

Wymagana dokumentacja dla projektu "AiDoc"

Jakub Możejko, Kacper Nowak, Krzysztof Wolski, AiB, N42-05

1. Charakterystyka oprogramowania

Nazwa skrócona:

AiDoc

Nazwa pełna:

AiDoc – aplikacja internetowa do udzielania porad zdrowotnych na podstawie danych ze smartwatcha.

Krótki opis ze wskazaniem celów:

AiDoc to aplikacja internetowa, która analizuje dane dostarczone przez smartwatch i udziela porad zdrowotnych dotyczących aktywności fizycznej, diety oraz ogólnego zdrowego trybu życia. Aplikacja generuje również tygodniowe raporty na podstawie kondycji zdrowia i ciała użytkownika, które są wysyłane na jego adres e-mail.

2. Prawa autorskie

Autor

Jakub Możejko, Kacper Nowak, Krzysztof Wolski

Warunki licencyjne do oprogramowania wytworzonego przez grupę

MIT License

Copyright © 2024, Jakub Możejko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3. Specyfikacja wymagań - Epics/User Stories

Typy użytkownika:

Zalogowany użytkownik, niezalogowany użytkownik

Grupa funkcjonalności dla projektu "AiDoc":

Epic 1: Zarządzanie kontem użytkownika

Ten epik obejmuje wszystkie funkcje związane z rejestracją, logowaniem, wylogowywaniem oraz zarządzaniem profilem użytkownika. Funkcjonalności te zapewniają użytkownikom dostęp do aplikacji i pozwalają na personalizację ich konta.

User Stories:

1. Rejestracja użytkownika

- Jako niezalogowany użytkownik chcę zarejestrować się w systemie, aby móc korzystać z pełnej funkcjonalności aplikacji.
- Priorytet: 1 - wymagane

2. Logowanie użytkownika

- Jako niezalogowany użytkownik chcę zalogować się do systemu, aby uzyskać dostęp do moich wcześniejszych sesji i raportów zdrowotnych.
- Priorytet: 1 - wymagane

3. Wylogowanie użytkownika

- Jako zalogowany użytkownik chcę mieć możliwość wylogowania się z systemu, aby zakończyć sesję w bezpieczny sposób.
- Priorytet: 1 - wymagane

4. Zarządzanie profilem użytkownika

- Jako zalogowany użytkownik chcę móc aktualizować moje dane osobowe, aby moje konto było zawsze aktualne.
- Priorytet: 2 - przydatne

Epic 2: Interakcje z modelem GPT

Ten epik obejmuje wszystkie funkcje związane z komunikacją i przetwarzaniem danych przez model GPT, w celu dostarczania użytkownikowi odpowiedzi i porad zdrowotnych. Umożliwia użytkownikom uzyskiwanie spersonalizowanych informacji na podstawie wprowadzonych danych zdrowotnych.

User Stories:

1. Wprowadzanie danych zdrowotnych

- Jako zalogowany użytkownik chcę wprowadzić dane zdrowotne z mojego smartwatcha, aby uzyskać porady zdrowotne.

- Priorytet: 1 - wymagane

2. Otrzymywanie porad zdrowotnych

- Jako zalogowany użytkownik chcę otrzymać porady dotyczące aktywności fizycznej i diety na podstawie danych zdrowotnych, aby poprawić mój styl życia.

- Priorytet: 1 - wymagane

3. Historia interakcji z modelem GPT

- Jako zalogowany użytkownik chcę zapisać historię moich interakcji z modelem GPT, aby móc do niej wracać w przyszłości.

- Priorytet: 2 - przydatne

4. Przeglądanie historii interakcji z modelem GPT

- Jako zalogowany użytkownik chcę przeglądać historię moich interakcji z modelem GPT, aby móc analizować wcześniejsze odpowiedzi.

- Priorytet: 2 - przydatne

Epic 3: Analiza danych zdrowotnych

Ten epic obejmuje wszystkie funkcje związane z analizą danych zdrowotnych użytkownika, aby dostarczyć mu spersonalizowane porady i zalecenia zdrowotne. Funkcjonalności te pomagają użytkownikom monitorować i poprawiać ich zdrowie oraz kondycję fizyczną.

User Stories:

1. Analiza aktywności fizycznej

- Jako zalogowany użytkownik chcę otrzymywać porady dotyczące mojej aktywności fizycznej na podstawie danych z smartwatcha, aby poprawić moją kondycję.

- Priorytet: 1 - wymagane

2. Analiza diety

- Jako zalogowany użytkownik chcę otrzymywać porady dotyczące mojej diety na podstawie danych z smartwatcha, aby prowadzić zdrowszy tryb życia.

- Priorytet: 1 - wymagane

3. Monitorowanie parametrów zdrowotnych

- Jako zalogowany użytkownik chcę, aby system monitorował moje parametry zdrowotne, takie jak tętno, ilość snu, itp., aby otrzymywać wskazówki zdrowotne.

- Priorytet: 2 - przydatne

Epic 4: Generowanie raportów

Ten epic obejmuje wszystkie funkcje związane z generowaniem i wysyłaniem tygodniowych raportów zdrowotnych użytkownikom na podstawie ich danych zdrowotnych.

Funkcjonalności te pomagają użytkownikom śledzić swoje postępy i uzyskiwać regularne podsumowania swojego zdrowia.

User Stories:

1. Generowanie tygodniowych raportów zdrowotnych

- Jako zalogowany użytkownik chcę, aby system generował tygodniowe raporty zdrowotne, aby móc monitorować moje postępy.

- Priorytet: 2 - przydatne

2. Wysyłka raportów na email

- Jako zalogowany użytkownik chcę otrzymywać tygodniowe raporty zdrowotne na mój adres email, aby mieć łatwy dostęp do moich danych.

- Priorytet: 2 - przydatne

3. Personalizacja raportów zdrowotnych

- Jako zalogowany użytkownik chcę, aby raporty zdrowotne były spersonalizowane na podstawie moich danych i celów zdrowotnych, aby lepiej spełniały moje potrzeby.

- Priorytet: 2 - przydatne

4. Testy

Scenariusze testów

1. Scenariusz rejestracji użytkownika:

- Działanie użytkownika: Użytkownik wprowadza adres email i hasło, a następnie klika "Zarejestruj się".

- Odpowiedź aplikacji: System rejestruje użytkownika i przekierowuje go na stronę logowania.

- [Wymaganie funkcjonalne: 1]

2. Scenariusz logowania użytkownika

- Działanie użytkownika: Użytkownik wprowadza adres email i hasło, a następnie klika "Zaloguj się".

- Odpowiedź aplikacji: System loguje użytkownika i przekierowuje go na stronę główną.

- [Wymaganie funkcjonalne: 2]

3. Scenariusz wylogowania użytkownika:

- Działanie użytkownika: Zalogowany użytkownik klika "Wyloguj się".

- Odpowiedź aplikacji: System wylogowuje użytkownika i przekierowuje go na stronę główną.

- [Wymaganie funkcjonalne: 3]

4. Scenariusz wprowadzenia danych zdrowotnych:

- Działanie użytkownika: Zalogowany użytkownik wprowadza dane zdrowotne z smartwatcha i klika "Wyślij".
- Odpowiedź aplikacji: System analizuje dane i wyświetla odpowiednie porady zdrowotne.
- [Wymaganie funkcjonalne: 4]

5. Scenariusz otrzymywania porad zdrowotnych:

- Działanie użytkownika: Zalogowany użytkownik otrzymuje porady dotyczące aktywności fizycznej i diety na podstawie wprowadzonych danych.
- Odpowiedź aplikacji: System wyświetla porady zdrowotne.
- [Wymaganie funkcjonalne: 5]

6. Scenariusz generowania tygodniowych raportów:

- Działanie użytkownika: System automatycznie generuje tygodniowe raporty zdrowotne i wysyła je na adres email użytkownika.
- Odpowiedź aplikacji: Użytkownik otrzymuje raporty zdrowotne na swój email.
- [Wymaganie funkcjonalne: 6]

Sprawozdanie z wykonania scenariuszy testów:

Do każdego z powyższych scenariuszy testów dołączone będą szczegółowe raporty, zawierające wyniki przeprowadzonych testów, wykryte błędy oraz podjęte działania korygujące.

5. Architektura oprogramowania

Technologie i narzędzia:

- Język programowania: Python
- Framework backendowy: Flask
- Frontend: HTML, CSS, JavaScript
- Model GPT: Wykorzystanie API do modelu GPT
- Baza danych: MongoDB (przechowywanie danych zdrowotnych użytkowników)
- Środowisko programistyczne: Visual Studio Code
- Komunikacja między warstwami: RESTful API
- Bezpieczeństwo: Szyfrowanie danych użytkowników, w tym haseł
- Raportowanie: Generowanie raportów w formacie PDF i wysyłka przez email

Opis architektury:

Frontend

- Tworzony przy użyciu HTML, CSS, JavaScript.
- Interfejs użytkownika do wprowadzania danych zdrowotnych, przeglądania porad i zarządzania kontem.

Backend

- Framework Flask do obsługi logiki aplikacji.
- Implementacja endpointów RESTful API do komunikacji z frontendem i zewnętrznym API modelu GPT.

Baza danych

- MongoDB do przechowywania danych użytkowników oraz historii interakcji.

Moduł integracji z API GPT

- Komunikacja z modelem GPT za pomocą dedykowanego API.
- Przetwarzanie i analiza danych zdrowotnych użytkowników w celu generowania spersonalizowanych porad.

Moduł raportowania

- Automatyczne generowanie tygodniowych raportów zdrowotnych w formacie PDF.
- Wysyłka raportów na adres email użytkowników.

Bezpieczeństwo

- Szyfrowanie danych wrażliwych, takich jak hasła użytkowników.
- Regularne kopie zapasowe bazy danych.
- Ochrona przed atakami typu SQL injection i XSS.

Skalowalność

- Architektura umożliwiająca łatwe dodawanie nowych funkcjonalności.
- Możliwość skalowania aplikacji zarówno na poziomie backendu, jak i bazy danych.