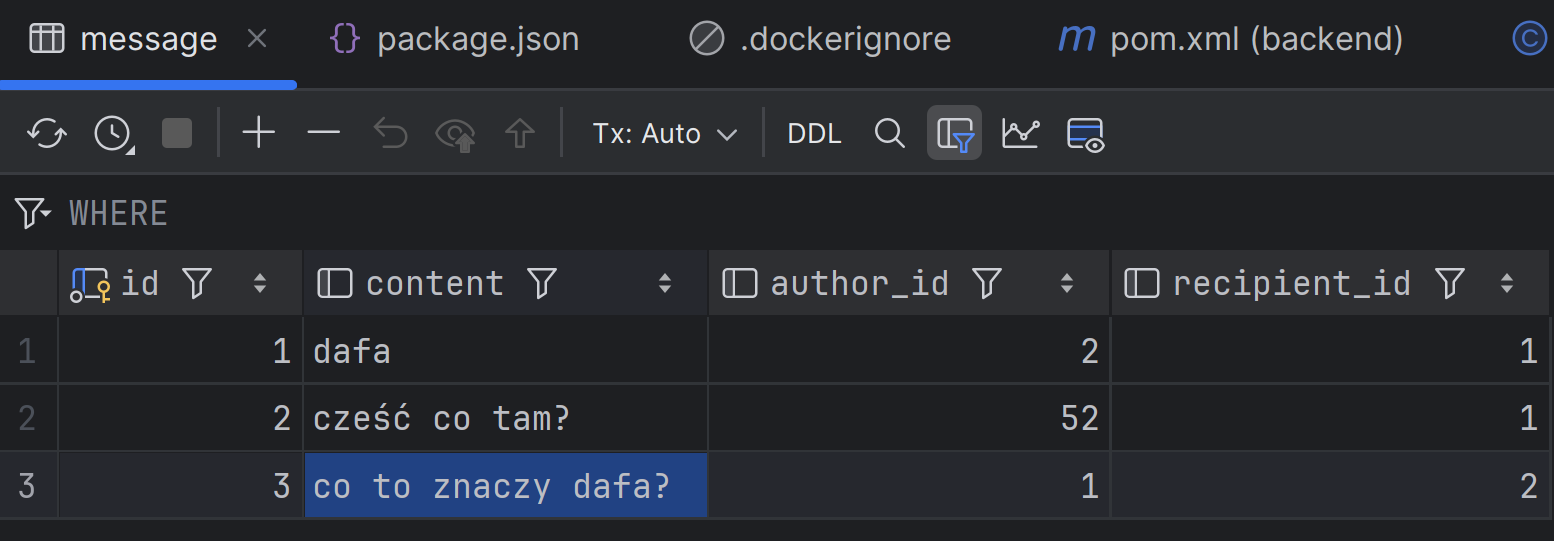
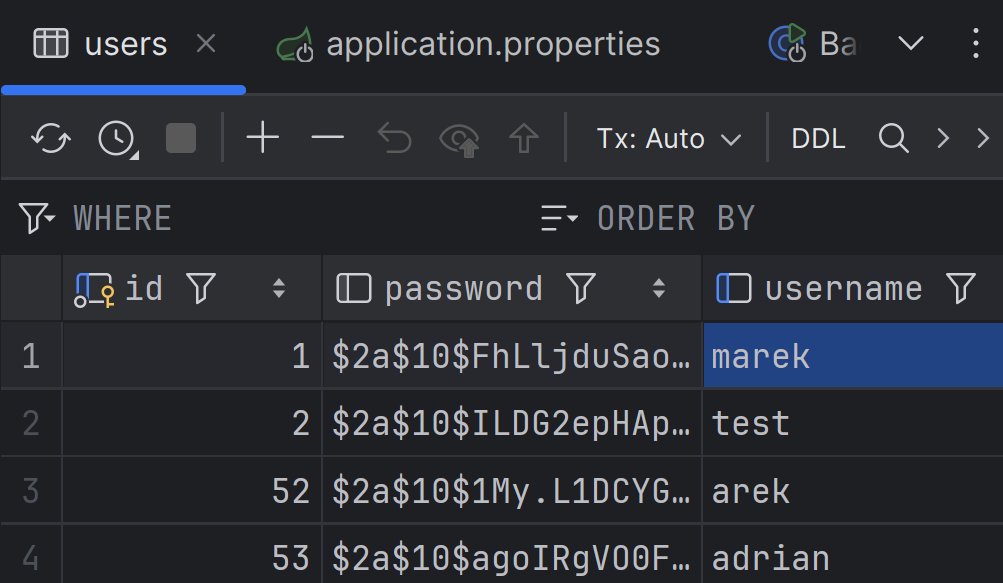
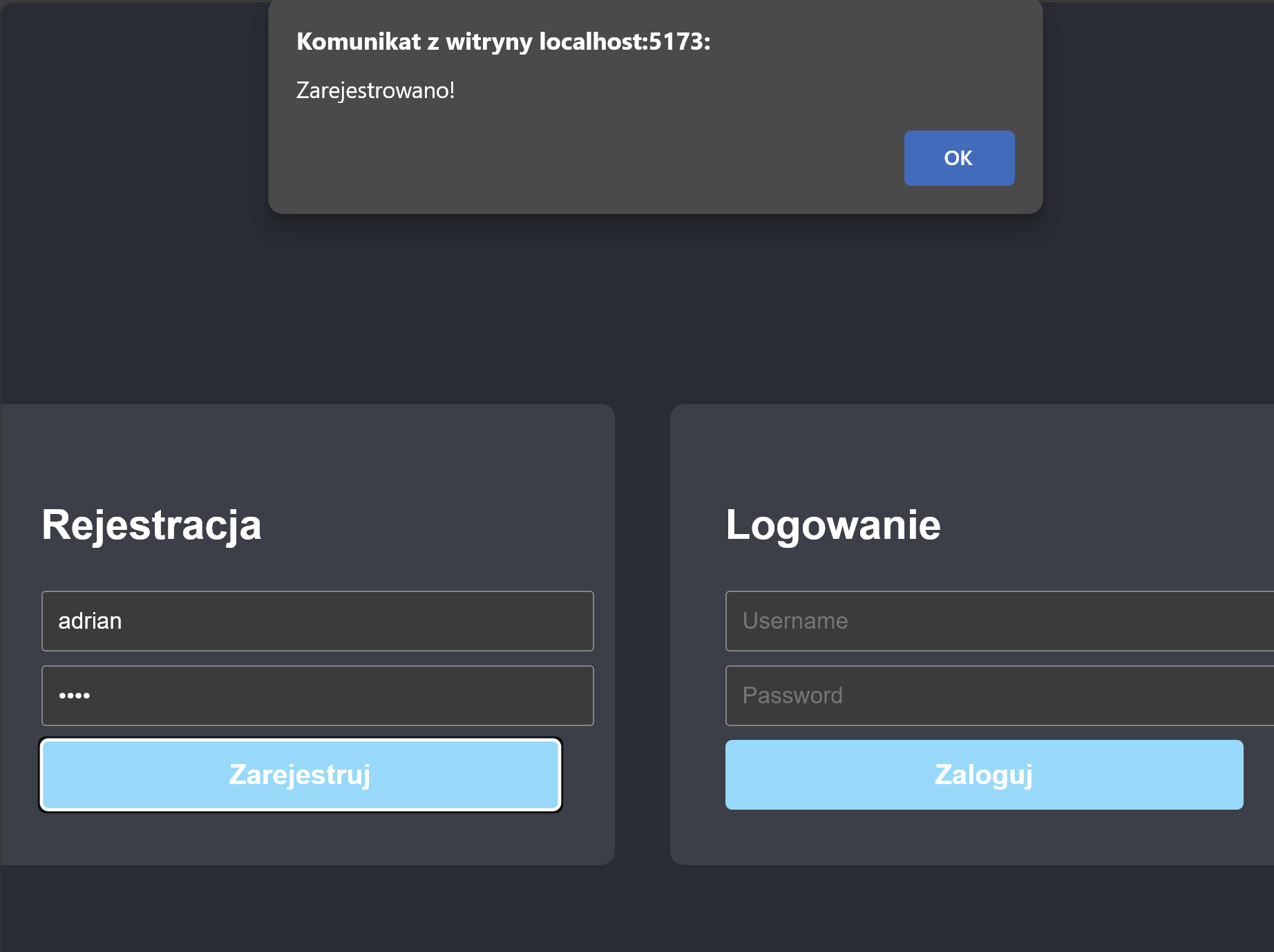
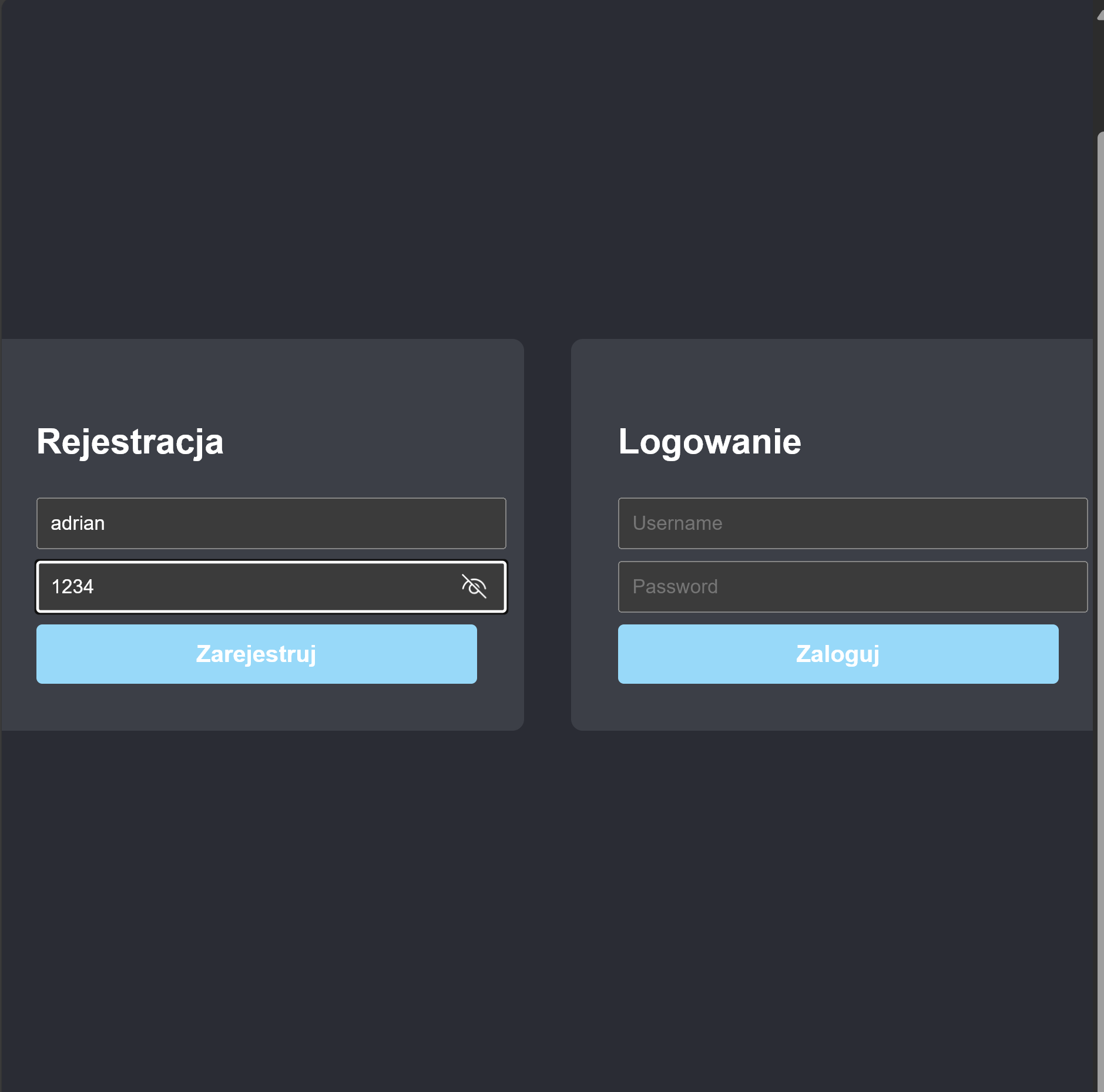
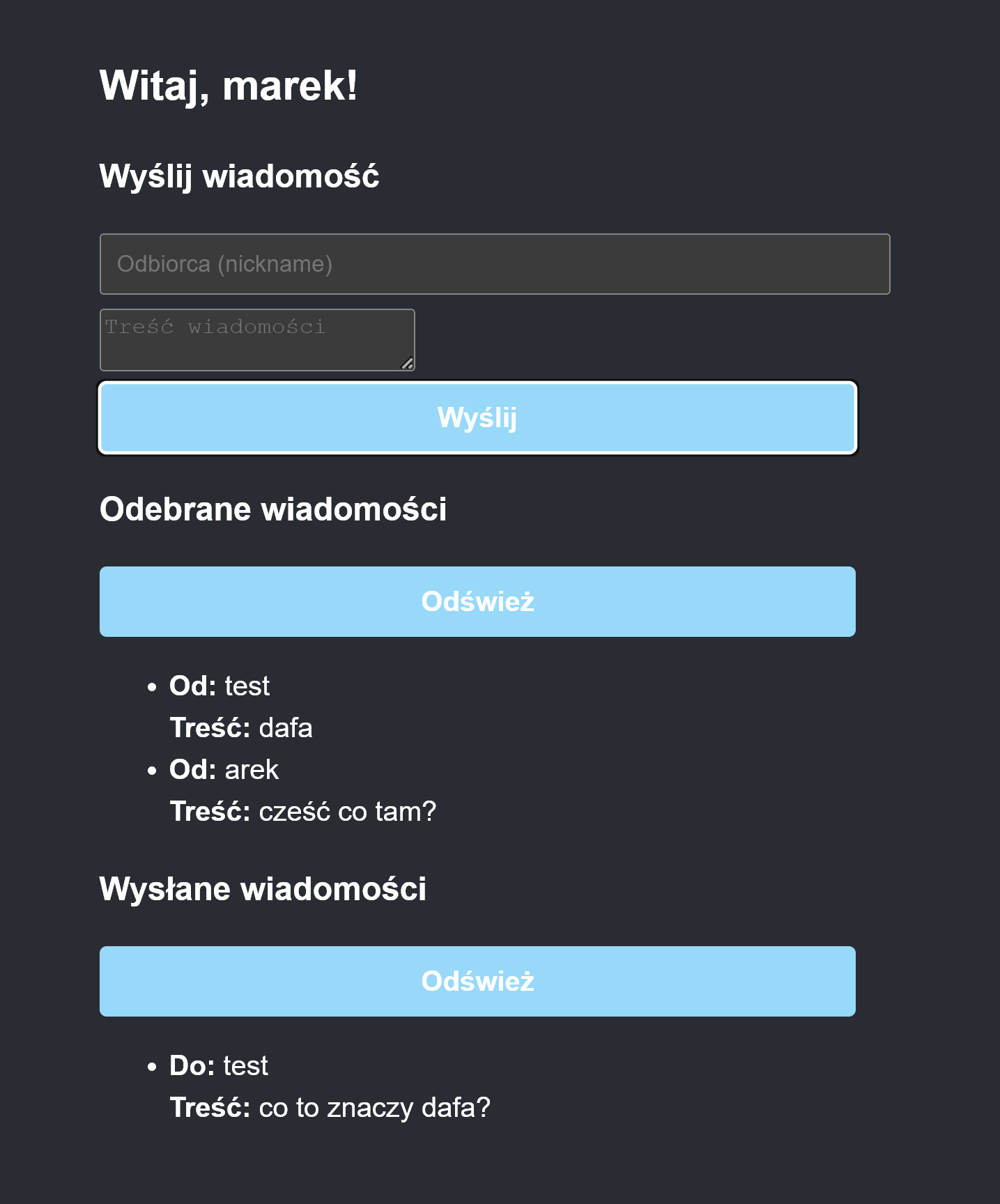
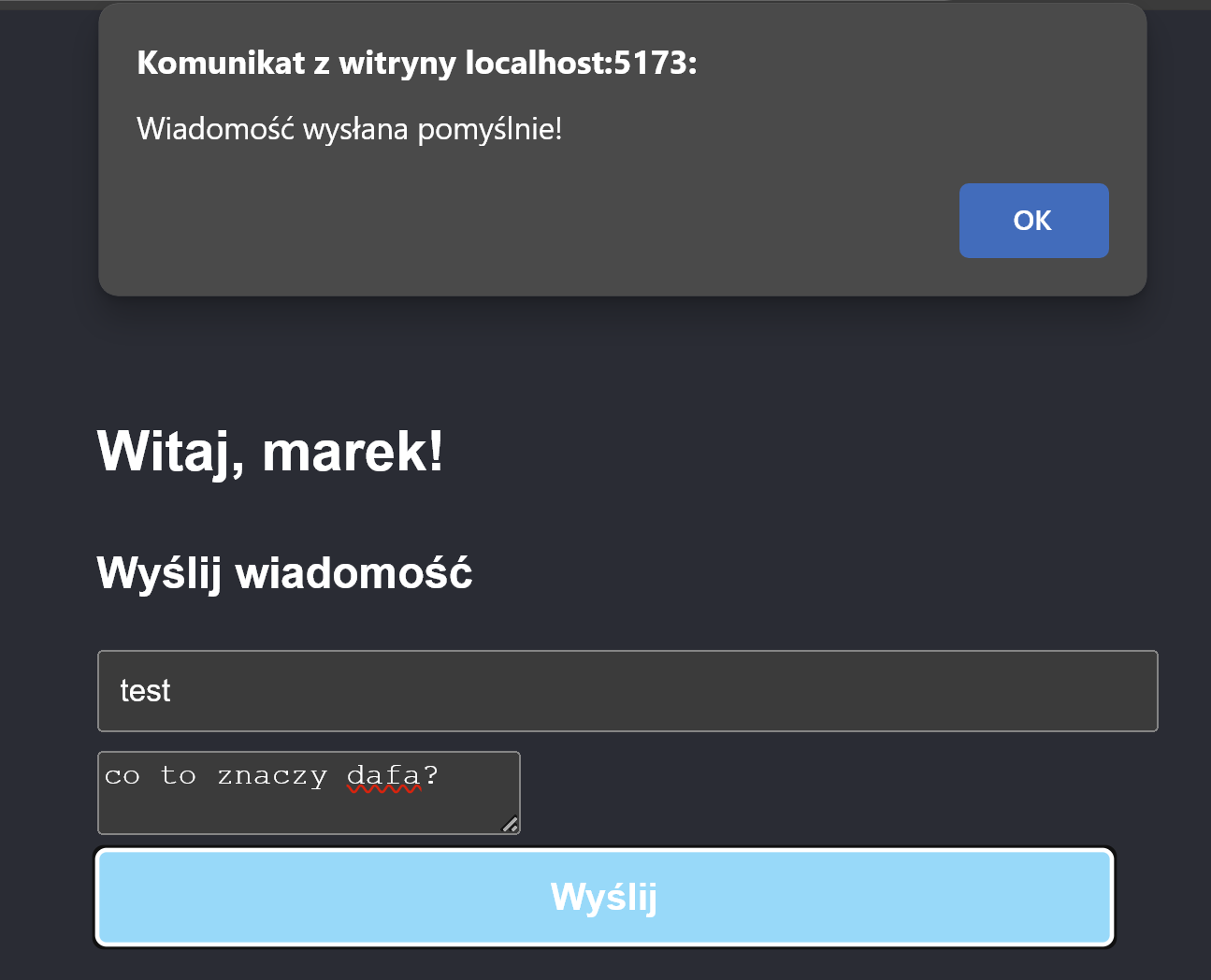
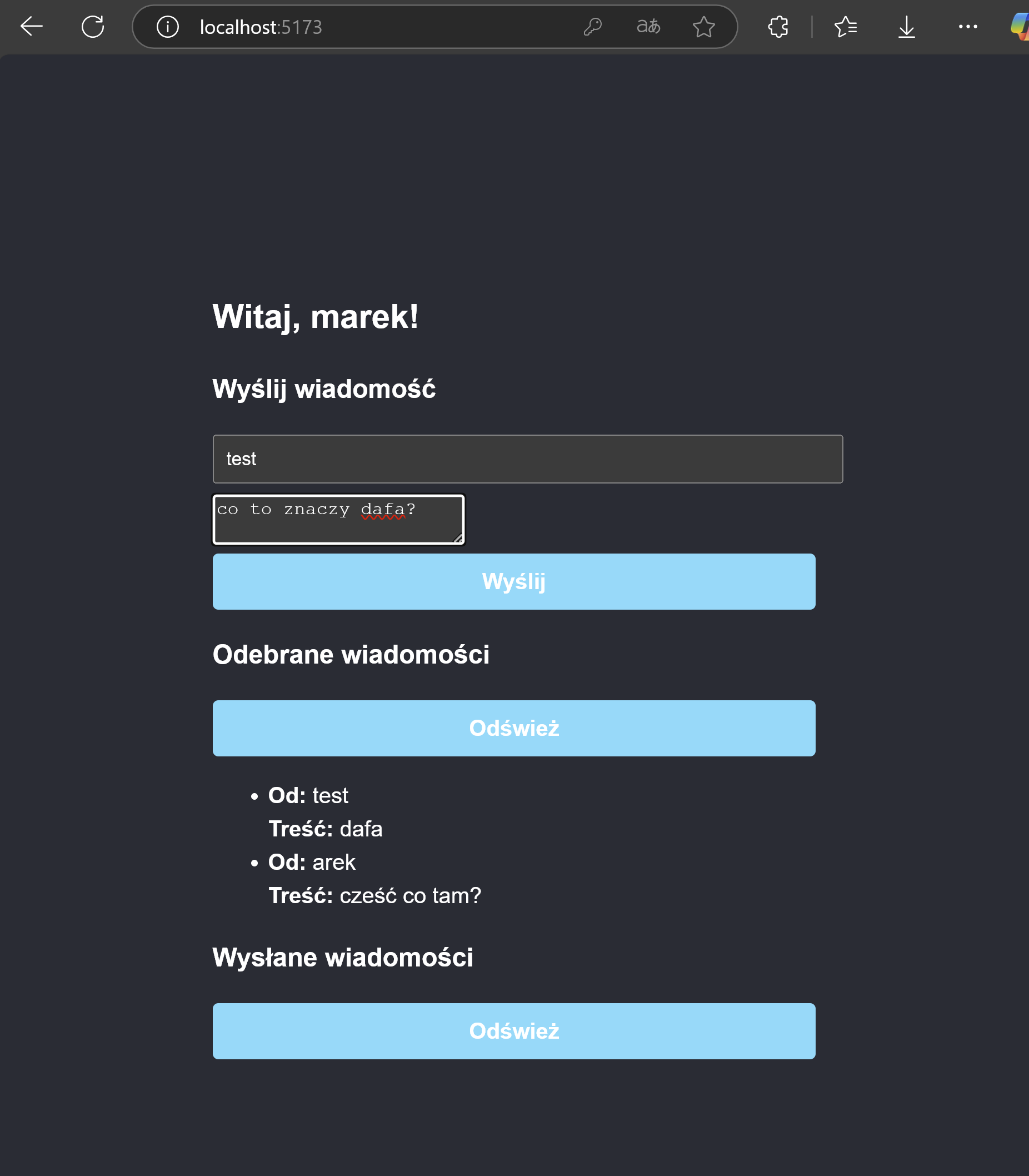
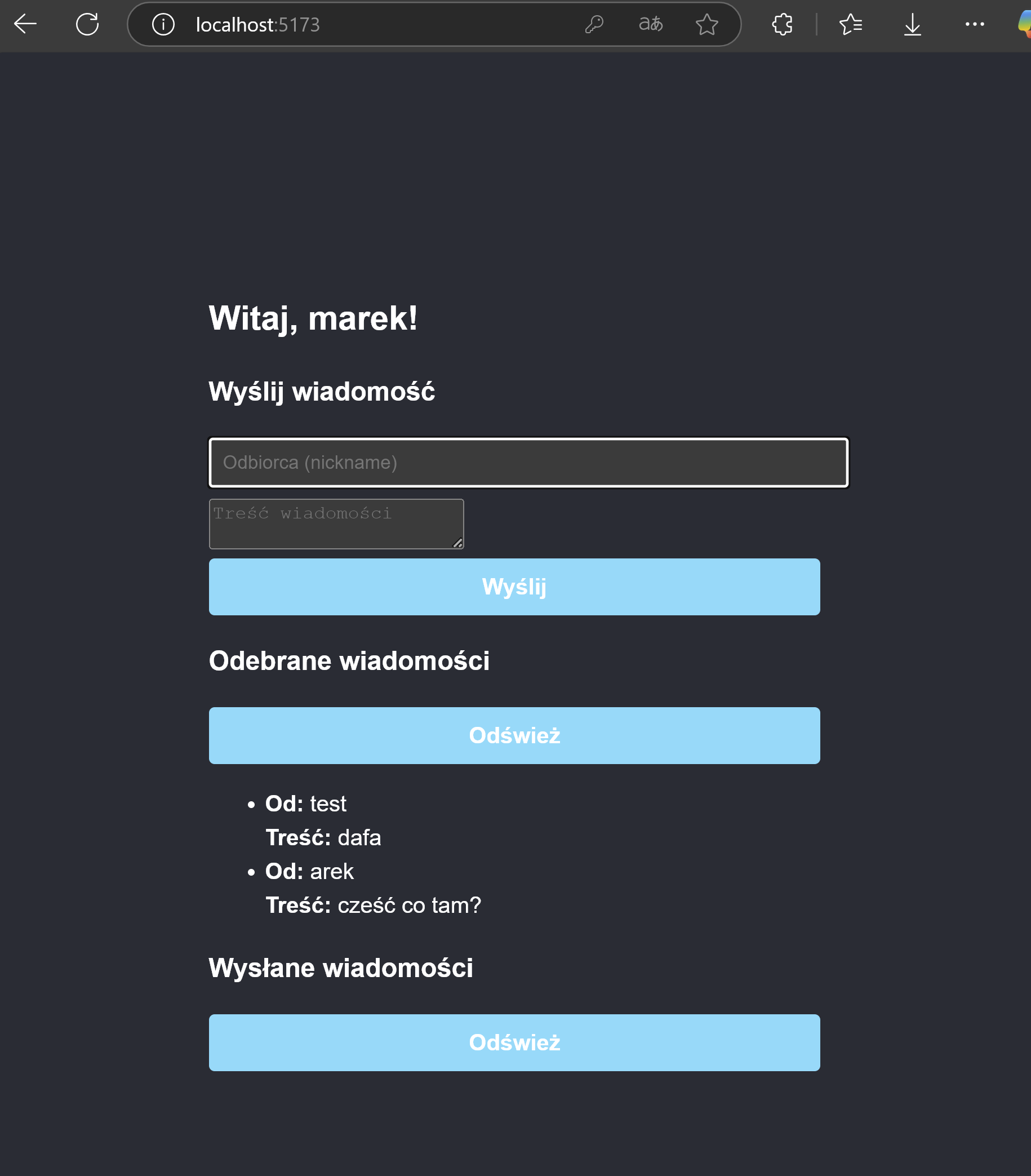
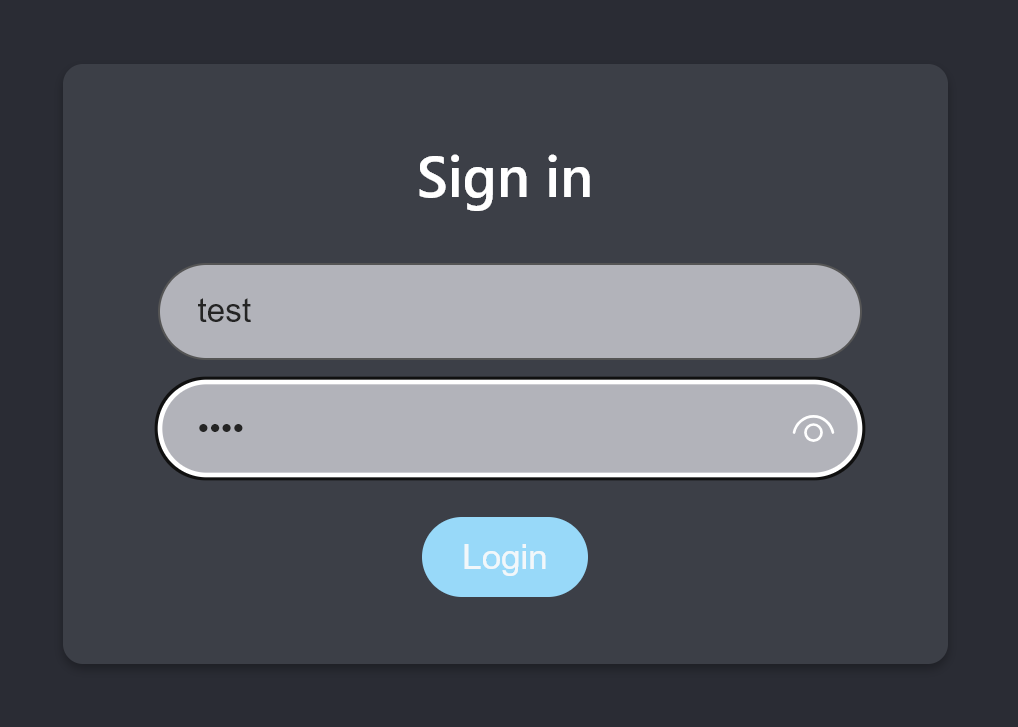
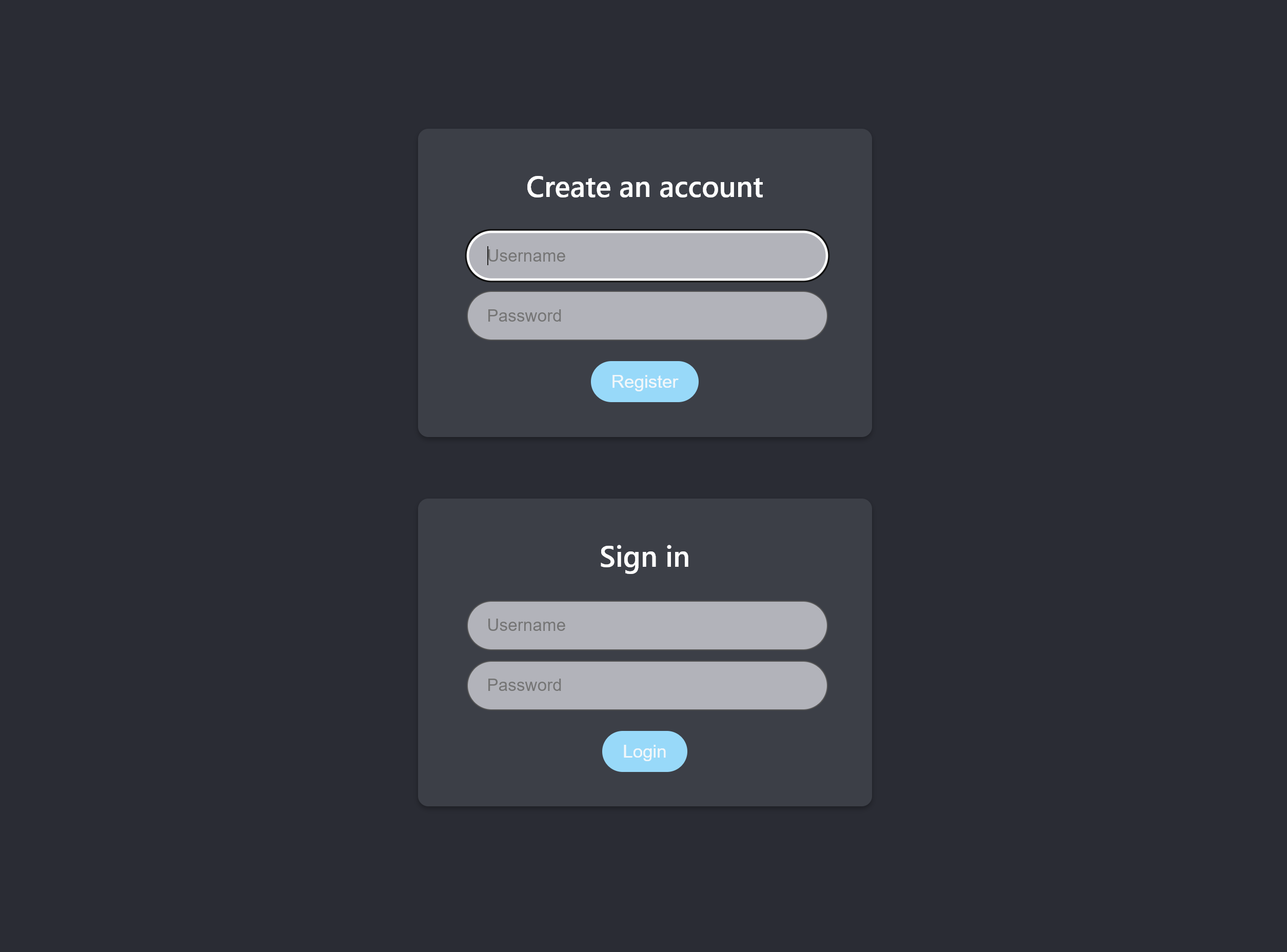


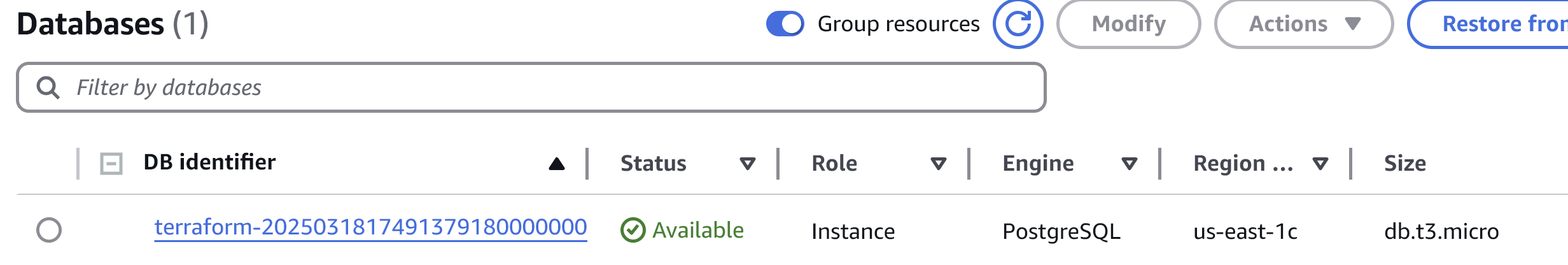
Wszystko zdockerowane przed synchronizacją z AWS ^

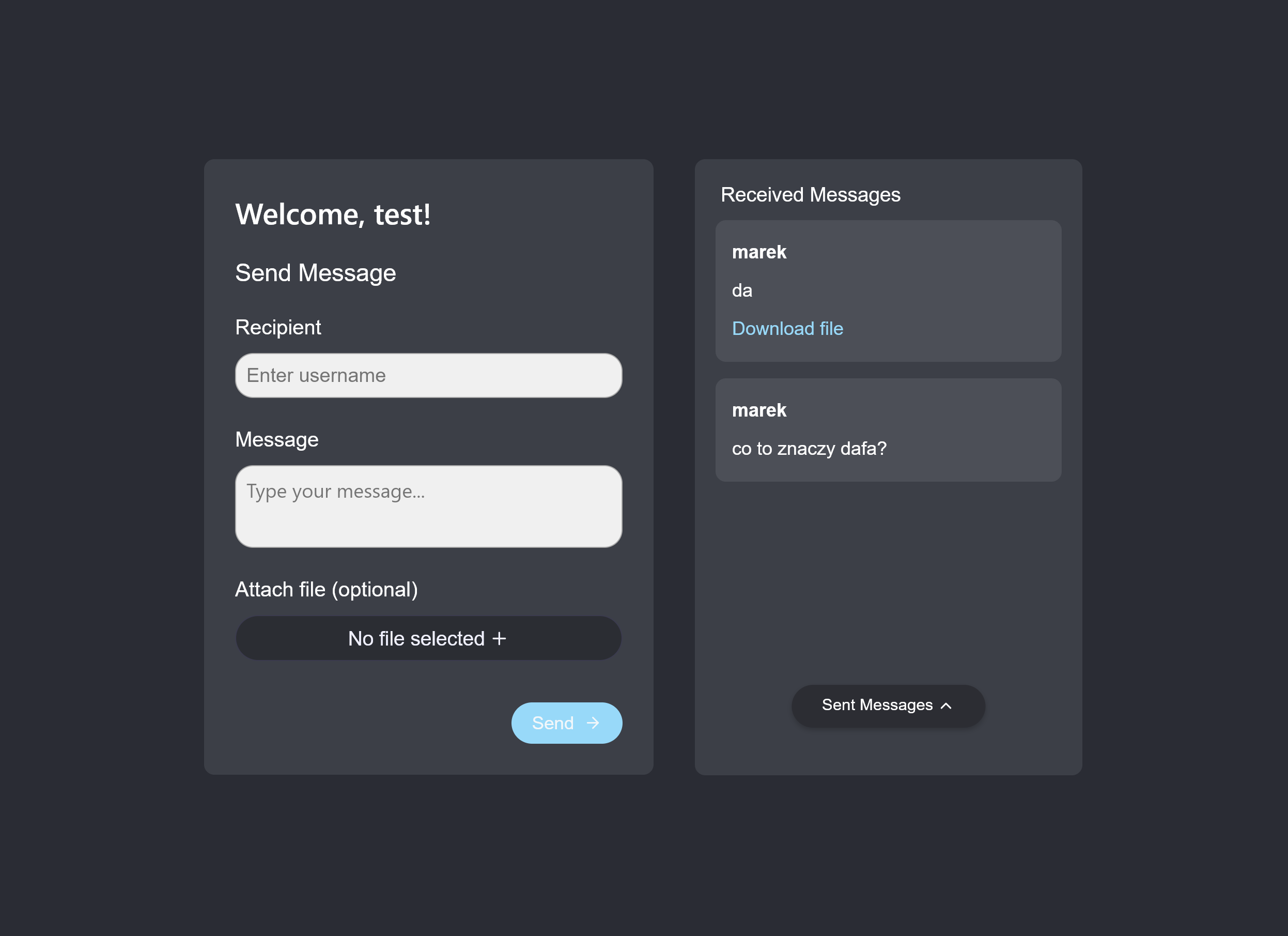
Najpierw skupiłem się aby wszystko działało na localhoście, zanim zsynchronizuję projekt z aws’em

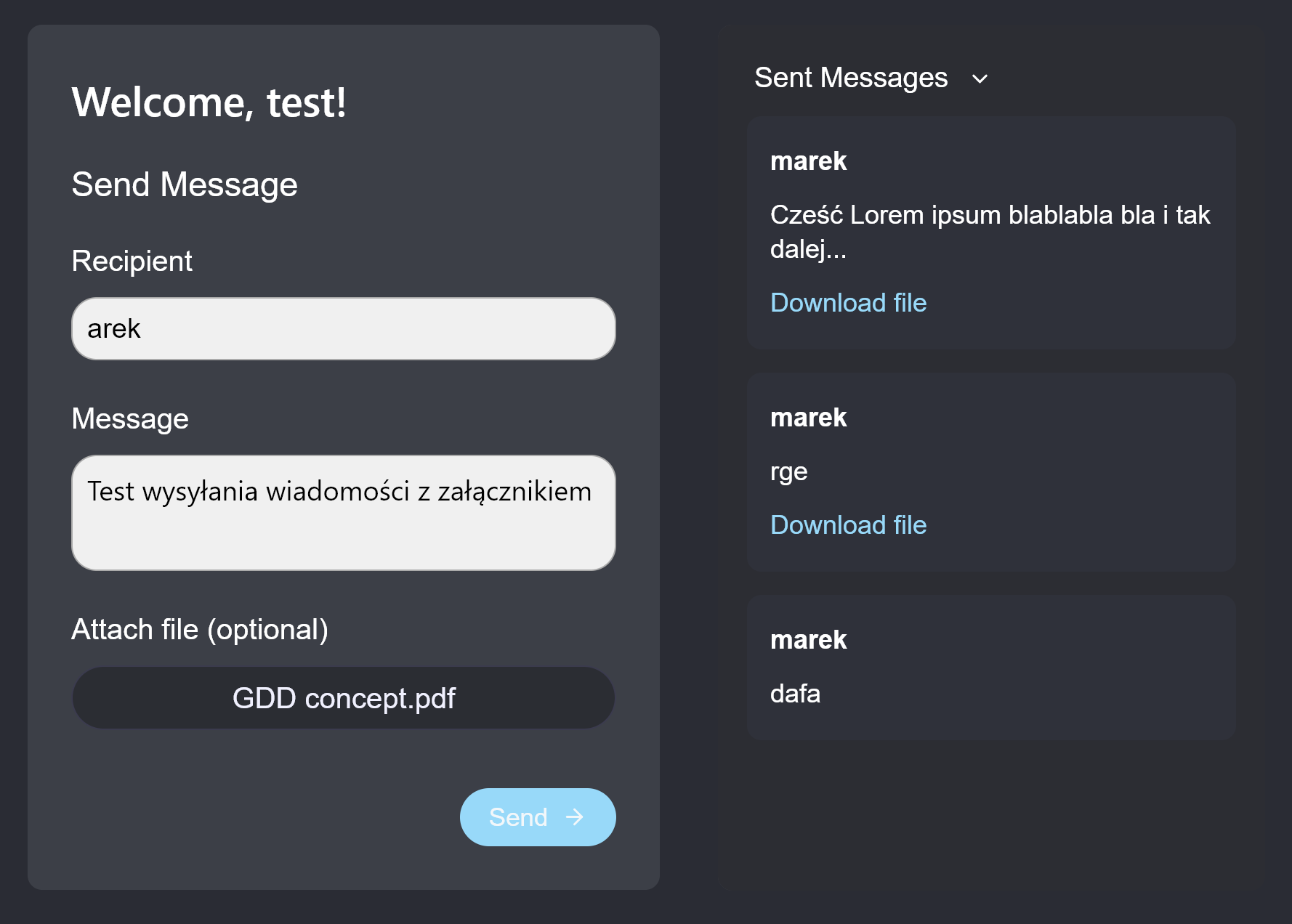


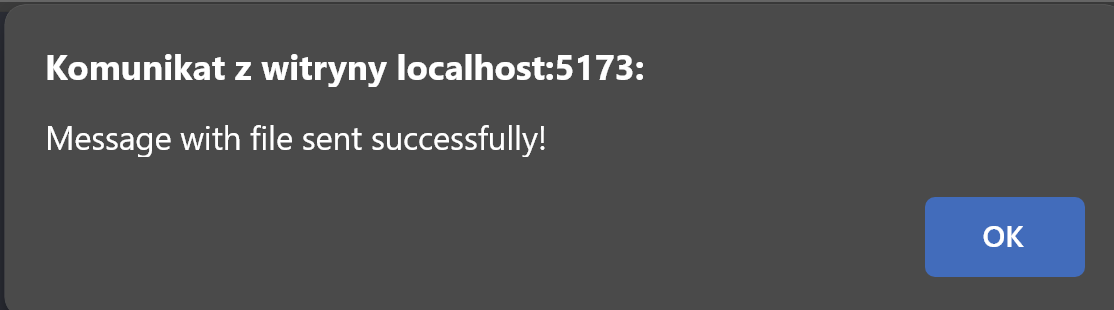


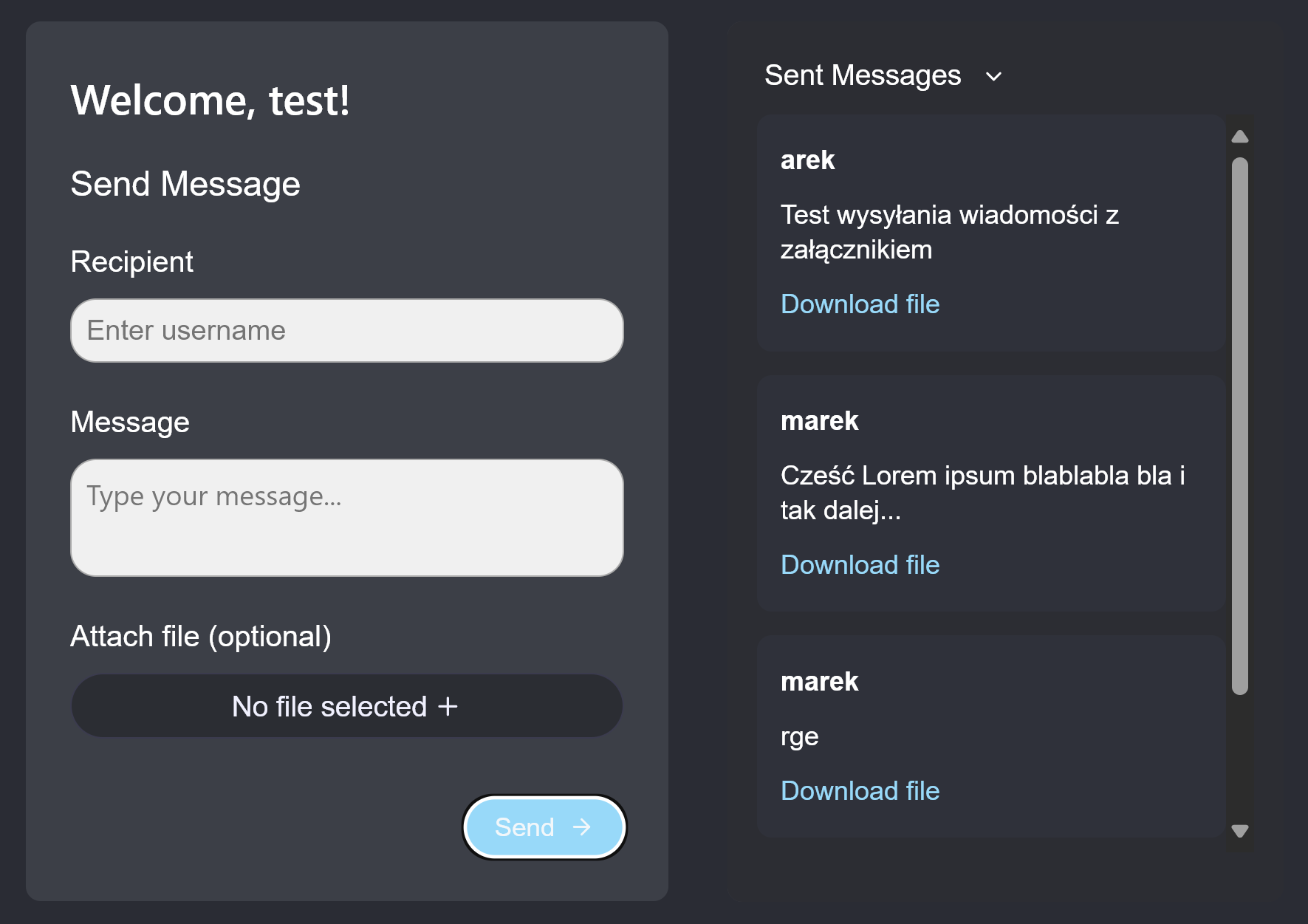
Po synchronizacji z AWS:



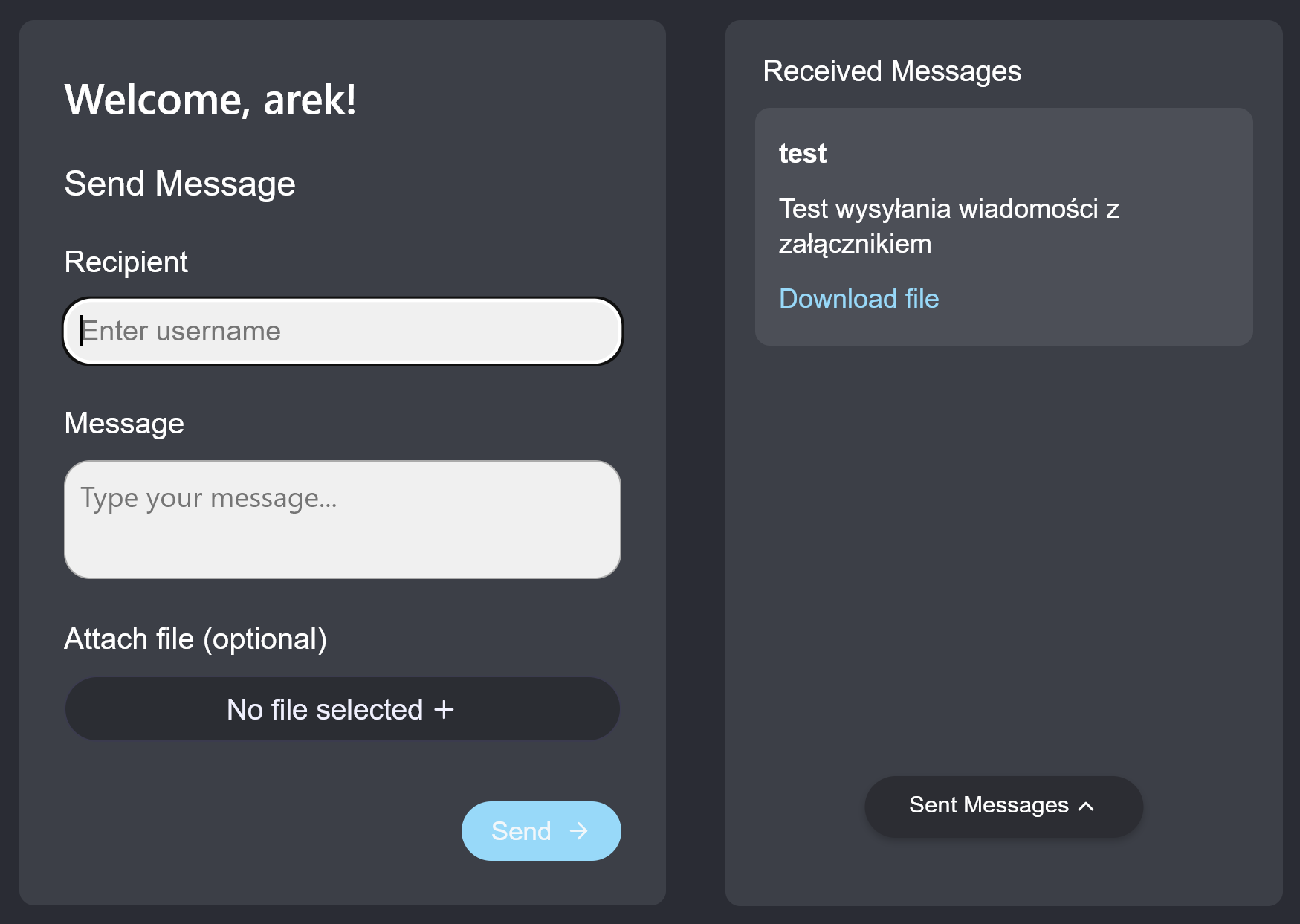




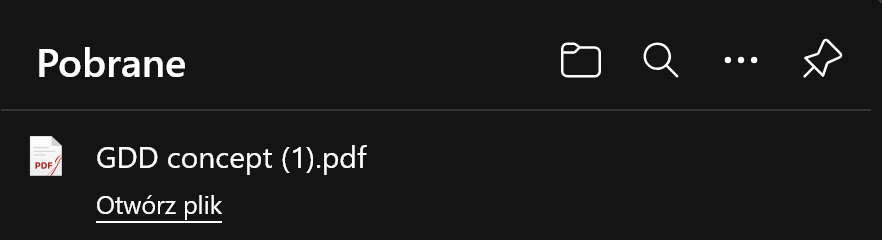


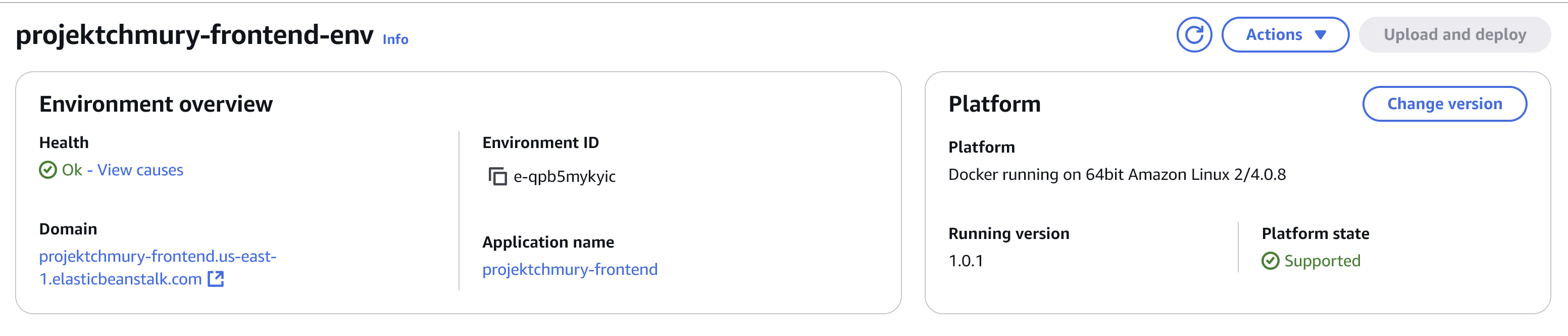
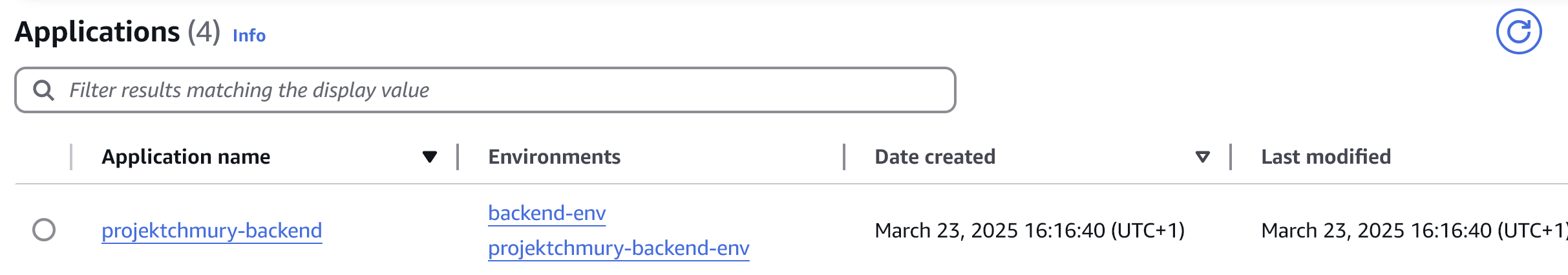


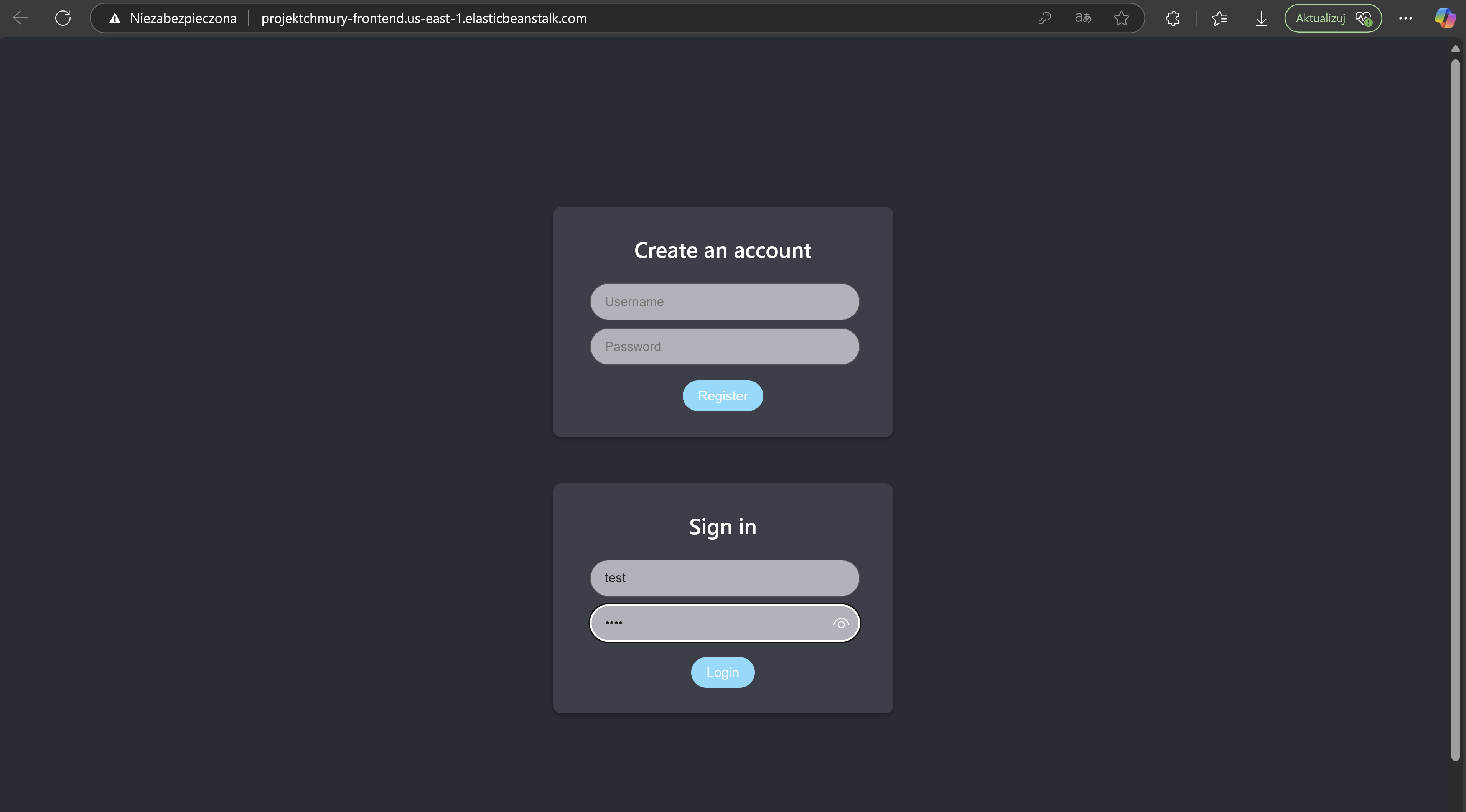
Teraz sprawdzamy u arka



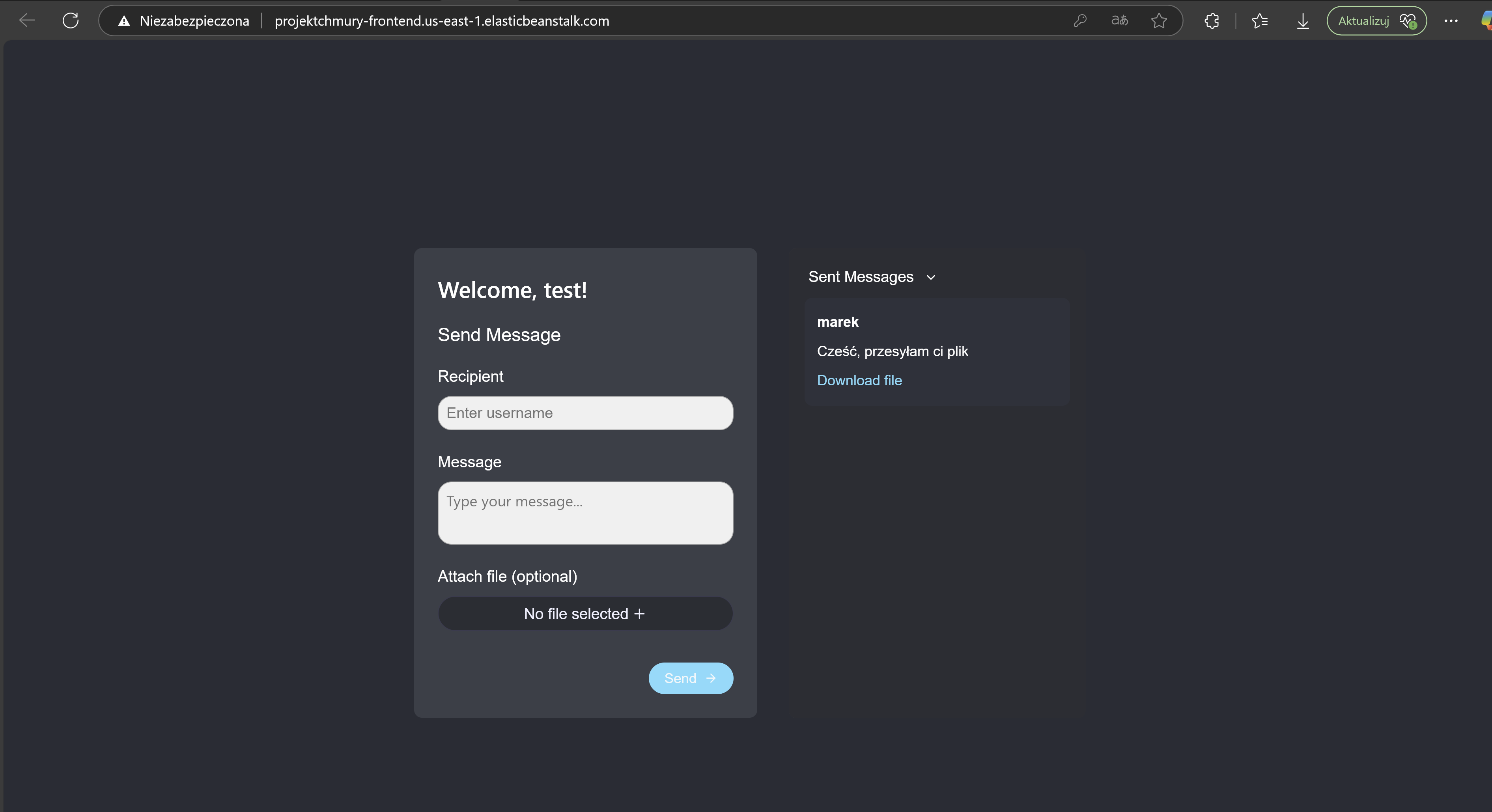
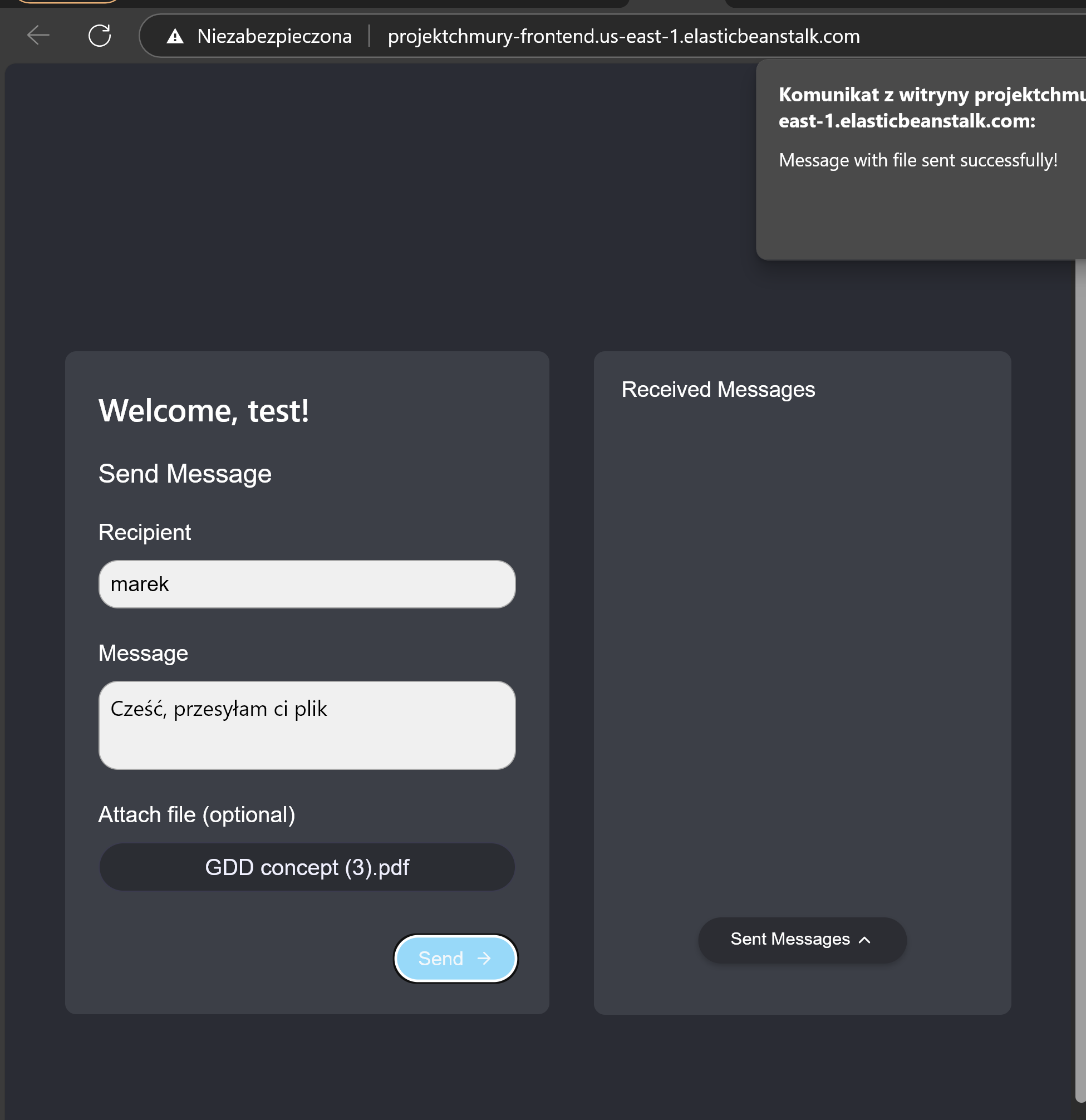
Po kliknięciu na przycisk download file



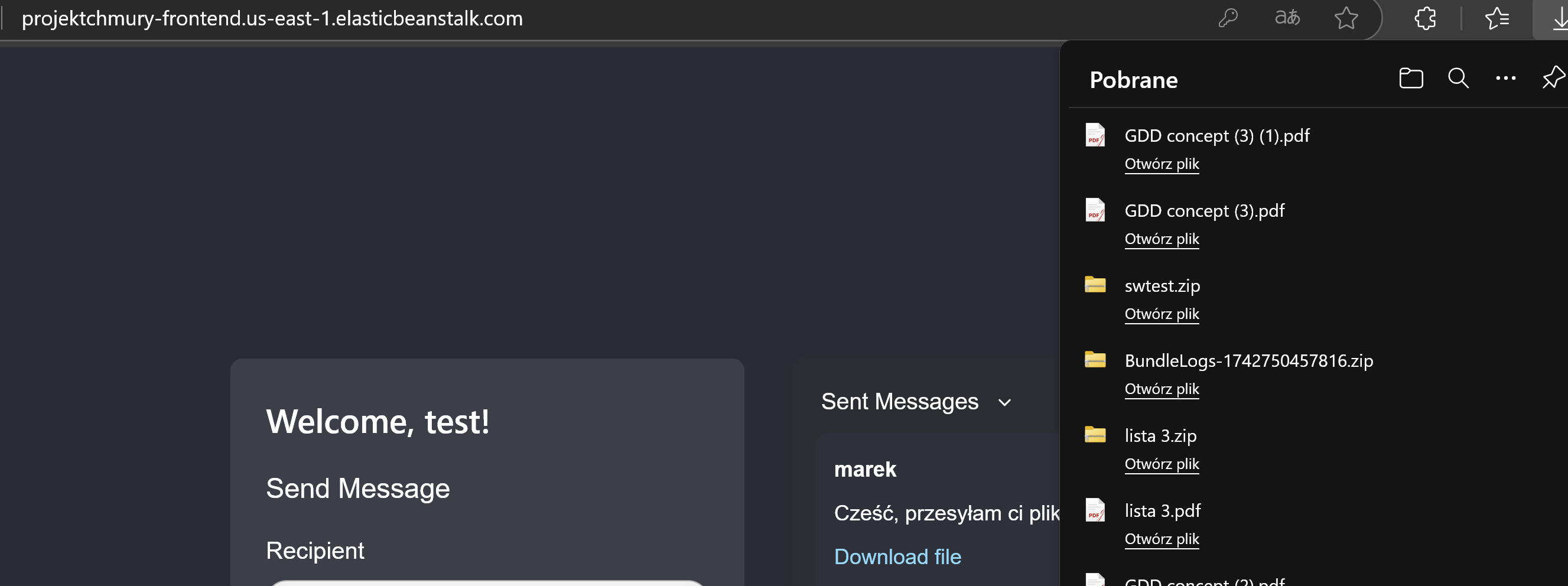
Manualnie skonfigurowany aws:

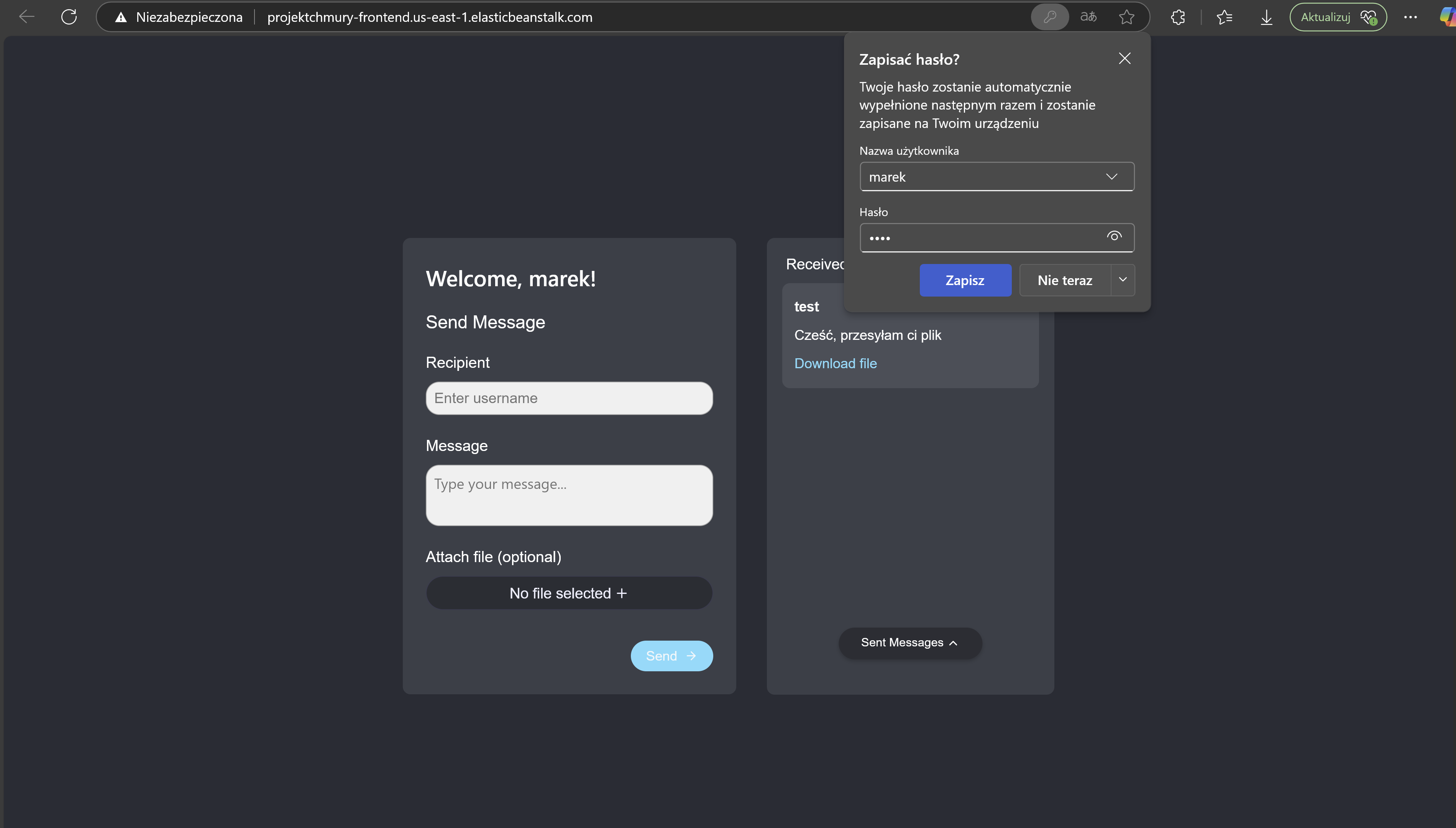


Działający frontend i backend na aws z linka, po zalogowaniu wysyłamy wiadomość do innego użytkownika (test do marek)

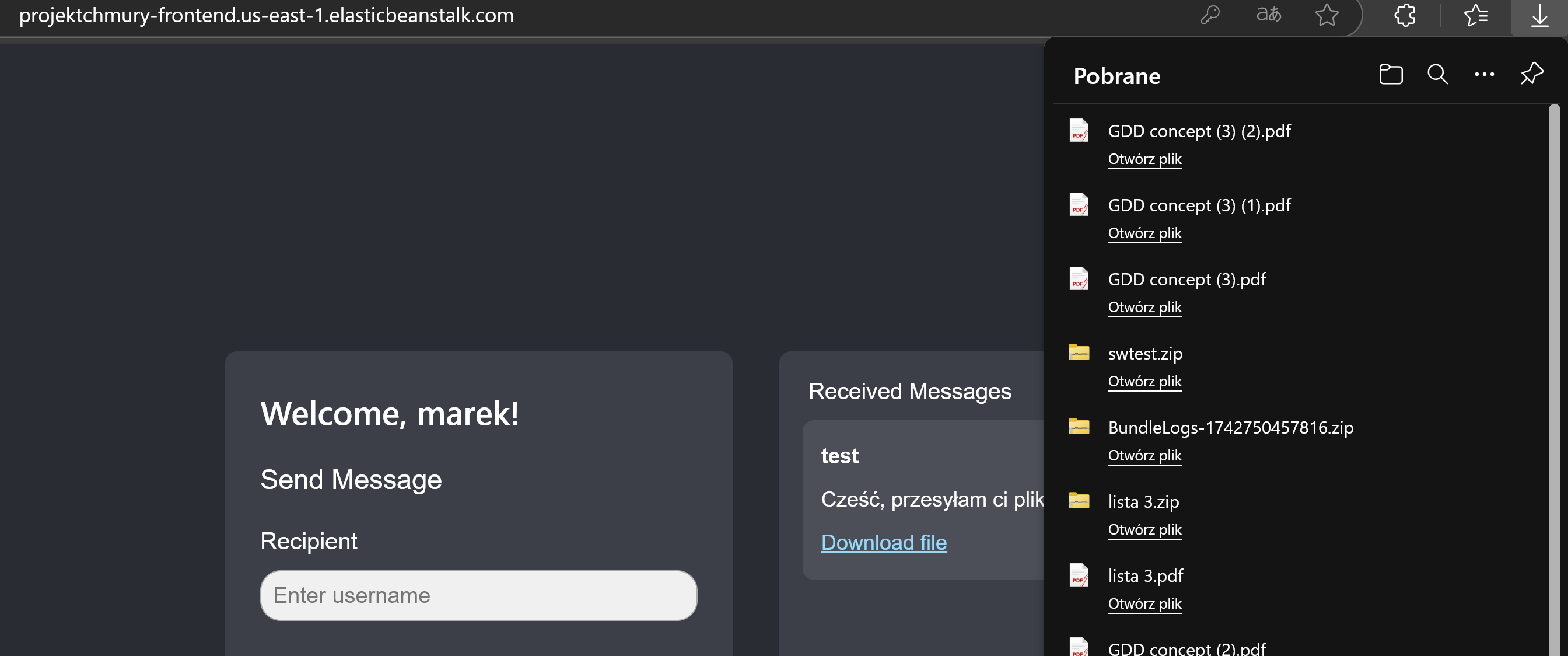


Po kliknięciu Download file plik się pobiera





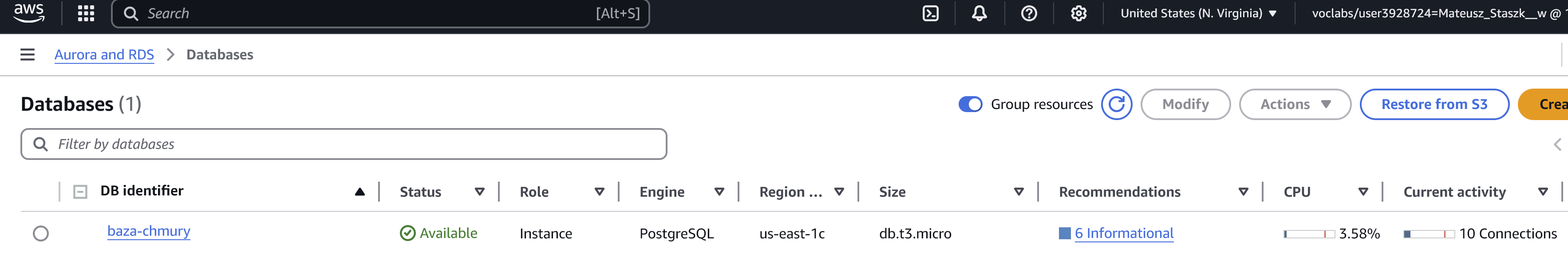
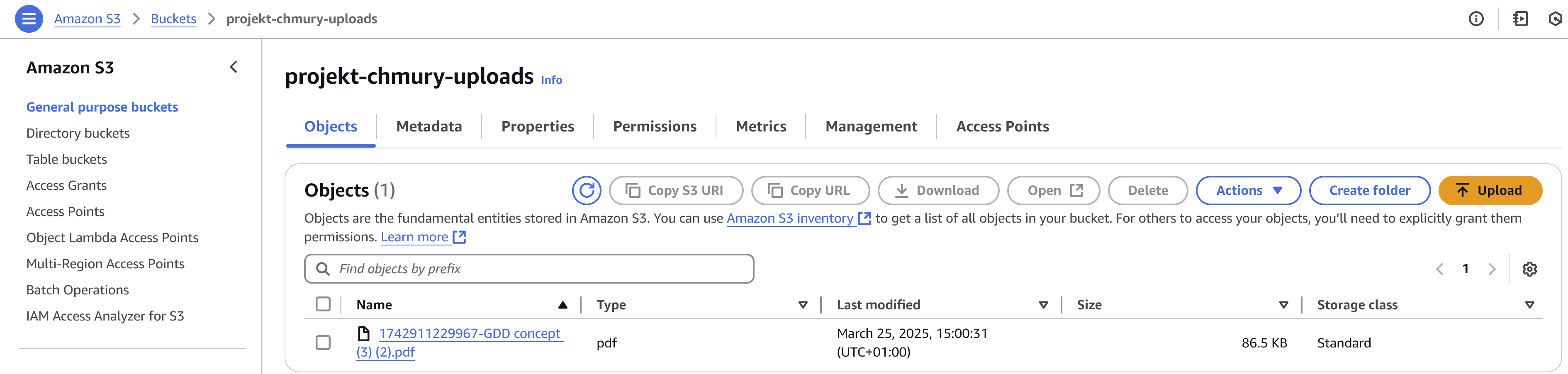
Po zalogowaniu na drugie konto mamy wiadomość od test w otrzymanych wiadomościach



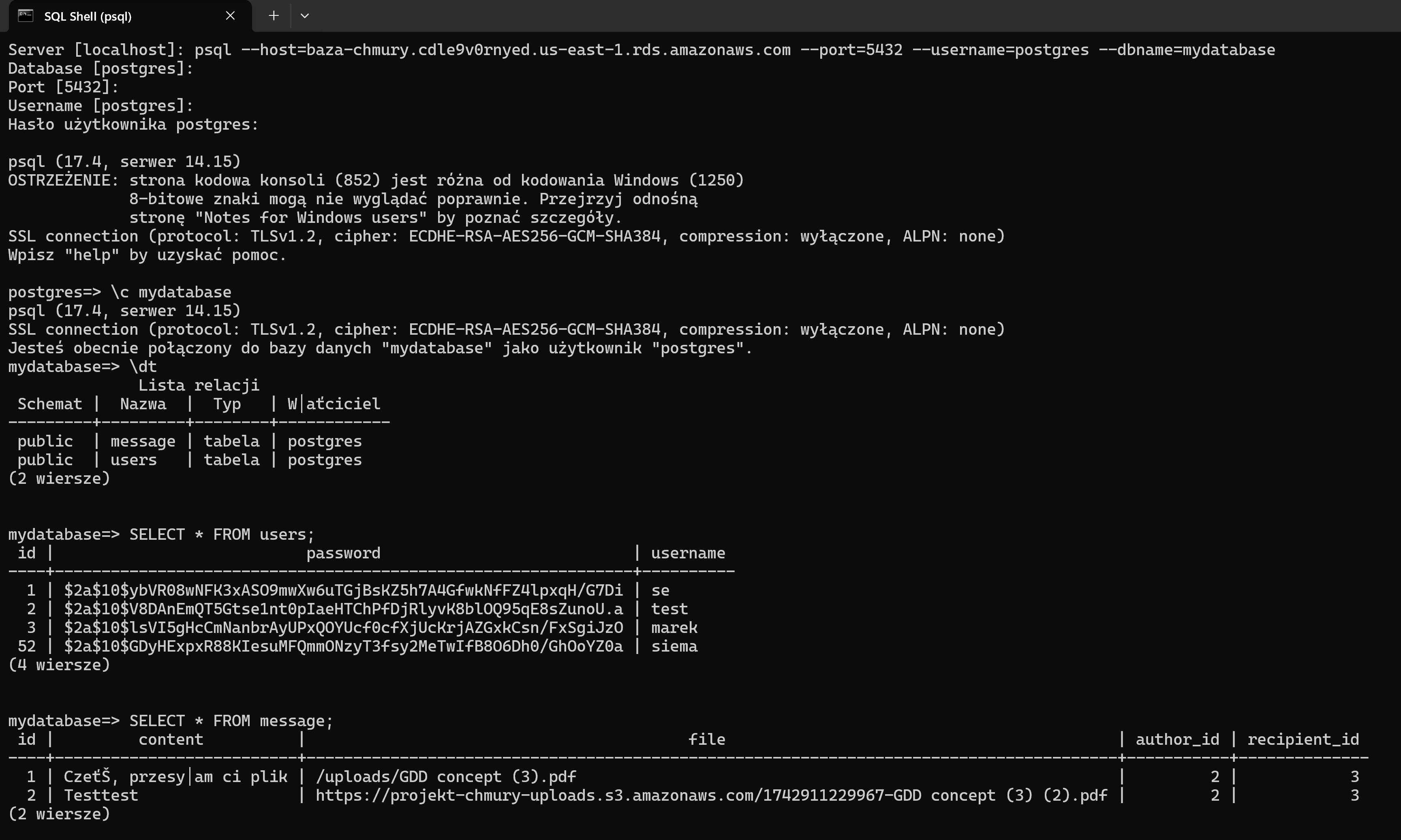
I również plik pobiera się bez problemu

Na ten moment skonfigurowana baza danych w RDS, frontend i backend w Elastick Beanstalk oraz przechowywanie wysyłanych plików w S3

Przesyłane pliki przechowują się w S3:

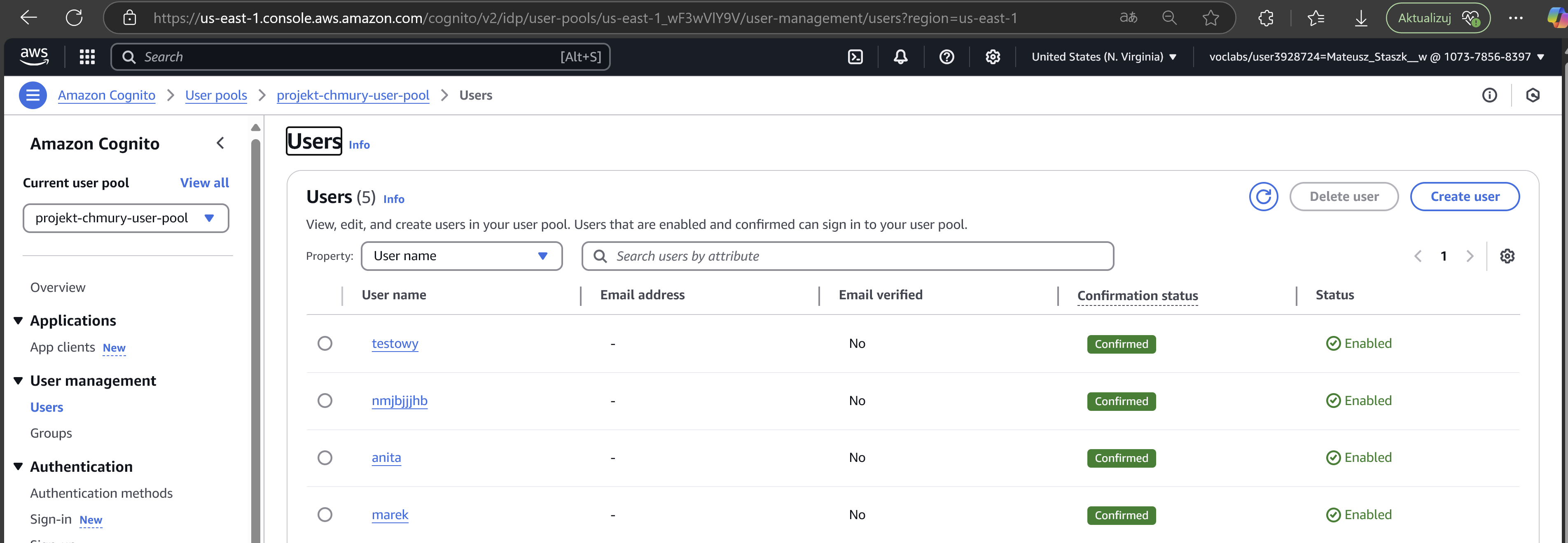


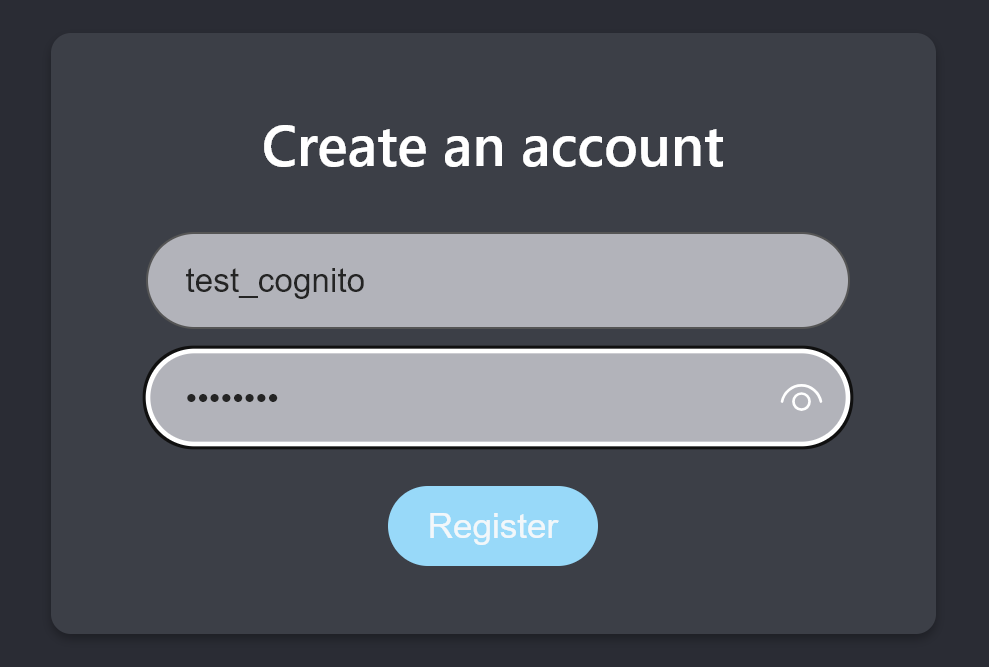
Potwierdzenie poprawnej konfiguracji z RDS i prawidłowego przechowywania danych łącząc się z RDS poprzez psql:

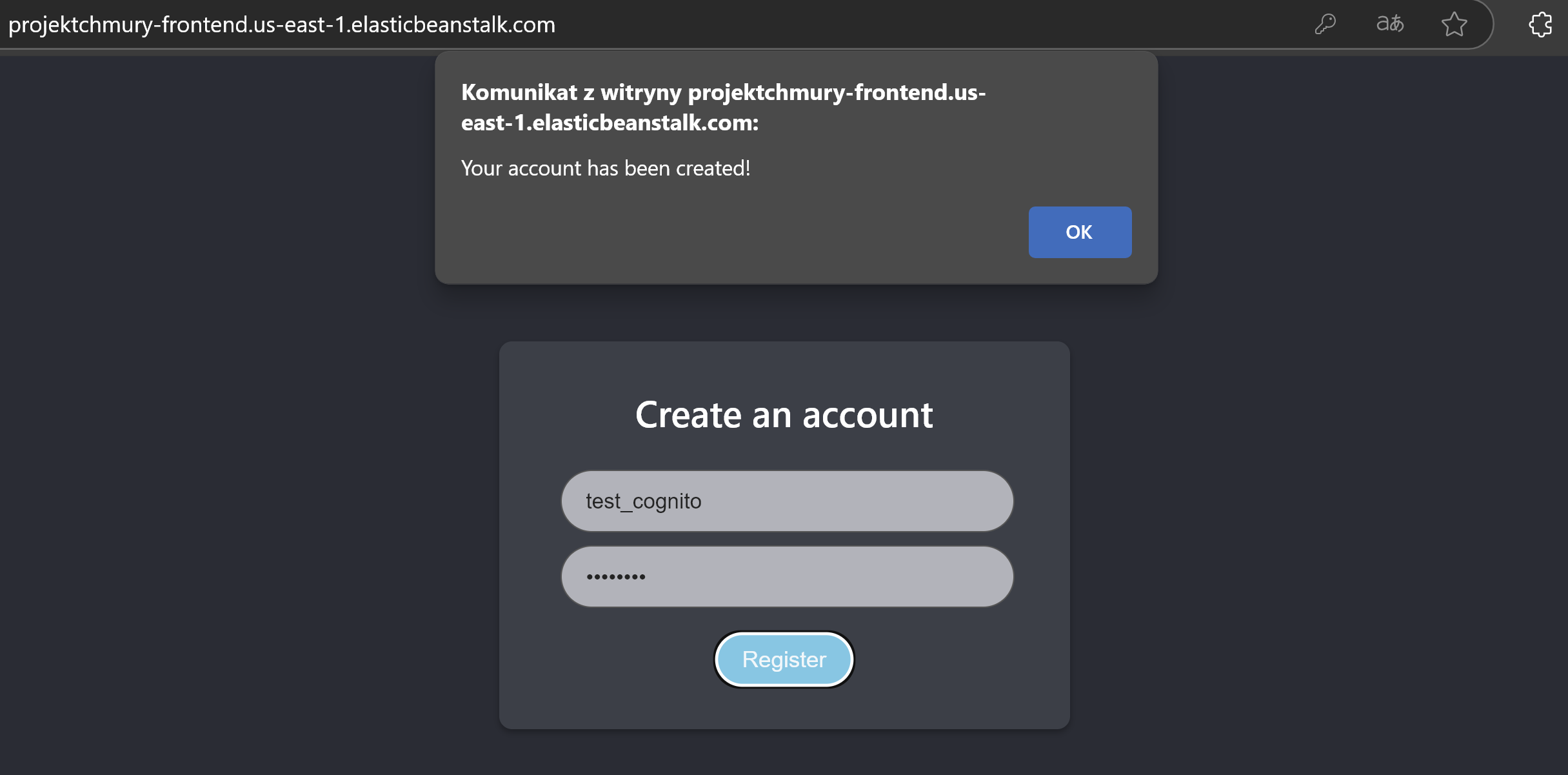


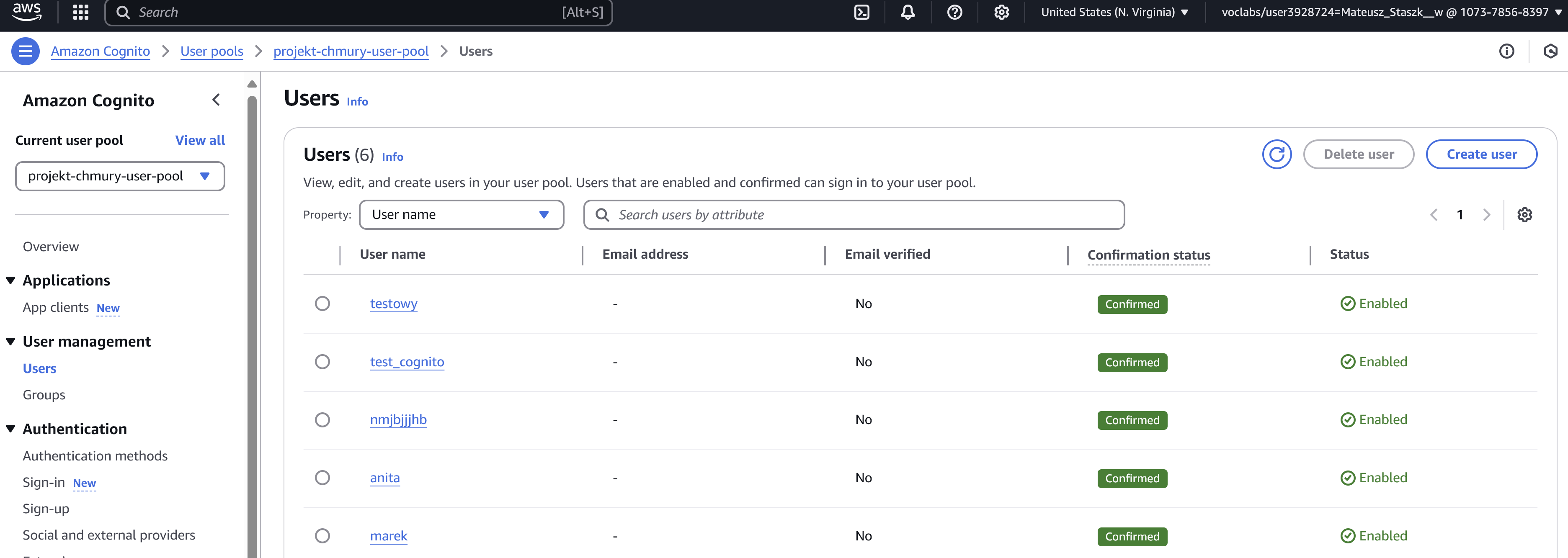
Cognito

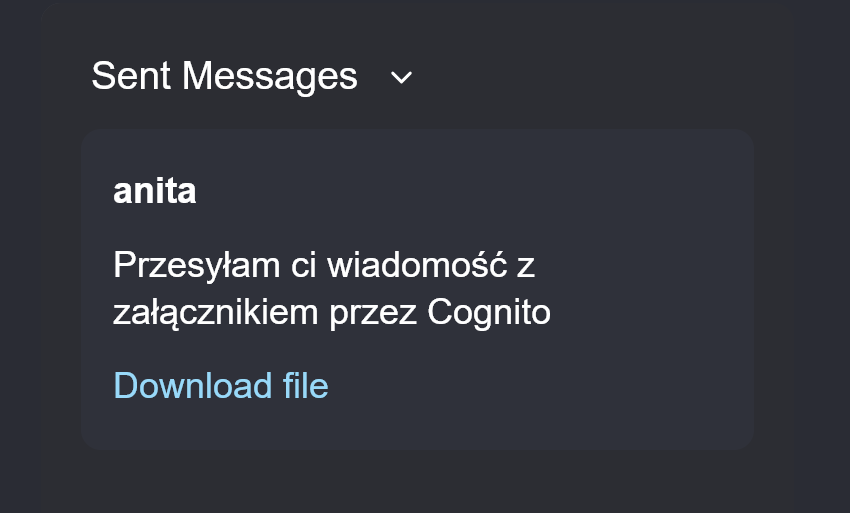
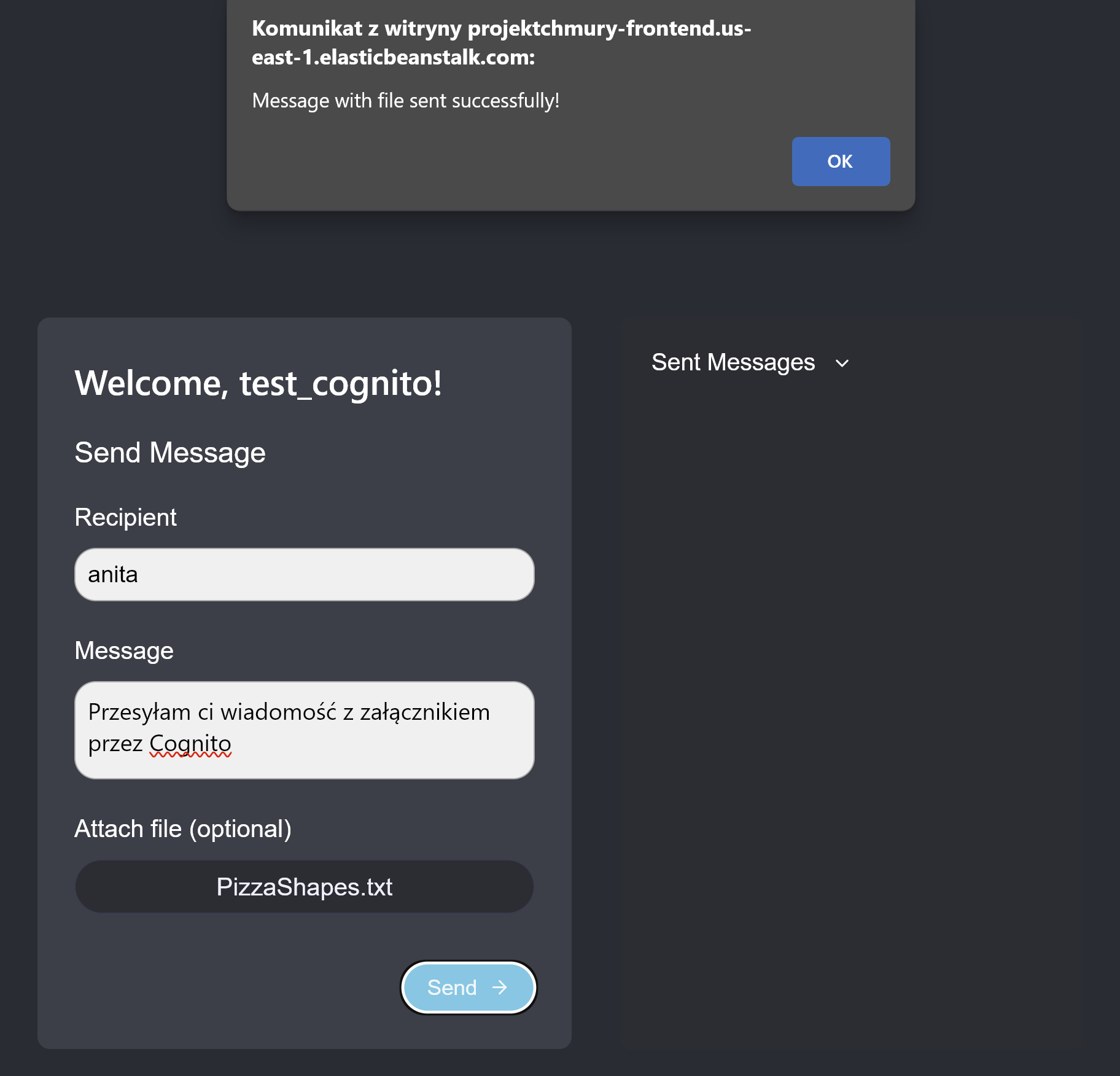
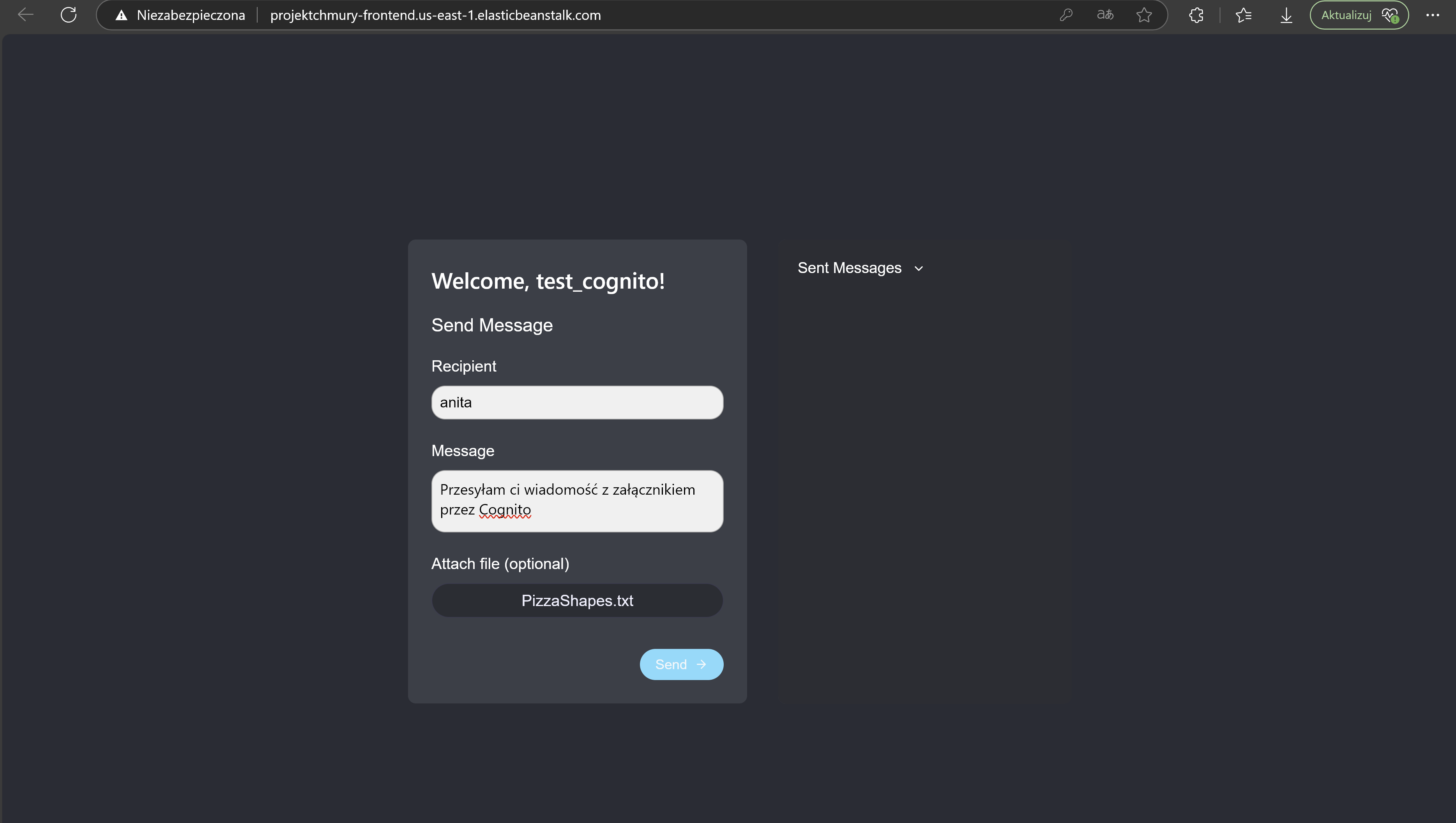
Spróbujmy założyć nowego użytkownika i zobaczymy czy pojawi się jego instancja na stronie AWS Cognito











I jak zalogujemy się na konto anita to również mamy wiadomość od test\_cognito