

ZAI - Projekt 1 /PHP/

Zaimplementuj prosty system „pamiętnikowy” pozwalający odnotowywać na osi czasu wydarzenia mające postać odcinków czasu oraz prezentować je w estetyczny sposób. Potencjalne zastosowanie możesz wyobrazić sobie jako np. prezentacja dla klientów firmy szczegółowych informacji z historii jej rozwoju. Może to być także narzędzie o charakterze intranetowym, w którym odnotowywane są wydarzenia z życia danej społeczności. Potencjalnych zastosowań takiej aplikacji może być wiele – można zaproponować własne zastosowanie.

Specyfikacja funkcjonalna aplikacji (minimalne wymagania):

1. Wydarzenie rozumiemy jako odcinek na osi czasu charakteryzujący się następującymi cechami:
 1. nazwa wydarzenia,
 2. daty rozpoczęcia i zakończenia wydarzenia (dzień, miesiąc, rok),
 3. opis tekstowy,
 4. ilustracja graficzna,
 5. kategoria wydarzenia.
2. Kategorie pozwalają powiązać ze sobą podobne wydarzenia i np. wyświetlić je w sposób charakterystyczny. Typ powinien mieć edytowalną nazwę oraz konfigurowalną jakąś cechę graficzną (np. kolor lub/i ikonę).
3. Są dwie grupy użytkowników: czytelnicy (niezalogowani) i administratorzy (zalogowany).
4. Użytkownicy niezalogowani mogą swobodnie przeglądać zawartość pamiętnika, ale bez możliwości edycji wpisów.
5. Administratorzy mogą zarówno przeglądać zawartość jak i ją edytować (operacje CRUD na wydarzeniach i kategoriach).
6. Każdy użytkownik może wydrukować pełną zawartość pamiętnika w czytelny sposób. Drukowanie powinno być zrealizowane przez odpowiednio przygotowany za pomocą CSS widok pozbawiony kontrolek sterujących aplikacją.
7. Zadbać o właściwe przechowywanie hasła dostępowego i możliwość jego zmiany przez zalogowanego użytkownika.
8. Zadbać o estetyczną formę prezentacji pamiętnika. Można skorzystać z bibliotek JS, które ułatwiają oznaczanie zdarzeń na osi czasu, można spróbować samodzielnej implementacji z użyciem odpowiednio przygotowanego arkusza CSS i elementów języka HTML5.
9. Kliknięcie w dane wydarzenie na osi czasu powinno wyświetlać pełen opis danego elementu.

Inspiracje dla implementacji osi czasu:

<https://github.com/ka215/jquery.timeline>

<https://timeline.knightlab.com/>

<https://www.cssscript.com/demo/responsive-animated-timeline-javascript-css3/>

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

Wymagania techniczne:

- proszę wykorzystać technologie PHP, MariaDB / MySQL, HTML, JavaScript, CSS we współczesnych wersjach (w miarę możliwości),
- zaprojektować relacyjną bazę danych adekwatną do postawionego problemu: dobrać odpowiednie typy pól, zaplanować zależności pomiędzy tabelami, zastosować klucze obce,
- zachęcam do korzystania z frameworków, ale nie ma takiego wymogu; tym samym np. mechanizm uwierzytelniania użytkownika może zostać stworzony samodzielnie (można wykorzystywać materiały dołączone do kursu); użycie frameworka upraszcza pewne

kwestie implementacyjne, ale wymaga zrozumienia i poznania samego frameworka – wybór leży w gestii Studenta,

- aplikacja powinna zapewniać choćby w minimalnym stopniu bezpieczeństwo przed włamaniami:
 - zastosować zabezpieczenie przed SQL-Injection,
 - hasła użytkowników muszą być przechowywane w bazie w formie hashy.

Stronę należy umieścić na dowolnym publicznie dostępnym serwerze w sieci Internet. Darmowe lub niedrogo hostingi oferują m.in.:

<https://cba.pl/>

<https://www.ugu.pl/>

<https://azure.microsoft.com/en-us/products/virtual-machines/>

<https://aws.amazon.com/>

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

- kody źródłowe zaimplementowanego systemu (wraz z kodami ewentualnych bibliotek zewnętrznych),
- skrypty tworzące bazę danych zawierającą tabele z przykładową zawartością pozwalającą na zapoznanie się z funkcjonalnością aplikacji,
- dokumentację projektu składającą się z:
 - opisu konfiguracji serwera i uruchomienia witryny (w tym działające dane logowania),
 - adresu URL serwera, na którym umieszczono stronę,
 - opis wybranej struktury bazy danych (znaczenie poszczególnych tabel, ich pól, relacji itp.),
 - opisu kodu stworzonej aplikacji (krótka informacja wskazująca, za co odpowiadają kluczowe funkcje/pliki/klasy),
 - zrzuty ekranu demonstrujące zaimplementowane funkcjonalności.

Cechy i funkcjonalności wpływające na ocenę (10 p. + 2 pkt bonus):

- zapewnienie żądanej funkcjonalności (wymagane, stanowi mnożnik w skali 0...1 dla sumy pozostałych punktów – po 0.1 za każdy z punktów),
- wykorzystanie żądanych technologii (2 p.),
- poprawny mechanizm uwierzytelniania (1 p.),
- sensowny projekt bazy danych z wykorzystaniem kluczy obcych (2 p.),
- spełnienie wymagań dotyczących bezpieczeństwa aplikacji (1 p.)
- udostępnienie strony na serwerze (2 p.)
- dokumentacja zawierająca wymagane elementy i zredagowana w sposób właściwy dla formalnych dokumentów technicznych (2 p.)
- dodatkowe 2 punkty mogą zostać przyznane dla prac wyróżniających się pod względem estetyki interfejsu aplikacji czy zastosowanych rozwiązań.