

Zaawansowane Aplikacje Internetowe - Projekt /25Z/

Zaprojektuj i zaimplementuj aplikację wspomagającą zbieranie i prezentację wyników pomiarów wybranej/nych wielkości fizycznej/ych w wielu seriach. Prezentacja zebranych danych w GUI w formie wykresu w funkcji czasu oraz w formie tabeli, z możliwością ograniczenia przedziału czasu podlegającego prezentacji. Użytkownik może wybrać, które serie mają znaleźć się na wykresie. Aplikacja powinna umożliwiać zalogowanemu użytkownikowi edycję dowolnego rekordu.

Specyfikę zbieranych danych proszę dopasować do swoich zainteresowań, potrzeb itp.

Opcjonalne rozszerzenie: obsługa różnych źródeł danych, tj. oprócz interfejsu użytkownika działającego w przeglądarce internetowej, aplikacja może obsługiwać autonomiczne sensory, które komunikują się z backendem z pominięciem GUI, tj. bezpośrednio przez odpowiednie endpointy w API. Aplikacja powinna przyjmować dane wyłącznie od zarejestrowanych wcześniej sensorów. W ramach demonstracji zaproponować skrypt (np. w Pythonie) imitujący taki sensor (makietę wysyłającą wyniki obliczane lokalnie, tj. bez angażowania realnego sprzętu).

Specyfikacja funkcjonalna aplikacji (minimalne wymagania):

1. Jako wynik pomiaru rozumiemy przynajmniej jedną liczbę zmiennoprzecinkową opatrzoną znacznikiem czasu oraz identyfikatorem serii, do której ten wynik należy.
2. Serie pozwalają powiązać ze sobą wyniki pochodzące z tego samego eksperymentu, sensora itp., a tym samym wyświetlić je w osobnej kolumnie w tabeli oraz jako odrębną krzywą na wykresie. Seria powinna mieć mieć edytowalną nazwę, dopuszczalny przedział wartości min/max oraz konfigurowalną jakąś cechę graficzną (np. kolor lub ikonę).
3. Są dwie grupy użytkowników: czytelnicy (niezalogowani) i administratorzy (zalogowany).
4. Użytkownicy niezalogowani mogą swobodnie przeglądać zawartość zebranych danych, ale bez możliwości edycji wpisów.
5. Administratorzy mogą zarówno przeglądać zawartość jak i ją edytować (operacje CRUD na wynikach i seriach).
6. Proszę zwrócić uwagę na definicję przedziału min/max dla serii – wartości te powinny być wykorzystywane przez aplikację do walidacji danych wprowadzanych przez użytkownika (wyniki spoza przedziału zdefiniowanego dla danej serii należy odrzucać i komunikować o tym użytkownika).
7. Zadbać o elementy UX (User Experience), np. poprzez umożliwienie zatwierdzania nowych rekordów poprzez wcisnięcie klawisza Enter, walidację wartości wyników po stronie frontendu itp.
8. Każdy użytkownik może wydrukować skonfigurowany wykres wraz z danymi w formie tabelarycznej zaprezentowane w czytelny sposób, tj. bez elementów interfejsu użytkownika (przycisków, odnośników itp.).
9. Zadbać o właściwe przechowywanie hasła dostępowego i możliwość jego zmiany przez zalogowanego użytkownika.
10. Zadbać o responsywność interfejsu użytkownika aplikacji.
11. Kliknięcie w dany rekord w tabeli powinno podświetlać (bądź w inny sposób graficznie wyróżniać) odpowiadający mu punkt na wykresie.

Interfejs aplikacji powinien być przejrzysty – zastosować style CSS3 do formatowania wyglądu elementów.

Wymagania techniczne:

1. Technologia implementacji backendu (do wyboru): PHP, Node.js, ASP.NET MVC, Python.
2. Frontend zaimplementowany w stylu SPA (Single Page Application) z wykorzystaniem wybranego frameworka, np. React, Angular, Vue, itp.
3. Komunikacja front-back przez poprawnie zdefiniowane i udokumentowane REST API zaimplementowane na poziomie dojrzałości 2.3. Dokumentacja API przygotowana w Swaggerze lub czytelnie opisane samodzielnie wszystkie endopointy (**10 pkt**).
4. (backend) Zaprojektuj strukturę bazy danych adekwatną do postawionego problemu: dobierz odpowiednie typy pól, zaplanuj relacje pomiędzy tabelami, zastosuj klucze obce itp. (**5 pkt**)
5. (frontend) Interfejs aplikacji zaprojektuj zgodnie z koncepcją RWD (Responsive Web Design) (**5 pkt**).
6. Drukowanie powinno być zrealizowane jako ten sam widok, co zwykły widok UI aplikacji, modyfikowany wyłącznie przez odpowiednio skonfigurowane zapytania medialne w CSS, które ukryją kontrolki sterujące aplikacją podczas wydruku. (**5 pkt**)
7. Zaadresuj w projekcie wybrane przez siebie 3 różne aspekty dotyczące bezpieczeństwa aplikacji (**5 pkt**)

<https://leia.pw.edu.pl/mod/book/view.php?id=1249>

np.:

- a. Poprawny mechanizm uwierzytelniania i kontroli sesji,
 - b. Zabezpieczenie przed wstrzykiwaniem kodu SQL,
 - c. Przechowywanie haseł użytkowników w odpowiedni sposób (funkcje skrótu).
8. Gotową aplikację udostępnij na publicznie dostępnym serwerze i zadbaj o jej dostępność do momentu wystawienia oceny za projekt (**10 pkt**).
 9. Przygotuj kompletną dokumentację projektu pozwalającą uruchomić aplikację lokalnie. Na platformę Leia załaduj komplet plików. Zadbaj o przykładowe dane umożliwiające szybkie zapoznanie się z funkcjonalnością aplikacji (kilka serii przykładowych wyników, każda przynajmniej po kilkanaście punktów). (**10 pkt**)

Uzyskana ocena punktowa może zostać obniżona w szczególności za niezrealizowanie wymaganych funkcjonalności, niestabilność działania aplikacji, brak poprawnych walidacji itp.

Dodatkowe punkty (do 10 pkt) mogą zostać przyznane za szczególnie dopracowany interfejs aplikacji, realizację rozszerzonej funkcjonalności itp.

Aplikację należy umieścić na dowolnym publicznie dostępnym serwerze w sieci Internet i zadbać o jej dostępność do czasu wystawienia oceny. Można do tego wykorzystać np. zasoby prywatne (jeśli ktoś ma takie w dyspozycji), darmowe hostingi (np. cba.pl) lub usługi chmurowe (Azure, AWS, GCP itp.).

Rozwiązanie należy załadować na platformę LeIA w wyznaczonym terminie i powinno ono zawierać:

- kody źródłowe zaimplementowanej aplikacji (wraz z kodami ewentualnych bibliotek zewnętrznych lub konfigurację pozwalającą automatycznie pobrać niezbędne zależności),
- skrypty SQL tworzące bazę danych zawierającą tabele z przykładową zawartością pozwalającą na zapoznanie się z funkcjonalnością aplikacji,
- dokumentację zaimplementowanego API (może być wygenerowana automatycznie za pomocą odpowiedniego narzędzia, np. Swagger),
- dokumentację projektu (w postaci dokumentu PDF) zawierającą:

- opis pozwalający lokalnie uruchomić dostarczoną aplikację (w tym wersje wykorzystanych narzędzi, polecenia niezbędne do uruchomienia aplikacji, działające dane logowania),
- adres URL serwera, na którym umieszczono aplikację,
- opis wybranej struktury bazy danych (znaczenie poszczególnych tabel, ich pól, relacji itp.) wraz z diagramem ERD,
- opisu kodu stworzonej aplikacji (krótka informacja wskazująca, za co odpowiadają kluczowe funkcje/pliki/klasy),
- zrzuty ekranu demonstrujące zaimplementowane funkcjonalności.