

ServiceDesk — Dokumentacja

1. Opis Systemu

1.1 Wprowadzenie

ServiceDesk to centralna platforma internetowa stworzona do zarządzania wszystkimi zgłoszeniami serwisowymi, technicznymi i prośbami o usługi w organizacji. Jego głównym zadaniem jest usprawnienie komunikacji i przekształcenie chaotycznych zapytań (e-mail, telefon) w uporządkowane, mieralne zadania (tzw. tickety). System umożliwia użytkownikom zgłaszanie problemów, a administratorom ich obsługę oraz rozwiązywanie problemów.

1.2 Główne Funkcjonalności

Dla Użytkownika (User)

- Rejestracja i Logowanie: Możliwość założenia konta i bezpiecznego logowania (JWT).
- Tworzenie Zgłoszeń: Formularz zgłaszania problemów z wyborem kategorii, priorytetu (domyślnie Medium) i opisem.
- Przeglądanie Zgłoszeń: Lista własnych zgłoszeń z możliwością filtrowania po statusie i wyszukiwania.
- Szczegóły Zgłoszenia: Podgląd szczegółów zgłoszenia i zamieszczonych komentarzy.
- Komentowanie: Możliwość dodawania komentarzy do własnych zgłoszeń w celu doprecyzowania problemu.
- Potwierdzanie Rozwiązania: Możliwość samodzielnego zamknięcia zgłoszenia o statusie "Rozwiązany" (potwierdzenie skuteczności naprawy).
- Odrzucanie Rozwiązania: Możliwość odrzucenia rozwiązania i ponownego otwarcia zgłoszenia (status "Otwarty").

Dla Administratora (Admin)

- Pełny Dostęp: Wgląd we wszystkie zgłoszenia w systemie.
- Zarządzanie Zgłoszeniami: Zmiana statusu (Nowy, Otwarty, Rozwiązany). Status 'Zamknięty' jest zarezerwowany dla potwierdzenia przez użytkownika.
- Statystyki: Dashboard ze statystykami (wykresy kołowe i słupkowe).
- Raporty PDF: Generowanie raportów PDF z listą zgłoszeń.
- Zarządzanie Komentarzami: Moderacja notatek administratora (usuwanie/edycja notatek).
- Notatki Wewnętrzne: Możliwość dodawania notatek widocznych tylko dla administratorów.
- Ograniczenia: Brak możliwości tworzenia nowych zgłoszeń (tylko edycja i zarządzanie).

1.3 Bezpieczeństwo

Automatyczne Wylogowanie: Sesja użytkownika wygasa po zamknięciu przeglądarki (Session Storage).

1.4 Architektura Systemu

System w architekturze klient-serwer:

Backend: ASP.NET Core Web API (.NET 7), Entity Framework Core, SQL Server, Uwierzytelnianie JWT.

Frontend: React (Vite), Axios, CSS Modules / Vanilla CSS (Glassmorphism design), Recharts, QuestPDF (Backend).

1.5 Baza Danych

Tabele: Users, Tickets, Categories, Comments.

Wykorzystano procedury składowane, widoki oraz triggery do optymalizacji operacji i zapewnienia spójności danych.

2. Diagram ERD

Poniżej znajduje się kod diagramu ERD w formacie Mermaid.

```
```mermaid
erDiagram
 Users ||--o{ Tickets : "tworzy"
 Users ||--o{ Comments : "pisze"
 Categories ||--o{ Tickets : "kategoryzuje"
 Tickets ||--o{ Comments : "posiada"

 Users {
 int Id PK
 string Username
 string PasswordHash
 string Role
 }

 Categories {
 int Id PK
 string Name
 }

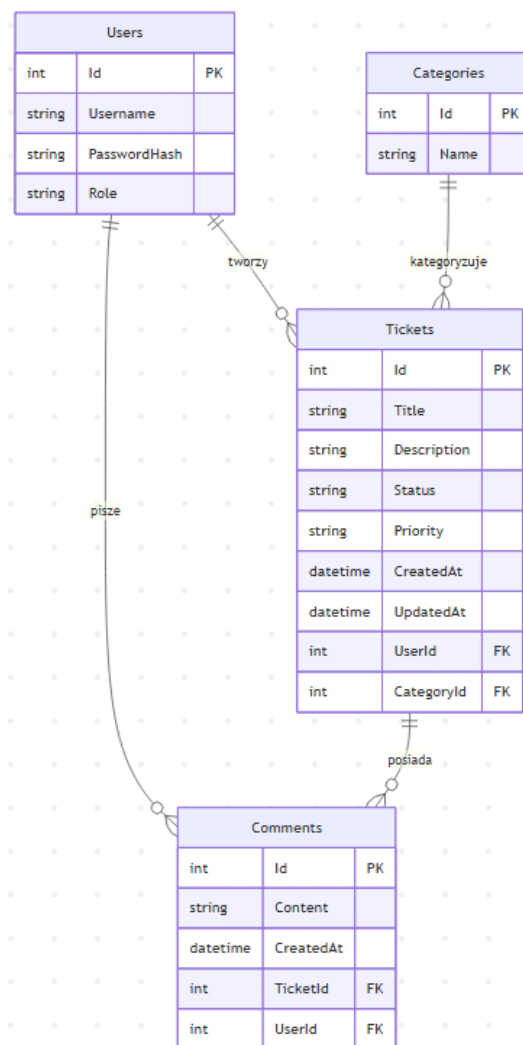
 Tickets {
 int Id PK
 string Title
 string Description
 string Status
 }
```

```

 string Priority
 datetime CreatedAt
 datetime UpdatedAt
 int UserId FK
 int CategoryId FK
 }

 Comments {
 int Id PK
 string Content
 datetime CreatedAt
 bool IsInternal
 int TicketId FK
 int UserId FK
 }
}

```



### 3. Wykaz Procedur Składowanych, Funkcji i Wyzwalaczy

#### Procedury Składowane

1. **GetTicketStatistics** — Zwraca statystyki zgłoszeń, grupując je po statusie i zliczając wystąpienia. Parametry: brak.
2. **GetUserTicketsWithDetails** — Pobiera szczegółową listę zgłoszeń dla konkretnego użytkownika (parametr: @UserId INT).
3. **GetUserTickets** — Pobiera podstawową listę zgłoszeń przypisanych do użytkownika (parametr: @UserId INT).
4. **UpdateTicketStatus** — Aktualizuje status zgłoszenia oraz ustawia UpdatedAt (parametry: @TicketId INT, @NewStatus NVARCHAR).

#### Funkcje

- **dbo.CalculateAverageResolutionTime()** — Oblicza średni czas (w godzinach) potrzebny na rozwiązanie zgłoszenia. Zwraca FLOAT.
- **GetTicketCount** — Zwraca całkowitą liczbę zgłoszeń utworzonych przez danego użytkownika (parametr: @UserId INT). Zwraca INT.

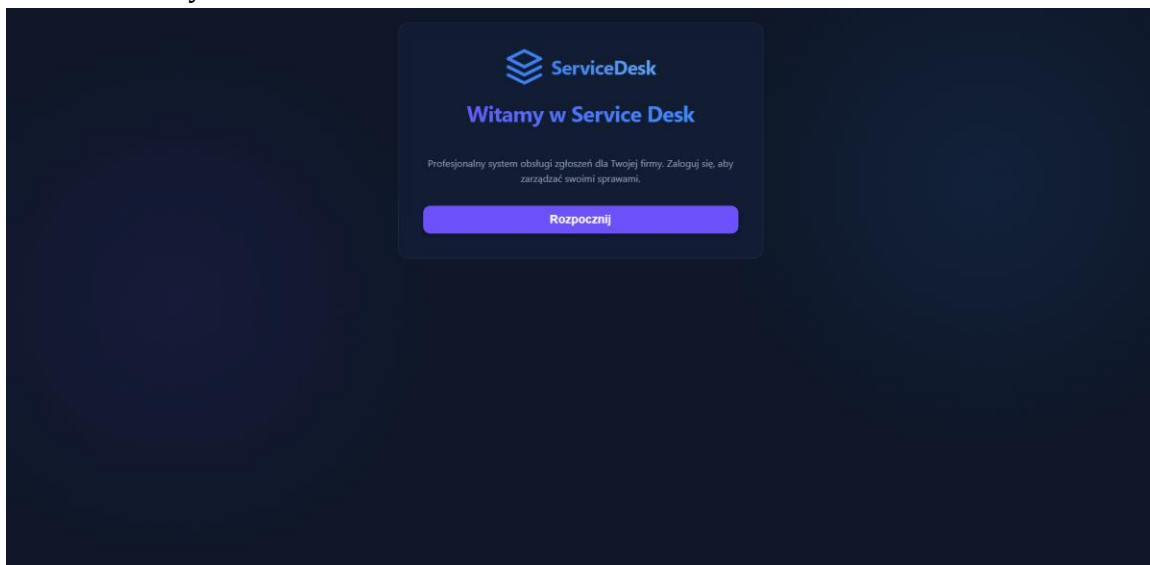
#### Wyzwalacze (Triggers)

- **UpdateTicketModifiedDate (Tickets)** — AFTER UPDATE: Automatycznie aktualizuje kolumnę UpdatedAt na GETDATE().
- **trg\_UpdateTicketTimestamp (Tickets)** — AFTER UPDATE: Mechanizm dublujący funkcjonalność aktualizacji czasu modyfikacji (pozostałość po migracjach).

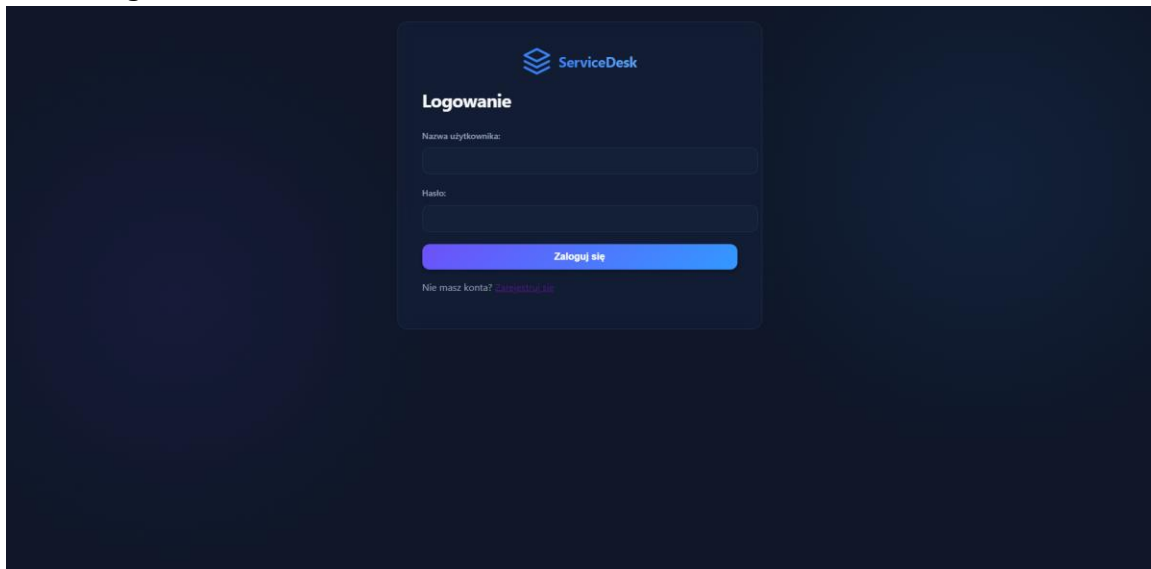
### 4. Prezentacja Systemu (zrzuty ekranu)

W dokumentacji umieść zrzuty ekranu pokazujące działającą aplikację. Sugerowane ujęcia:

-Panel startowy

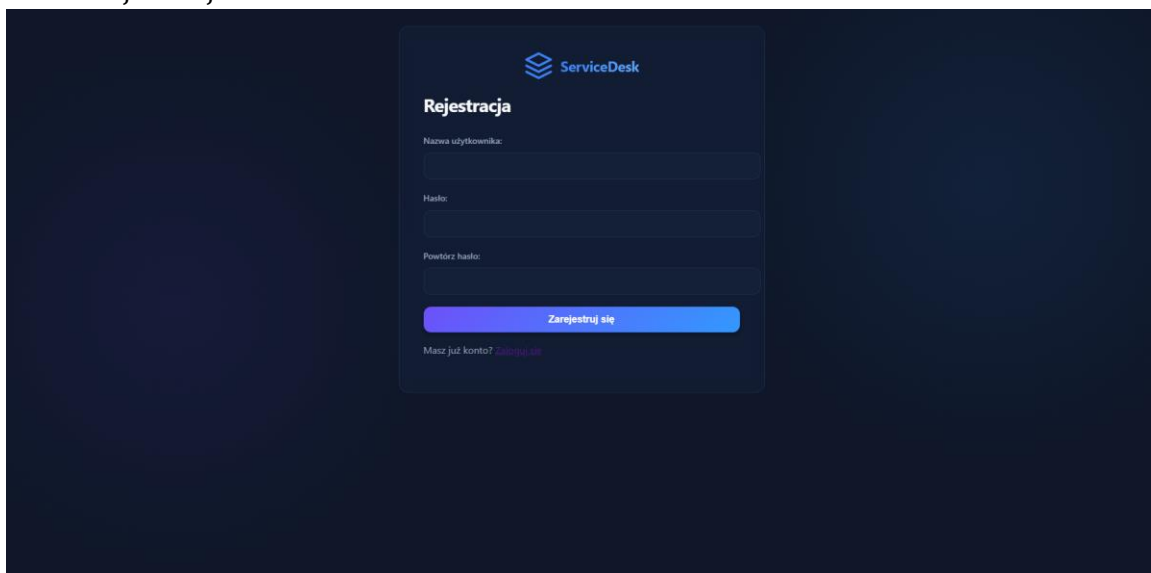


- Ekran logowania



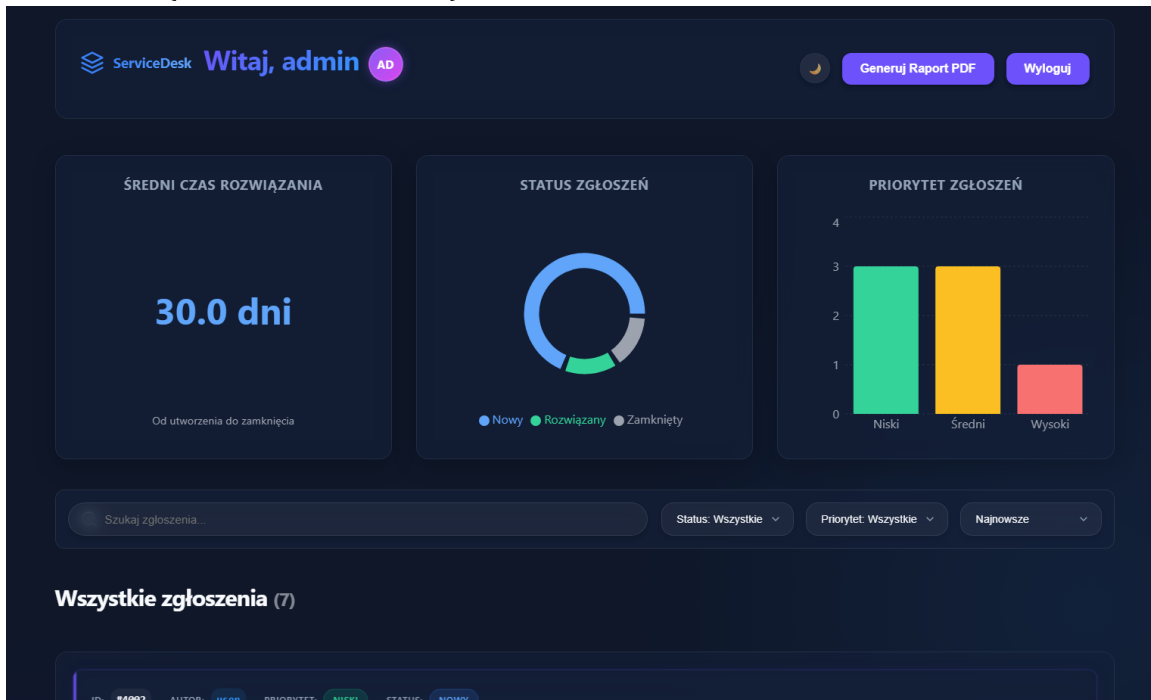
The login screen features a dark blue background with a central white card. At the top of the card is the ServiceDesk logo, consisting of a blue icon of three stacked squares and the text 'ServiceDesk'. Below the logo is the title 'Logowanie' in bold. The form includes two input fields: 'Nazwa użytkownika:' and 'Hasło:'. A blue button with the text 'Zaloguj się' is positioned below the password field. At the bottom of the card, there is a link that reads 'Nie masz konta? [Zarejestruj się](#)'.

-Ekran rejestracji

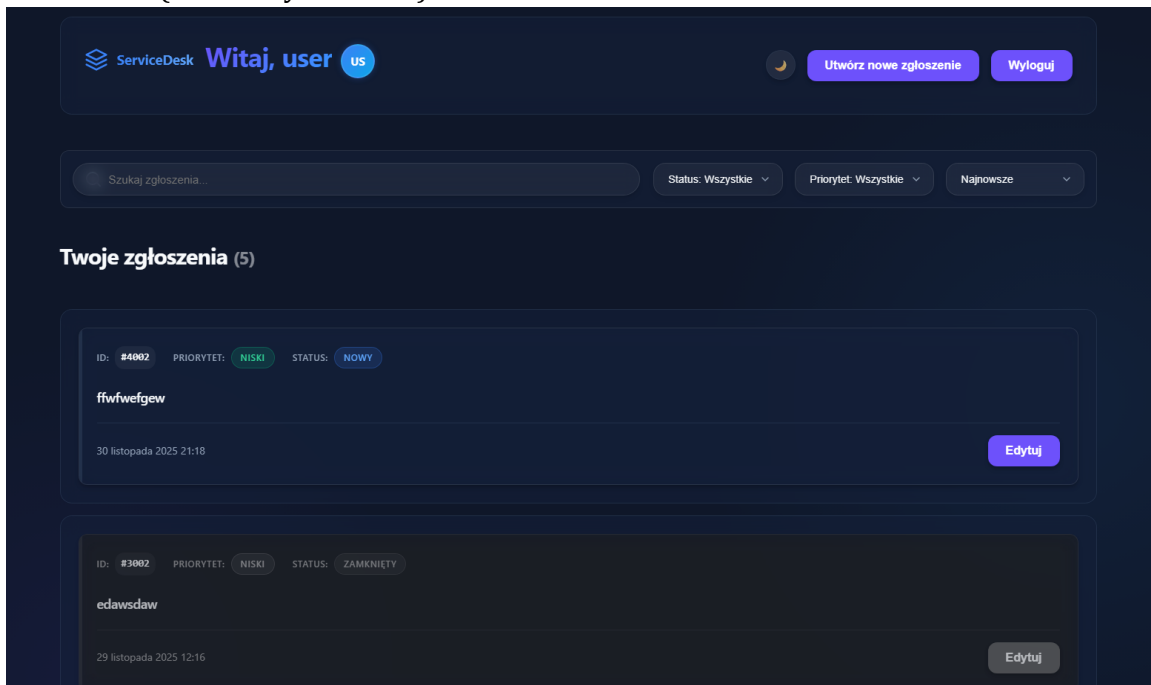


The registration screen has a dark blue background with a central white card. At the top of the card is the ServiceDesk logo, consisting of a blue icon of three stacked squares and the text 'ServiceDesk'. Below the logo is the title 'Rejestracja' in bold. The form includes three input fields: 'Nazwa użytkownika:', 'Hasło:', and 'Powtórz hasło:'. A blue button with the text 'Zarejestruj się' is positioned below the third input field. At the bottom of the card, there is a link that reads 'Masz już konto? [Zaloguj się](#)'.

- Dashboard (widok administratora)



-Dashboard (widok użytkownika)



## - Formularz nowego zgłoszenia

### Utwórz nowe zgłoszenie

Tytuł:

Opis:

Kategoria:

Wybierz kategorię

Priorytet:

Średni

Utwórz Anuluj

## -Edycja zgłoszenia

### Edytuj zgłoszenie

Tytuł:

Opis:

Kategoria:

Inne

Priorytet:


Niski

Zaktualizuj Anuluj

## -Szczegóły zgłoszenia (widok administratora)

ServiceDesk

Szczegóły zgłoszenia #5002



PowrótWyloguj

OTWARTY

WYSOKI3.12.2025, 13:35:06

e21e2edqwrdf

21eq2deqddq2

Autor zgłoszenia: user

Notatki Administratora

admin3.12.2025, 13:53:09

Błąd nie został dalej naprawiony. Zajmę się dalszą analizą systemu

admin3.12.2025, 13:53:09

Błąd nie został dalej naprawiony. Zajmę się dalszą analizą systemu

Dodaj notatkę wewnętrzną...

Dodaj notatkę

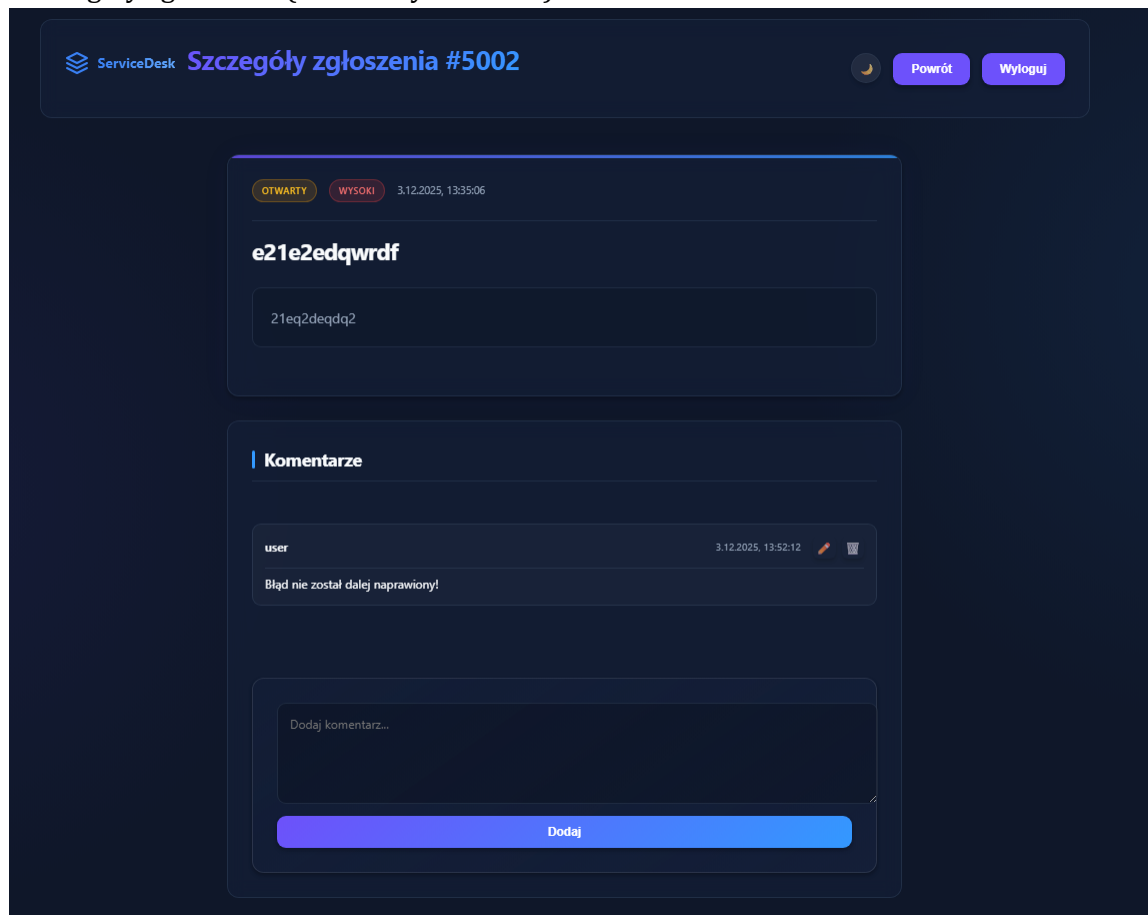
Komentarze użytkownika user

user3.12.2025, 13:52:12

Błąd nie został dalej naprawiony!



-Szczegóły zgłoszenia (widok użytkownika)



## 5. Instrukcja Uruchomienia

### Wymagania

.NET 7 SDK (lub nowszy), Node.js (v18+), SQL Server (LocalDB lub pełna instancja).

### Krok 1: Konfiguracja Bazy Danych

1. Otwórz plik appsettings.json w folderze ServiceDesk (Backend) i dostosuj ConnectionStrings:DefaultConnection.

2. Uruchom migracje, aby utworzyć bazę danych:

```
cd ServiceDesk
```

```
dotnet ef database update
```

### Krok 2: Uruchomienie Backend

```
cd ServiceDesk
```

```
dotnet run
```

API będzie dostępne pod adresem: `http://localhost:5054`

### **Krok 3: Uruchomienie Frontend**

```
cd Frontend
```

```
npm install
```

```
npm run dev
```

Aplikacja będzie dostępna pod adresem: <http://localhost:5173>