Warsztaty VI - JEE

v3.1

Twitter

Celem warsztatów jest napisanie pełnej i funkcjonalnej aplikacji w stylu Twittera. Aplikacja ma implementować następujące funkcjonalności:

- **Użytkownicy**: dodawanie, modyfikacja niekluczowych informacji o sobie, usuwanie swojego konta. Użytkownik ma być identyfikowany po emailu (nie może się powtarzać).
- **Wpisy**: Każdy użytkownik może stworzyć nieograniczoną liczbę wpisów. Maksymalna długość wpisu to `140` znaków.
- **Komentarze**: pod każdym wpisem inni użytkownicy mają mieć możliwość wpisywania komentarzy. Maksymalna długość komentarza to `60` znaków.
- **Wiadomości**: Każdy użytkownik może wysłać innemu użytkownikowi wiadomość.

Strony, które ma mieć aplikacja

Strona główna

Strona wyświetlająca wszystkie Tweety jakie znajdują się w systemie (od najnowszego do najstarszego). Nad nimi ma być widoczny formularz do stworzenia nowego wpisu.

Strona logowania

Strona ma przyjmować email użytkownika i jego hasło.

- jeżeli są poprawne, to użytkownik jest przekierowany do strony głównej,
- jeżeli nie – do strony logowania, która ma wtedy wyświetlić komunikat o błędnym loginie lub haśle,
- strona logowania ma mieć też link do strony tworzenia użytkownika.

Strony, które ma mieć aplikacja

Strona tworzenia użytkownika

Strona ma pobierać email i hasło.

- Jeżeli takiego adresu email nie ma jeszcze w systemie (tabeli w bazie), to rejestrujemy użytkownika i logujemy (przekierowanie na stronę główną).
- Jeżeli taki adres email jest, to przekierowujemy do strony tworzenia użytkownika (ta sama strona) i wyświetlamy komunikat o zajętym adresie email.

Strona wyświetlania użytkownika

Strona ma pokazać wszystkie wpisy danego użytkownika (dodatkowo pod każdym liczbę komentarzy do danego wpisu).

Na tej stronie ma być też przycisk, który umożliwi nam wysłanie wiadomości do tego użytkownika.

Strony, które ma mieć aplikacja

Strona wyświetlania wpisu (tweeta)

Ta strona ma wyświetlać:

- wpis
- autora wpisu
- wszystkie komentarze do każdego z wpisów
- formularz do tworzenia nowego komentarza dla wpisu

Strona edycji użytkownika

Użytkownik ma mieć możliwość edycji informacji o sobie i zmiany hasła.

Pamiętaj o tym, że użytkownik może edytować tylko i wyłącznie swoje informacje.

Strony, które ma mieć aplikacja

Strona z wiadomościami

Użytkownik ma mieć możliwość wyświetlenia listy wiadomości, które otrzymał i wysłał.

- Wiadomości wysłane mają wyświetlać odbiorcę, datę wysłania i początek wiadomości (**pierwsze 30 znaków**).
- Wiadomości odebrane mają wyświetlać nadawcę, datę wysłania i początek wiadomości (**pierwsze 30 znaków**).

Wiadomości jeszcze nieprzeczytane powinny być jakoś oznaczone (np. pogrubiony tekst zawierający nadawcę).

Stronę pojedynczej wiadomości

Wszystkie informacje o wiadomości:

- nadawca
- odbiorca
- treść

Informacje

- Bez zalogowania mają być dostępne **tylko**:
  - strona logowania
  - strona tworzenia nowego użytkownika
- Gdy skończysz pracę nad warsztatem (w dniu warsztatu), wyślij Mentorowi informację zawierającą następujące dane:
  - adres repozytorium, na którym znajduje się twój kod
  - listę zaimplementowanych funkcjonalności
  - zrzut bazy danych
- Te same informacje prześlij gdy całkowicie zakończysz pracę nad warsztatem.

Zadania

Zadanie 1

Przygotowanie

- Utwórz projekt o nazwie **Warsztat6**.
- Uzupełnij zestaw zależności odpowiedzialnych za Spring MVC.
- Uzupełnij podstawowy zestaw zależności dla korzystania z Hibernate.
- Uzupełnij podstawowy zestaw zależności dla korzystania z Spring Data.
- Dodaj plik konfiguracyjny dla hibernate - persistence.xml.
- Utwórz plik konfiguracyjny aplikacji oraz niezbędne ziarna.

Zadanie 1

Przygotowanie

- Załóż nowe repozytorium Git na GitHubie i nową bazę danych
- Pamiętaj o robieniu backupów bazy danych (najlepiej co każde ćwiczenie) i tworzeniu commitów z opisem w języku angielskim (również co każde ćwiczenie).
- Stwórz plik .gitignore i dodaj do niego elementy ignorowane np. podstawowe pliki projektu (.project, .settings).
- Możesz skorzystać z serwisu https://www.gitignore.io/

Zadanie 2

Ćwiczenia z wykładowcą

Utwórz encję **User** encja ma posiadać pola:

- id
- username
- password
- enabled
- email

Ustaw wszystkie pola jako wymagane

Ustaw walidację poprawności adresu email.

Pamiętaj o haszowaniu hasła.

Zadanie 2

Ćwiczenia z wykładowcą

Utwórz kontroler wyświetlający stronę główną naszej aplikacji.

Utwórz kontroler umożliwiający logowanie oraz rejestrację użytkownika.

Utwórz widoki oraz niezbędne formularze.

Zadanie 3

Wpisy

- Czas na dodanie głównej funkcjonalności do naszej strony – czyli wpisy.
- Pamiętaj o stworzeniu relacji między tą tabelą a tabelą użytkowników.
- Użytkownik może mieć wiele wpisów, wpis może mieć tylko jednego Usera.

Stwórz encję **`Tweet`**.

Ma ona zawierać co najmniej:

- **`id`**
- **`user`**
- **`text`**
- **`created`**

Zadanie 3 – repozytorium Tweet

Ma implementować następujące funkcjonalności:

- Metodę, która wczyta wszystkie Tweety stworzone przez zadanego użytkownika.
- Metodę, która wczyta wszystkie Tweety stworzone przez zdanego użytkownika - podając jako parametr id użytkownika
- Jeżeli widzisz jeszcze jakieś potrzebne funkcje to możesz je dopisać.

Zadanie 3 – ciąg dalszy

- Zmodyfikuj stronę główną tak, aby wyświetlała wszystkie wpisy.
- Pobierz je za pomocą odpowiedniej metody repozytorium, przekaż do widoku, a następnie wyświetl.
- Zmodyfikuj stronę główną tak, aby miała formularz do stworzenia nowego wpisu. Pamiętaj o obsłużeniu tego formularza na tej samej stronie. Ma on tworzyć nowy tweet przypisany do zalogowanego Usera.
- Utwórz stronę użytkownika tak, żeby wyświetlała wszystkie jego wpisy.
- Stwórz stronę, która wyświetli wszystkie informacje o pojedynczym wpisie.

Zadanie 4

Komentarze

- Dodajemy możliwość pisania komentarzy pod wpisami.
- Stwórz relację między komentarzami a wpisami.
- Stwórz relację między użytkownikami a wpisami.

Stwórz encję **`Comment`**.

Ma ona zawierać:

- **`id`**
- **`user`**
- **`post`**
- **`created`**
- **`text`**

Zadanie 4 – repozytorium Comment

Ma implementować następujące funkcjonalności:

- Metodę pobierającą wszystkie komentarze dla zadanego postu.
- Metodę pobierającą wszystkie komentarze dla zadanego postu - podając jako parametr id postu.
- Aplikacja ma pokazywać komentarze posortowane od najnowszego do najstarszego.
- Zmodyfikuj stronę pojedynczego wpisu tak, aby wyświetlał on wszystkie swoje komentarze i miał formularz do stworzenia nowego komentarza.
- Komentarz ma wyświetlić informację o autorze.

Zadanie 5

Wiadomości

Czas na wysyłanie wiadomości.

- Stwórz encję **Message**, która będzie trzymała wiadomości.
- Połącz ją z tabelą użytkowników relacją wiele do dwóch. Czyli mają powstać dwie relacje wiele do jednego. Wiadomość ma dwóch użytkowników nadawcę i odbiorcę a użytkownik ma wiele wiadomości.
- W tabeli stwórz pole trzymające informację, czy wiadomość została przeczytana np.:
  - `1` – wiadomość przeczytana,
  - `0` – wiadomość nieprzeczytana.

Stwórz repozytorium wiadomości, wzorując się na poprzednich zadaniach.

Zadanie 5 – ciąg dalszy

- Stwórz stronę wyświetlającą wszystkie wiadomości, które użytkownik otrzymał i wysłał.
- Do strony wyświetlającej użytkownika dodaj przycisk przekierowujący na stronę z formularzem do wysłania wiadomości do tego użytkownika (nie powinno się dać wysłać wiadomości do samego siebie!).
- **Pamiętaj o tym, że nowo stworzona wiadomość powinna być oznaczona jako nieprzeczytana.**
- Dodaj stronę, która wyświetli informację o wiadomości (jeżeli otwiera ją odbiorca, pamiętaj żeby oznaczyć wiadomość jako przeczytaną).

Jeżeli coś nie jest jasne, zapytaj wykładowcę.