

Chatbot AI – Zadanie Rekrutacyjne

Twoim zadaniem będzie stworzenie prototypu chatbota AI, który symuluje prostego asystenta. Głównym celem jest zbudowanie kompletnej aplikacji, w której można łatwo zaimplementować prawdziwą integrację z modelem AI w przyszłości. Prototyp powinien działać z podstawowym zestawem funkcjonalności, a odpowiedzi chatbota mają być generowane losowo lub według prostych reguł, wystarczy "Lorem ipsum...".

Wymagania funkcjonalne

- **Historia czatu:** Chatbot powinien wyświetlać historię rozmowy, z rozróżnieniem wiadomości użytkownika i odpowiedzi chatbota.
- **Ocena odpowiedzi:** Każda odpowiedź chatbota powinna mieć możliwość oceny (np. przyciskami "👍" i "👎"). Użytkownik powinien móc zmieniać ocenę odpowiedzi w dowolnym momencie.
- **Generowanie odpowiedzi:** Odpowiedzi chatbota powinny być wyświetlane stopniowo, symulując proces "pisanie" (litera po literze lub linia po linii, tak jak to wygląda w innych aplikacjach tego typu). Długość odpowiedzi powinna być zróżnicowana, od krótkich, zawierających 1-2 zdania, przez średnie obejmujące 3-5 zdań, aż po bardzo długie, składające się z kilku akapitów.
- **Anulowanie generowania odpowiedzi:** Powinna istnieć możliwość anulowania generowanej odpowiedzi w trakcie jej wyświetlania. W takim przypadku, generowanie tekstu powinno zostać natychmiast przerwane, a system powinien zapisać i wyświetlić tylko tę część odpowiedzi, która została wygenerowana do momentu anulowania.
- **Przechowywanie danych:** Wszystkie informacje, takie jak historia czatu i oceny odpowiedzi, powinny być zapisywane w bazie danych. Nie jest wymagana żadna autoryzacja użytkowników.

Wymagania techniczne

1. **Frontend:** Wykorzystaj framework Angular w dowolnej wersji. Stylizację interfejsu oprzyj na Angular Material – minimalna estetyka i domyślne komponenty są wystarczające.
2. **Backend:** Stwórz API w ASP.NET Core Web API, użyj SQL Server jako bazy danych, oraz Entity Framework Core jako ORM. Możesz też wykorzystać dowolne inne biblioteki. Mile widziane będzie wykorzystanie biblioteki MediatR. Zaprojektuj API w sposób umożliwiający łatwe rozszerzenie o integrację z prawdziwym modelem AI.
3. **Stylizacja i UX:** Skup się na funkcjonalności. Używaj domyślnych stylów i komponentów Angular Material. Aplikacja nie musi być responsywna, ale powinna działać poprawnie w popularnych przeglądarkach desktopowych (Chrome, Firefox, Safari).
4. **Korzystanie z narzędzi AI:** Możesz wspierać się dowolnymi narzędziami AI podczas pisania kodu, pod warunkiem, że rozumiesz i potrafisz wyjaśnić wszystkie fragmenty wygenerowanego rozwiązania.