169506 Piotr Sparzak
169509 Mikołaj Symołon
166362 Remigiusz Zień
169432 Dawid Kaczyński

1. Ogólne Kryteria

1.1 Kod źródłowy

Kod jest zgodny z ustalonymi standardami programistycznymi.

Wszystkie zmiany są zarządzane w systemie kontroli wersji (GitHub/GitLab) poprzez Pull Request.

Każdy Pull Request musi być zaakceptowany przez co najmniej jednego innego programistę przed scaleniem do głównej gałęzi.

1.2 Testy jednostkowe

Kod posiada testy jednostkowe z pokryciem minimum 90% dla kluczowych modułów.

Testy są uruchamiane automatycznie.

1.3 Testy integracyjne

Testy integracyjne potwierdzają poprawność współpracy modułów (np. komunikacja API z bazą danych, uwierzytelnianie użytkowników).

Testy są przeprowadzane narzędziami takimi jak Postman (dla API) oraz Selenium (dla interfejsu użytkownika).

1.4 Wdrożenie na środowisko testowe

Kod jest wdrażany na środowisko testowe za pomocą CI/CD pipeline w GitHub Actions lub GitLab CI.

Test Manager dokonał weryfikacji funkcjonalnej aplikacji.

2. Kryteria dla Poszczególnych Modułów

2.1 Frontend

Interfejs użytkownika jest zgodny z makietami UI/UX.

Widoki są responsywne i poprawnie działają na urządzeniach mobilnych i desktopowych (Chrome, Firefox, Edge, Safari).

Kluczowe funkcje działają poprawnie:

Rejestracja i logowanie: formularz rejestracji oraz logowania działa, a użytkownicy otrzymują e-mail weryfikacyjny.

Dodawanie opinii: użytkownik może dodawać oceny restauracji gwiazdki 1-5 oraz recenzje tekstowe.

Przegląd restauracji: można filtrować, sortować i wyszukiwać restauracje według kategorii, ocen i lokalizacji.

2.2 Backend

API obsługuje wszystkie wymagane zapytania i zwraca poprawne odpowiedzi:

Przetestowane narzędziem Postman (GET, POST, PUT, DELETE).

System logowania i rejestracji działa zgodnie z wymaganiami bezpieczeństwa:

Hasła przechowywane w formie zaszyfrowanej.

Mechanizm resetowania hasła poprzez e-mail działa poprawnie.

Testy wydajnościowe wykazały zdolność obsługi przewidywanego ruchu.

2.3 Integracja AI/ML

Rekomendacje restauracji działają z dokładnością minimum 85%.

Czas odpowiedzi systemu rekomendacji nie przekracza 2 sekund.

2.4 Baza Danych

Struktura bazy zoptymalizowana pod kątem wydajności i zgodności z RODO.

Przeprowadzono migrację danych za pomocą narzędzi takich jak Alembic (Python).

Regularne kopie zapasowe (co 24h).

2.5 Bezpieczeństwo

Dane użytkowników są szyfrowane zgodnie z najlepszymi praktykami.

Przeprowadzono testy penetracyjne narzędziami OWASP ZAP i Burp Suite.

Autoryzacja i uwierzytelnianie działają zgodnie z JWT (JSON Web Token).

3.1 Kryteria Akceptacji dla Przesyłania Pliku

1. **Sprawdzenie istnienia pliku na serwerze**
    - Jeśli plik o tej samej nazwie już istnieje na serwerze, system powinien odrzucić przesyłanie lub nadpisać go zgodnie z określoną polityką.
    - Możliwe rozwiązania: nadpisanie, zmiana nazwy pliku, odrzucenie przesyłania.
2. **Weryfikacja formatu pliku**
    - Plik musi być w dozwolonym formacie (np. .jpg, .png, .pdf, .txt itp.).
    - System powinien odrzucić pliki w nie obsługiwanych formatach.
3. **Sprawdzenie rozmiaru pliku**
    - Plik nie może przekraczać maksymalnego dozwolonego rozmiaru (np. 10 MB).
4. **Weryfikacja bezpieczeństwa pliku**
    - Plik nie może zawierać złośliwego oprogramowania (np. wirusów, skryptów wykonywalnych).
    - System powinien skanować plik pod kątem zagrożeń
5. **Sprawdzenie uprawnień użytkownika**
    - Użytkownik musi mieć odpowiednie uprawnienia do przesyłania pliku (np. zalogowany użytkownik).
6. **Integracja z systemem logowania zdarzeń**
    - Każda operacja przesyłania pliku powinna być rejestrowana (np. kto przesłał, kiedy, jaki plik).

3.2 Kryteria Akceptacji dla Zmiany Hasła

1. **Autoryzacja użytkownika**
    - Użytkownik musi być zalogowany lub podać aktualne hasło przed zmianą.
2. **Wymogi dotyczące nowego hasła**
    - Nowe hasło musi spełniać określone wymagania bezpieczeństwa:
        - Minimum 8 znaków
        - Przynajmniej jedna wielka litera
        - Przynajmniej jedna cyfra
        - Przynajmniej jeden znak specjalny
3. **Weryfikacja nowego hasła**
    - Użytkownik musi dwukrotnie wprowadzić nowe hasło, aby uniknąć literówek.
4. **Brak ponownego użycia starego hasła**
    - System powinien uniemożliwić ustawienie hasła, które było używane wcześniej.
5. **Powiadomienie o zmianie hasła**
    - Po pomyślnej zmianie hasła użytkownik otrzymuje powiadomienie e-mail lub SMS.
6. **Bezpieczeństwo operacji**
    - Po zmianie hasła system powinien wymusić ponowne zalogowanie użytkownika na wszystkich urządzeniach.