Raport 24.10.21 r.

1. Cele
   1. [Security] Przygotowanie repozytorium i narzędzi ciągłego wdrażania w taki sposób, aby wrażliwe dane typu:
      1. Klucze SSH
      2. Hasła i loginy
      3. Adresy IP

Nie były publicznie dostępne.

* 1. [DI] Instalacja i konfiguracja kontenera zależności Autofac po stronie serwerowej aplikacji
  2. [CQRS] Instalacja i konfiguracja biblioteki MediatR w celu utworzenia wzorca Mediatora i CQRS po stronie serwerowej aplikacji
  3. [STATE] Instalacja i konfiguracja biblioteki NgXs w warstwie prezentacji, aby wykorzystać wzorzec zarządzenia stanem.
  4. [MATERIAL] Instalacja biblioteki zawierającej komponenty implementującą podejście Material Design
  5. [AZURE] Przygotowanie potoku GitHub Actions w celu wdrożenia aplikacji warstwy prezentacji w chmurze obliczeniowej Azure
  6. [FRONTEND] Stworzenie podstawowej strutkury projektu warstyw prezentacji
  7. [BACKEND] Stworzenie podstawowej struktury projektu warstwy serwerowej

1. Realizacja
   1. [Security] Kolejny raz nie udało się. Nie znalazłem rozwiązania, które podmieniłoby pliki konfiguracje i pozwoliłby ukryć wrażliwe dane w repozytorium. Możliwe, że będzie potrzebne inne rozwiązanie niż GitHub Secrets z którego próbowałem skorzystać.
   2. [DI] [CQRS] Dodano biliboteki Autofac i MediatR w celu implementacji wzorca CQRS w warstwie serwerowej. Dodatkowo dodano bibliotekę Automapper, aby ułatwić mapowanie pomiędzy modelami.
   3. [State] Zainstalowano bibliotekę NgXS implementującą wzorzec Redux.
   4. [Material] Zainstalowano bibliotekę Angular Material Design
   5. [AZURE] Przygotowano środowisko Azure Application Service oraz skrypt GitHub Actions, które po wypchaniu zmian do main aktualizują wersję aplikacji w chmurze obliczeniowej Azure.
   6. [FRONTEND] Stworzono podstawową strukturę projektu warstwy prezentacji. Dodano bibliotekę Angular Material, aby wykorzystać gotowe kontrolki. Ponadto utworzono podstawowy routing aplikacji oraz widoki rejestracji i logowania.
   7. [BACKEND] Stworzenie podstawowej struktury projektu warstwy serwerowej. W ramach podstawowej struktury utworzono podstawowe projekty, rozszerzenia języka .NET, dodano biblioteki:
      1. Automapper – automatycznie mapowanie typów
      2. Autofac – kontener wstrzykiwania zależności
      3. NUnit – bilbioteka do tworzenia testów jednostkowych
      4. MediatR – bilblioteka implementująca wzorzec CQRS

Ponadto dodano mechanizm który automatycznie rejestruje komponenty w kontenerze wstrzykiwania zależności.