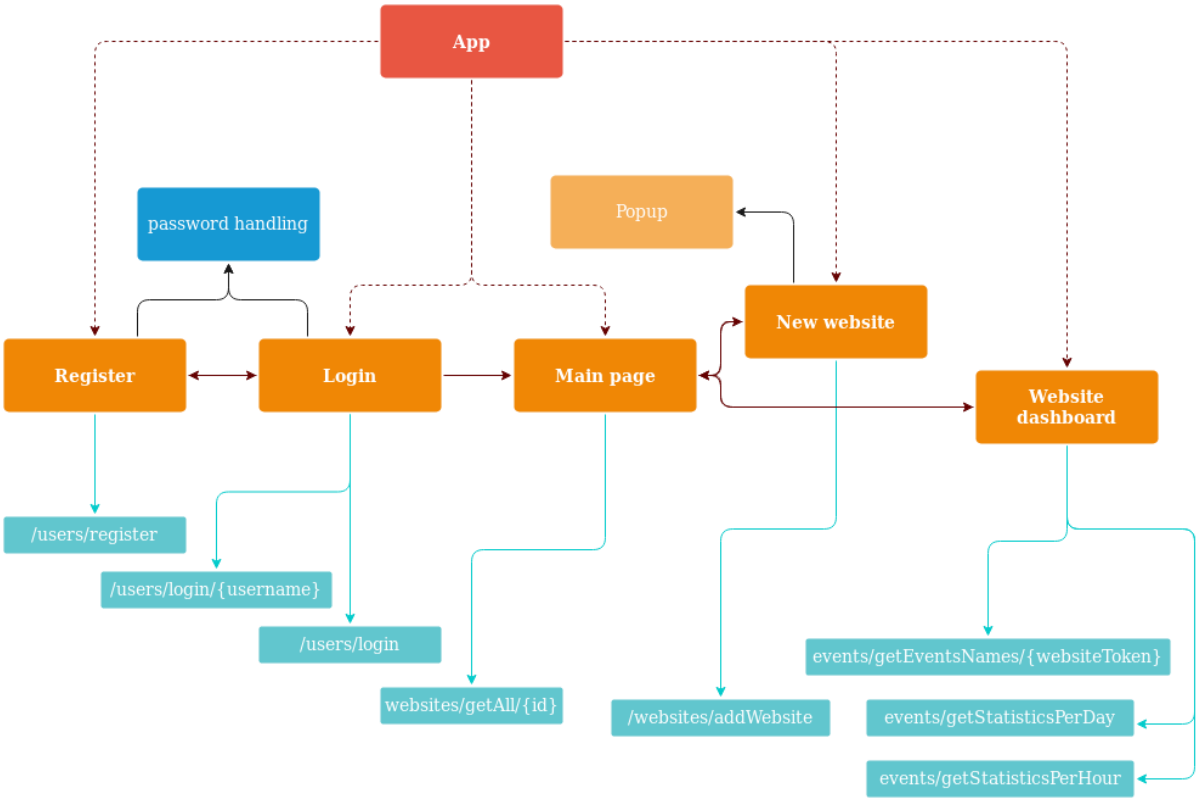
Contributions: Commits

Contributions to main, excluding merge commits and bot accounts





FRONTEND



KOMPONENTY

REGISTER

Strona pozwalająca użytkownikowi na zarejestrowanie nowego konta. Wymaga podania adresu email oraz dwukrotnego powtórzenia hasła. Wykonuje zapytanie do backendu na endpoint /users/register.

LOGIN

Strona, która domyślnie pokazuje się użytkownikowi jako pierwsza. Wykonuje dwa zapytania: pierwsze przekazuje do backendu jedynie email użytkownika, w celu pobrania soli. Następnie, otrzymana wartość jest wykorzystana do obliczenia funkcji skrótu. W kolejnym zapytaniu wysyłana jest nazwa użytkownika i skrót hasła. W odpowiedzi zwracane jest id użytkownika lub informacja o niepoprawności hasła.

MAIN PAGE

Na tej stronie znajduje się tabela, w której wyświetlane są podstawowe informacje na temat stron, które użytkownik śledzi. W tabeli znajdują się informacje odnośnie najpopularniejszego eventu na danej stronie, liczbie eventów łączenie czy dacie ostatniego eventu. Dane wyświetlane w tabeli pobierane są z endpointu user/{userId}.

NEW WEBSITE

Widok pozwalający użytkownikowi zarejestrować nową stronę, na której chciałby monitorować ruch. Formularz widoczny w oknie zawiera dwa pola: nazwę i url strony, którą użytkownik chce zarejestrować. Po

zatwierdzeniu podanych danych użytkownik otrzymuje token, który będzie wykorzystywany przy wysyłaniu zdarzeń do serwera. Po dodaniu strony użytkownik może przejść do okna głównego aplikacji lub dodać kolejną stronę.

WEBSITE DASHBOARD

W oknie przedstawione są statystyki danej strony. Użytkownik po wybraniu przedziału czasu, z którego mają pochodzić dane i poziomu grupowania ich (liczba zdarzeń na dzień / godzinę), wybiera nazwę zdarzenia, którego wystąpienia powinny pojawić się na wykresie.

POZOSTAŁE PLIKI

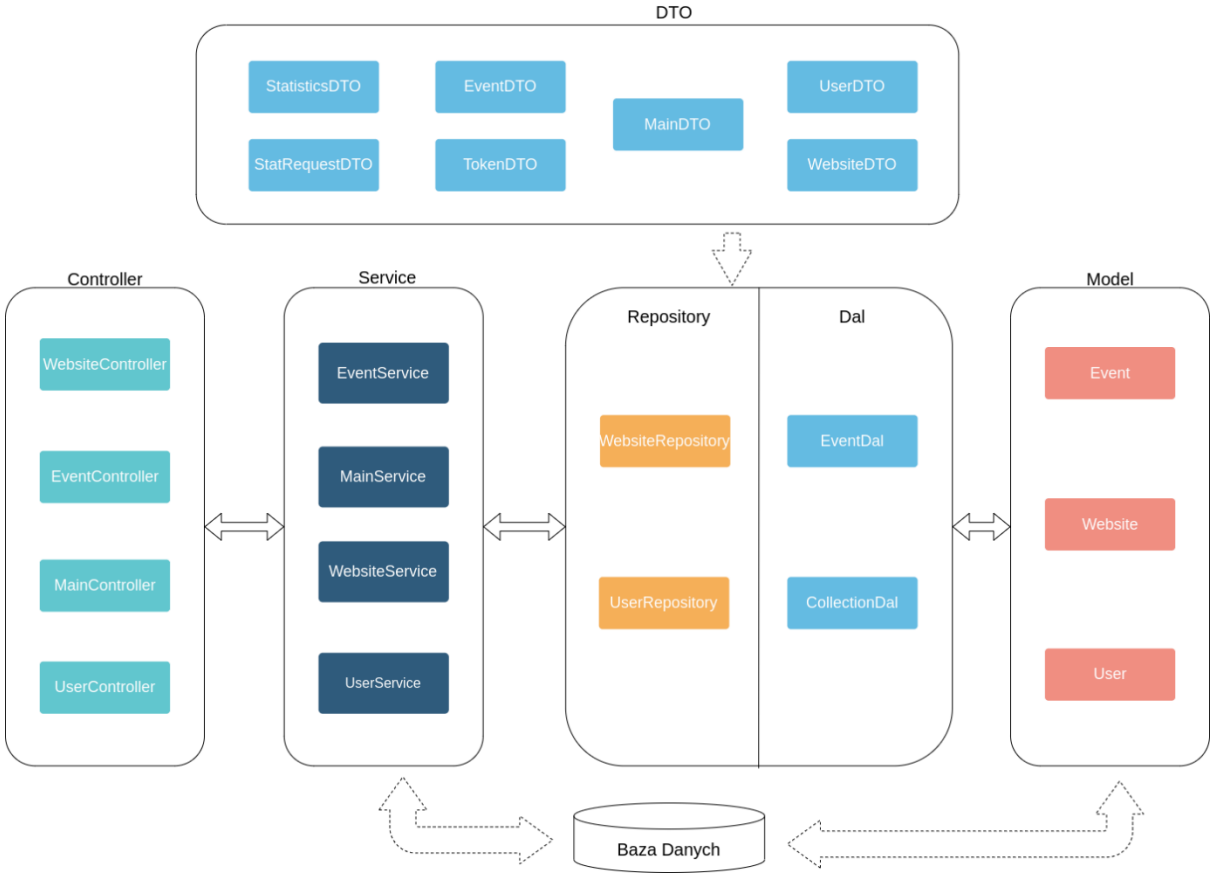
PASSWORD HANDLING

W tym pliku znajdują się funkcje przeznaczone do obliczania funkcji skrótu hasła. Definiowany jest także typ PasswordData, przechowujący hash hasła oraz sól, zwracany przez obie poniższe funkcje.

saltHashPassword(userpassword: string) – zwraca funkcję skrótu hasła oraz wygenerowaną sól

sha512(password: string, salt: string) – działa jak saltHashPassword, ale korzysta z soli podanej jako argument

BACKEND



WARSTWY

WARSTWA KONTROLERÓW

Jej zawartość przechowywana jest w pakiecie *controller*. Stworzyliśmy metody klas *WebsiteController*, *UserController* oraz *EventController*. Zapewniają one podstawową komunikację sieciową oraz udostępniają niezbędną funkcjonalność dla frontendu w oparciu o architekturę REST.

WARSTWA SERWISÓW

Znajduje się ona w pakiecie *service*. W tym sprincie zaimplementowaliśmy funkcjonalności klas: *UserService*, *WebsiteService* oraz częściowo *EventService*. Została w nich zamknięta logika biznesowa, a ich metody są wywoływane przez odpowiednie klasy z pakietu *controller*.

WARSTWA DOSTĘPU DO BAZ DANYCH

Należy do nich zawartość pakietów *Repository* oraz *dal*. W pierwszym z nich znajdują się klasy korzystające ze Spring Data JPA. Są one odpowiedzialne za wykonywanie podstawowych operacjach w naszej bazie PostgreSQL na tabelach *user* i *webistes*. Na potrzeby realizacji projektu napisaliśmy dodatkowe metody realizujące niezbędne zapytania.

Z kolei w pakiecie dal znajdują się klasy realizujące komunikację z bazą *MongoDB*. Pierwotnie do tego celu chcieliśmy wykorzystać interfejs *MongoRepository*, jednak nie pozwalał on na tworzenie nowych kolekcji w bazie, dlatego napisaliśmy od podstaw całe *data access layer* opartą na funkcjonalności klas *MongoTemplate* oraz *MongoClient* z frameworka *Spring*.

POZOSTAŁE

DATA TRANSFER OBJECTS

Wyszczególniliśmy następujące klasy tego typu: *EventDTO*, *TokenDTO*, *UserDTO*, *WebsiteDTO*. Są one wykorzystywane w warstwach kontrolerów oraz serwisów. W naszym projekcie znajdują się one w pakiecie DTO.

MODELS

Na tą część składają się klasy: *User*, *Website*, *Event*. Dwie pierwsze odpowiadają poszczególnym wierszom z bazy typu SQL, trzecia odzwierciedla elementy przechowywane w bazie NoSQL. Wspomniane wyżej klasy znajdują się w pakiecie *model*.