

Programowanie aplikacyjne

-

specyfikacja wymagań projektu

Mikołaj Rożek, Marcin Lisowski, Sofiya Nasiakaila, Maksymilian Bilski

Spis treści

1. Wprowadzenie	2
1.1. Cel projektu	2
2. Wymagania funkcjonalne	2
2.1. Rejestracja i logowanie	2
3. Zadania programistyczne	2
3.1. Przeglądanie zadań	2
3.2. Rozwiązywanie zadań	2
3.2.1. Edytor kodu	2
3.2.2. Sprawdzanie poprawności zadań	2
3.2.3. Porównywanie jakości rozwiązania z innymi użytkownikami	2
3.3. Profil użytkownika	2
3.3.1. Statystyki	2
3.3.2. Historia rozwiązań	2
4. Wymagania нефункционалне	2
4.1. Wydajność	2
4.2. Bezpieczeństwo	3
4.3. Kompatybilność	3
4.4. Skalowalność	3
4.5. Wsparcie Języków Programowania	3
4.6. Archiwizacja Danych	3
5. Wymagania techniczne	3

1. Wprowadzenie

1.1. Cel projektu

Celem projektu jest stworzenie aplikacji webowej, która umożliwia użytkownikom rozwiązywanie zadań programistycznych, głównie algorytmicznych. Użytkownik ma mieć możliwość sprawdzania jakości przesłanych rozwiązań na tle innych użytkowników, śledzenia swojej aktywności, zyskiwania punktów za rozwiązanie zadania oraz brania udziału w turniejach programistycznych.

2. Wymagania funkcjonalne

2.1. Rejestracja i logowanie

Użytkownik może utworzyć konto, podając unikalny adres e-mail oraz hasło. Logowanie przebiega w analogiczny sposób.

3. Zadania programistyczne

3.1. Przeglądanie zadań

Zadania są podzielone na kategorie, takie jak algorytmy, struktury danych, bazy danych itp. Użytkownik może przeglądać zadania w ramach wybranej kategorii oraz sortować je według poziomu trudności, popularności czy daty dodania. Filtry pozwalają zawęzić widok zadań na podstawie struktur danych, tagów.

3.2. Rozwiązywanie zadań

3.2.1. Edytor kodu

Użytkownik ma dostęp do edytora kodu online, który obsługuje różne języki programowania. Uwzględniona będzie funkcjonalność podpowiedzi kodu i kolorowania składni.

3.2.2. Sprawdzanie poprawności zadań

Po wysłaniu rozwiązania zadania, jego poprawność będzie sprawdzana na podstawie gotowych testów jednostkowych. Użytkownik otrzyma wiadomość zwrotną dotyczącą poprawności swojego rozwiązania, razem z czasem jego wykonania.

3.2.3. Porównywanie jakości rozwiązania z innymi użytkownikami

Razem z wiadomością zwrotną dotyczącą poprawności rozwiązania, użytkownik zobaczy szybkość swojego kodu, oraz jak plasuje się jego kod względem innych użytkowników.

3.3. Profil użytkownika

3.3.1. Statystyki

Użytkownik ma dostęp do statystyk swojego postępu w rozwiązywaniu zadań. W ramach tego może zobaczyć ile zadań na danym poziomie trudności rozwiązał, czasu spędzonego na platformie.

3.3.2. Historia rozwiązań

4. Wymagania niefunkcjonalne

4.1. Wydajność

Czas ładowania strony i działania funkcji powinien być akceptowalny, nawet przy dużej liczbie użytkowników.

4.2. Bezpieczeństwo

Bezpieczne przechowywanie haseł użytkowników. Zabezpieczenie przed atakami typu SQL injection i XSS.

4.3. Kompatybilność

Aplikacja powinna być dostępna na różnych przeglądarkach (Chrome, Firefox, Safari).

4.4. Skalowalność

System powinien być łatwo skalowalny, aby obsłużyć wzrost liczby użytkowników.

4.5. Wsparcie Języków Programowania

Aplikacja powinna obsługiwać co najmniej kilku popularnych języków programowania, takich jak Java, Python, JavaScript.

4.6. Archiwizacja Danych

Tworzenie kopii zapasowych danych użytkowników w celu zapewnienia bezpieczeństwa.

5. Wymagania techniczne