

APLIKACJA WEBOWA DLA SERWISU ELEKTRONICZNEGO

Kacper Jaskulski, Grzegorz Zych

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki
ul. Mikołaja Kopernika 1, 85-074 Bydgoszcz
e-mail: kacper.jaskulski@student.ukw.edu.pl

Streszczenie: Aplikacja webowa dla serwisu elektronicznego stworzona z uwagi na rosnące zainteresowanie wśród Polaków elektroniką używaną. Moduł logowania na bazie tokenów i przypisanych im ról dostępu pracowników do systemu. Panel pracowniczy umożliwia tworzenie, edycję zleceń naprawy, tworzenie potrzebnych dokumentów np. faktur. Panel administracyjny daje dostęp do zakładki zarządzania danymi pracowników, edycji opisów i obrazów wyświetlanych na stronie, dodawanie dostępnych usług czy dostępnych części. Klient ma możliwość sprawdzenia stanu swojego zlecenia poprzez wpisanie danych w odpowiednim oknie. Aplikacja ma charakter trójwarstwowy tj. interfejs oparty o język js z nakładką React, Rest API wykonane w języku c# odpowiada za poprawne wykonywanie funkcji i komunikację z bazą danych Ms SQL 2022 Express. Testy zostały przeprowadzone z pomocą biblioteki Swagger oraz aplikacji PostMan i zakończyły się pomyślnie.

Słowa kluczowe: Rest API, ASP.NET Core, React, elektronika, naprawa, serwis, relacyjne bazy danych, c#, js

A WEB APPLICATION FOR AN ELECTRONICS SERVICE

Abstract: A web application for an electronics service created due to the growing interest among Poles in used electronics. The login module is based on tokens and their assigned employee roles for system access. The employee panel allows for creating and editing repair orders, as well as creating necessary documents such as invoices. The admin panel provides access to employee data management tabs, editing descriptions and images displayed on the site, adding available services, and available parts. The client can check the status of their order by entering the relevant data in a dedicated window. The application has a three-tier architecture: a JavaScript interface with a React overlay, a Rest API written in C# responsible for proper function execution and communication with the Ms SQL 2022 Express database. Tests were conducted using the Swagger library and the PostMan application and were successfully completed.

Keywords: Rest API, ASP.NET Core, React, electronics, repair, service, relationaldatabases, C#, JS

1. Wstęp

Naprawa i sprzedaż urządzeń elektronicznych, zwłaszcza tych używanych i odnowionych, stają się w Polsce coraz popularniejsze. Trend ten zainspirował stworzenie aplikacji umożliwiającej potencjalnym klientom łatwe i przejrzyste naprawienie lub odnowienie swoich urządzeń. Na zmiany na rynku wpływ mają takie czynniki, jak niestabilna sytuacja geopolityczna, rosnące koszty transportu i opóźnienia w dostawach. Firmy takie jak Intel dywersyfikują produkcję chipów, przenosząc jej część do Europy, co może podnieść koszty elektroniki z powodu wyższych kosztów produkcji.

Jednym z celów stworzenia aplikacji była standaryzacja i systematyzacja działań związanych z serwisowaniem urządzeń elektronicznych. Forma aplikacji webowej jest przyjazna dla użytkownika końcowego, który w prosty sposób może sprawdzić status naprawy swojego urządzenia. Pracownicy otrzymują narzędzie do przejrzystego koordynowania zleceniami, a administratorzy mają

dostęp do zarządzania danymi klientów i pracowników oraz do nadawania nowych użytkowników i przypisywania im poziomów uprawnień. Internet w Polsce jest wszechobecny. W 2023 roku 93,3% gospodarstw domowych miało dostęp do sieci. E-commerce kwitnie: 64,3% osób w wieku 16-74 lat zrobiło zakupy online. Firmy także korzystają z internetu — 98,5% ma do niego dostęp, a 72,8% [4-7] wystawia faktury elektroniczne. Internet umożliwia łatwe porównywanie ofert i dokonywanie świadomych wyborów. Stał się nieodłączną częścią życia codziennego i biznesu.

Zjawisko Re-commerce to model biznesowy polegający na sprzedaży wcześniej używanych rzeczy w dobrym stanie lub odnowionych. Autorem tego terminu jest George F. Colony, który w lutym 2005 roku określił w ten sposób sprzedaż używanych komputerów. Taki rodzaj zakupów staje się coraz popularniejszy w Polsce, co jest wynikiem pogarszającej się sytuacji finansowej nabywców oraz zmieniającego się podejścia do zakupu używanych rzeczy w celu zmniejszenia śladu

węglowego. Największymi platformami agregującymi sprzedawców i usługodawców w Polsce są Allegro i OLX. OLX w swoim raporcie „Re-commerce bez tajemnic” podał, że 79% produktów w dziale elektronika oznaczonych jest jako „używane”. Z raportu IDC dotyczącego sprzedaży detalicznej na rynku komputerowym wynika, że wzrost sprzedaży nowej elektroniki w pierwszym kwartale 2023 roku wyniósł -29,0%, a w trzecim kwartale -7,6% [1-3]. Dane te wskazują na wyraźny spadek zainteresowania nową elektroniką na światowym rynku.

2. Opis narzędzi i technologii zastosowanych w pracy

Podstawą aplikacji była platforma ASP.NET Core 7.0 oraz darmowe biblioteki, takie jak Microsoft.EntityFrameworkCore, ADO.NET i Microsoft.AspNetCore.Mvc.

Microsoft.EntityFrameworkCore zarządza bazą danych, ADO.NET obsługuje transakcje, a Microsoft.AspNetCore.Mvc ułatwia skalowalność aplikacji oraz pisanie i testowanie kodu. Wprowadza również „Routing” do mapowania funkcji.

Microsoft.Cryptography.HMACSHA256 odpowiada za zabezpieczenia haseł. Proces przyznawania i walidacji tokenów realizują biblioteki takie jak System.IdentityModel.Tokens.Jwt i JWT. Microsoft.AspNetCore.Authorization definiuje role użytkowników i dostęp do zasobów po autentykacji [9-12].

React.js został użyty do stworzenia aplikacji webowej i interfejsu użytkownika, z JSX umożliwiającym pisanie kodu JavaScript bezpośrednio w HTML. Do żądań HTTP użyto biblioteki Axios [14], przesyłającej dane w formacie JSON.

MS SQL Server Management Studio 19 umożliwił konfigurację i projektowanie logiki bazy danych, wspierając mechanizm Write-Ahead (WAL), który zapisuje zmiany w dzienniku transakcyjnym przed ich wprowadzeniem do bazy.

Postman weryfikuje poprawność zwracanych danych i testuje zabezpieczenia. Umożliwia przeprowadzenie testów jednostkowych i integracyjnych oraz oceny wydajności metod.

Swagger dokumentuje, projektuje i testuje aplikacje REST, oferując prosty interfejs graficzny. Wyszczególnia funkcje i kod zwracany przez serwer, co ułatwia testy jednostkowe.

REST API komunikuje interfejs aplikacji z serwerem, przesyłając niezbędne informacje w formacie JSON. Aplikacja jest „beztanowa” i nie przechowuje informacji o sesji użytkownika. Uwierzytelnianie odbywa się za pomocą tokenów, a dane są przesyłane w różnych formatach [15].

REST API może być łatwo zintegrowana z innymi rozwiązaniami, oszczędzając czas i zasoby firmom e-commerce [8]. Dzięki temu można zintegrować aplikację webową z kontem firmy dostarczającej przesyłki, wyświetlając status i listę przesyłek.

3. Propozycja nowego systemu wspomagającego serwis elektroniki.

Przedstawiony system to trójwarstwowa aplikacja webowa służąca do zarządzania serwisem urządzeń elektronicznych. Składa się z trzech kluczowych warstw: interfejsu użytkownika, REST API oraz serwera z danymi. Każda z nich pełni istotną rolę w zapewnieniu funkcjonalności, bezpieczeństwa i wydajności systemu [13].

Elementy systemu:

1. **Usługa:** Odnosi się do możliwych napraw urządzeń z usterkami, takich jak np. brak dźwięku w głośnikach telefonu. Usługa zawiera informacje o etapie realizacji zlecenia oraz ewentualnej liście potrzebnych części.
2. **Kategorie:** Przedstawiają różne typy usług oferowanych przez serwis, takie jak naprawa smartfonów, tabletów czy głośników, wraz z przykładowymi cenami.
3. **Klient:** Użytkownik korzystający z usług serwisu. Dzięki unikalnemu kodowi zlecenia i hasłu może sprawdzić status naprawy oraz przeglądać informacje związane z jego zgłoszeniem.
4. **Pracownik:** Osoba odpowiedzialna za zarządzanie zleceniami, przekazywanie i odbieranie urządzeń do naprawy oraz udzielanie wsparcia w razie problemów.
5. **Administrator:** Odpowiada za zarządzanie treścią wyświetlaną na stronie serwisu, danymi klientów i pracowników. Może dodawać nowych użytkowników z różnymi rolami, edytować ich dane oraz usuwać konta.

Atrybuty systemu:

- **Bezpieczeństwo:** Kluczowy aspekt systemu. Uwierzytelnianie użytkowników odbywa się poprzez tokeny JWT z określonym czasem wygaśnięcia, przechowywane w przeglądarce jako HttpOnlyCookie. Chroni to przed nieautoryzowanym dostępem do funkcji i danych.

- **Skalowalność:** System został zaprojektowany modułowo, co umożliwia łatwe wprowadzanie nowych funkcjonalności bez ingerencji w istniejące komponenty. Niezależność modułów sprzyja elastyczności i przyszłemu rozwojowi.
- **Wydajność:** Aplikacja dba o szybki czas reakcji, nieprzekraczający jednej sekundy, by zapewnić komfort użytkownika. Funkcje są optymalizowane i testowane pod kątem sprawności działania.

Funkcjonalności API:

- **Autentykacja:** Rejestrowanie, logowanie i wylogowywanie użytkowników z wykorzystaniem tokenów JWT i unikalnej 128-bitowej wartości dodawanej do hasła. Hasła są hashowane dla zwiększenia bezpieczeństwa.
- **Zarządzanie Danymi:** Dodawanie i modyfikacja informacji związanych ze zleceniami naprawy, usługami serwisu oraz treściami wyświetlanymi na stronie.
- **Generowanie Dokumentów:** Tworzenie faktur i innych dokumentów w formacie PDF, niezbędnych do obsługi zleceń i komunikacji z klientami.

Walidacja: Uniemożliwienie wprowadzania nieprawidłowych lub szkodliwych danych poprzez odpowiednie mechanizmy kontrolne.

Funkcjonalności Interfejsu Użytkownika:

- **Komunikacja z API:** Wysyłanie bezpiecznych zapytań HTTPS do API za pomocą biblioteki Axios, co zapewnia płynną interakcję z systemem.
- **Wyświetlanie Danych:** Prezentacja informacji w sposób przyjazny dla użytkownika, z poprawną interpretacją znaków UTF-8.
- **Powiadomienia:** Informowanie użytkowników o sukcesach i błędach operacji, takich jak poprawne logowanie czy niewłaściwa składnia hasła podczas rejestracji.
- **Walidacja Formularzy:** Wstępna kontrola wprowadzanych danych, by zminimalizować błędy i usprawnić procesy.

Wymagania Funkcjonalne

System umożliwia użytkownikom:

- **Rejestrację i Logowanie:** Tworzenie kont i bezpieczne logowanie do aplikacji.
- **Zarządzanie Zleceniami:** Tworzenie zleceń naprawy z unikalnymi kodami, dostęp do informacji o statusie naprawy oraz możliwość pobrania faktury w formacie PDF po zakończeniu usługi.
- **Edycję Treści:** Dla administratorów – możliwość modyfikowania zawartości tekstowej i graficznej strony, zarządzanie użytkownikami oraz ich uprawnieniami.

Wymagania Niefunkcjonalne:

- **Bezpieczeństwo:** Ścisłe zabezpieczenia przed atakami, ochrona haseł i danych wrażliwych.
- **Wydajność:** Szybkie wykonywanie zadań, z czasem reakcji poniżej jednej sekundy, nawet pod dużym obciążeniem.
- **Elastyczność:** Łatwość rozszerzenia aplikacji o nowe funkcjonalności i moduły.

Architektura Trójwarstwowa Klient-Serwer:

- **Interfejs Użytkownika:** Warstwa prezentacji dostępna w przeglądarce, odpowiedzialna za interakcję z użytkownikiem. Przechowuje tokeny autentykacyjne i komunikuje się z API.
- **REST API:** Warstwa logiki biznesowej, zawierająca funkcje obsługujące żądania, autoryzację, walidację oraz generowanie dokumentów. Zabezpiecza system poprzez mechanizmy CORS i kontrolę dostępu.
- **Serwer Danych:** Warstwa przechowywania danych, obsługująca zapytania do bazy po autentykacji klucza SSL.

Diagramy Przepływu Danych (DFD)

Diagramy DFD obrazują, jak dane przepływają w systemie i jakie operacje mogą wykonywać poszczególni użytkownicy. Klient może uzyskać informacje o statusie zlecenia i dostępnych usługach. Pracownik ma możliwość zarządzania zleceniami i generowania dokumentów, a

administrator dodatkowo zarządza danymi użytkowników i treścią strony.

Podsumowując, system integruje funkcjonalność i bezpieczeństwo, zapewniając intuicyjne narzędzia dla klientów, pracowników i administratorów. Dzięki modularnej budowie i trójwarstwowej architekturze jest przygotowany na przyszły rozwój i skalowanie, oferując solidne rozwiązanie dla serwisu urzędów elektronicznych.

4. Prototyp systemu wspomagania serwisu elektronicznego

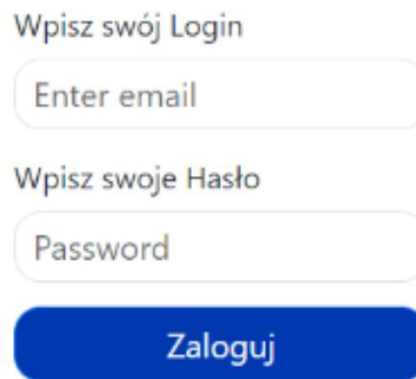
4.1 Opis stworzonej aplikacji

Aplikacja została zaprojektowana z myślą o pracownikach serwisu elektronicznego oraz ich klientach. Jest to trójwarstwowy system zbudowany zgodnie z architekturą RESTful. Moduły komunikują się z bazą danych za pośrednictwem API, a informacje o sesji są przechowywane wyłącznie po stronie przeglądarki użytkownika. Bezpieczeństwo logowania opiera się na tokenach JWT z rolami, umożliwiającymi wykonywanie określonych funkcji. Hasła są wzbogacane o 128-bitową wartość i haszowane przy użyciu algorytmu HMAC-SHA256.

Interfejs użytkownika wykorzystuje ikony z pakietu ReactIcons oraz komponenty z biblioteki ReactBootstrap, takie jak pasek nawigacji, tabele czy karuzela obrazków. Generowanie plików PDF realizowane jest za pomocą biblioteki React-pdf, która umożliwia tworzenie szablonów dokumentów wypełnionych danymi zleceń i klienta oraz ich eksport do formatu PDF.

4.2 Panel logowania

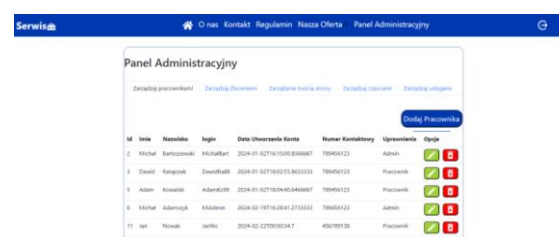
Po poprawnym zalogowaniu użytkownik otrzymuje komunikat powitalny i jest przekierowywany do strony odpowiadającej jego uprawnieniom zapisanym w tokenie JWT. W menu nawigacji pojawiają się dodatkowe zakładki, takie jak "Panel pracownika". Logowanie służy do uwierzytelniania i autoryzacji każdej czynności, aby uniemożliwić dostęp osobom nieuprawnionym.



Rys. 1. Okno logowanie pracownika do systemu.

Dane o uprawnieniach są pobierane z bazy danych wraz z innymi atrybutami, takimi jak ID pracownika. Informacje z tokenu są wykorzystywane np. przy tworzeniu zleceń naprawy, umożliwiając identyfikację pracownika odpowiedzialnego za zgłoszenie.

4.3 Panel administracyjny



Id	Imię	Nazwisko	Login	Data Utworzenia Konta	Numer Konta	Uprawnienia	Opcje
2	Michał	Karłowicz	Micha1Kar	2024-01-02T16:10:00Z	78904123	Admin	 
3	David	Kingman	DavidKall	2024-01-02T16:00:00Z	78904123	Pracownik	 
5	Adam	Kowalski	AdamK123	2024-01-02T16:00:00Z	78904123	Pracownik	 
6	Michał	Adamczyk	Micha1Ad	2024-01-02T16:00:00Z	78904123	Admin	 
11	Jan	Nowak	JanNo	2024-02-22T09:00:00Z	45678910	Pracownik	 

Rys. 2. Wygląd panelu administracyjnego.

Po zalogowaniu na konto administratora użytkownik zostaje przeniesiony do panelu administracyjnego. Panel umożliwia dodawanie pracowników do systemu, zmianę ich statusu aktywności, uprawnień oraz sortowanie według różnych kryteriów, np. nazwiska. Administrator może usuwać pracowników (jeśli nie mają aktywnych zleceń) lub zmieniać ich status na nieaktywny. Ma także możliwość zarządzania danymi klientów oraz edycji treści wyświetlanej na stronie, takiej jak opisy, kategorie usług i obrazy.

Rys. 3. Wygląd przykładowego okna dostępnych funkcji z panelu administratora.

W panelu dostępne są różne zakładki z funkcjami dostosowanymi do potrzeb administratora. Kolumna "Opcje" pozwala na edycję danych, dodawanie lub usuwanie użytkowników, generowanie faktur w formacie PDF na prośbę klienta czy dodawanie obrazów.

Podczas dodawania pracownika hasło podlega wstępnej walidacji i musi spełniać określone kryteria dotyczące długości oraz zawartości znaków. Imię, nazwisko, login i inne dane mają ustalone maksymalne długości. System umożliwia również edycję uprawnień, sortowanie pracowników oraz wyświetlanie kluczowych informacji, takich jak numer kontaktowy czy data zatrudnienia.

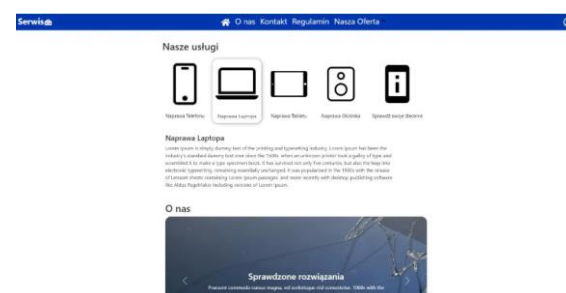
4.4 Panel pracowniczy

Po zalogowaniu na konto pracownika użytkownik zostaje przekierowany do panelu pracowniczego. Może tam tworzyć zlecenia naprawy, anulować je, zakończyć lub zmienić ich status. Na podstawie zleceń i danych klienta generowane są faktury lub dokumenty reklamacyjne. Pracownik może wyszukiwać zlecenia po kodzie oraz sortować je według daty czy innych kryteriów. Panel pracowniczy ma podobny wygląd do administracyjnego, ale z ograniczonymi funkcjonalnościami, skupionymi na zarządzaniu zleceniami i dokumentami.

Rys. 4. Wygląd okna opcji tworzenia zlecenia.

Przy tworzeniu zlecenia pracownik wybiera rodzaj urządzenia, wprowadza dane klienta oraz ustala dane logowania dla klienta do śledzenia zlecenia. Pola takie jak imię, nazwisko, numer telefonu, typ urządzenia i hasło są obowiązkowe. Hasło musi spełniać określone wymagania bezpieczeństwa.

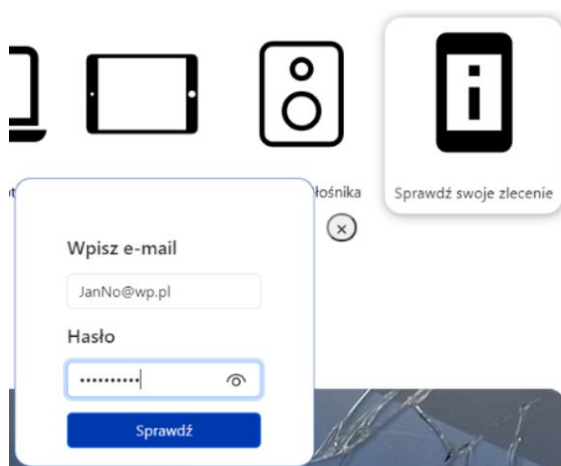
4.5 Możliwości klienta



Rys. 5. Wygląd strony głównej.

Klient odwiedzający stronę może przeglądać zakładki takie jak "Regulamin", "Kategorie" czy "Kontakt". Osoby bez uprawnień mogą jedynie przeglądać publiczne treści. Po zleceniu naprawy i wprowadzeniu danych przez pracownika, klient może zalogować się w sekcji "Sprawdź swoje zlecenie" za pomocą ustalonego maila i hasła. Otrzymuje wówczas dostęp do informacji o statusie zlecenia, opisie uszkodzenia oraz czynnościach naprawczych.

Na stronie głównej znajduje się sekcja "Nasze usługi", gdzie klient może wybierać interesujące go kategorie usług, a opisy są pobierane z bazy danych. W sekcji "O nas" umieszczono karuzelę z obrazkami i informacjami o firmie.



Rys. 6. Sprawdzenie zlecenia, wyświetlenie okna weryfikacji.

Klient może również korzystać z opcji "Sprawdź swoje zlecenie", gdzie po wprowadzeniu danych logowania zostaje przekierowany do strony zlecenia. Tam otrzymuje pełne informacje o zleceniu, statusie, podjętych działaniach i wycenie usługi. Kolor i nazwa statusu zmieniają się w zależności od etapu realizacji. Jeśli klient życzy sobie otrzymać fakturę VAT na firmę, wymagane jest uzupełnienie dodatkowych danych. Opcja pobrania faktury w formacie PDF pojawia się po zakończeniu zlecenia.

5. Testy poprawności działania funkcji Rest Api

Przeprowadzono testy integracyjne aplikacji serwisowej z wykorzystaniem narzędzia Postman. Celem było zweryfikowanie mechanizmów uwierzytelniania i autoryzacji użytkowników przy użyciu tokenów JWT. Poprawne logowanie skutkowało przydzieleniem **HttpOnlyCookie** z tokenem JWT, umożliwiającym dostęp do funkcji zgodnych z uprawnieniami użytkownika. W przypadku podania nieprawidłowych danych, serwer zwracał odpowiedni komunikat błędu, np. kod **400 Bad Request** przy błędnym hasle czy **403 Forbidden** przy braku uprawnień administracyjnych.

Testy jednostkowe REST API przeprowadzone za pomocą Swaggera potwierdziły poprawność działania kluczowych funkcji. Sprawdzono m.in. generowanie 128-bitowej wartości do hashowania haseł za pomocą algorytmu PBKDF2, enkrypcję i de-enkrypcję haseł, przyznawanie i weryfikację tokenów JWT w **HttpOnlyCookie**, a także operacje na bazie danych, takie jak dodawanie pracowników, zarządzanie treścią strony czy generowanie i obsługa dokumentów PDF.

Testy integracyjne wykazały, że wszystkie moduły aplikacji współdziałają zgodnie z założeniami. System poprawnie obsługuje zarówno prawidłowe scenariusze, zwracając kod **200 OK**, jak i

nieprawidłowe, blokując dostęp i zwracając odpowiednie kody błędów, takie jak **401 Unauthorized** czy **403 Forbidden**.

W testach akceptacyjnych wszystkie moduły, w tym rejestracja, logowanie, tworzenie zleceń, wystawianie faktur oraz zarządzanie danymi pracowników i klientów, zostały pozytywnie ocenione przez klienta i zaakceptowane do wdrożenia.

Wnioski: Aplikacja spełnia założone wymagania funkcjonalne i bezpieczeństwa. Mechanizmy uwierzytelniania i autoryzacji działają poprawnie, skutecznie chroniąc przed nieautoryzowanym dostępem. Wszystkie testy zakończyły się pomyślnie, potwierdzając gotowość systemu do implementacji w środowisku produkcyjnym.

6. Podsumowanie

Aplikację można rozwinąć o moduł sprzedaży odnowionej elektroniki od klientów, moduł wyceny dostarczonych urządzeń, integrację z API firmy kurierskiej oraz moduł ocen i komentarzy z moderacją. Zauważono, że każde żądanie zużywa od kilkudziesięciu do kilkuset bajtów pamięci; przy wielu zapytaniach konieczna będzie odpowiednia infrastruktura lub hosting. Trójwarstwowa architektura ułatwia rozwój bez wpływu na inne komponenty, co jest jej zaletą. Testy potwierdziły prawidłowe działanie aplikacji, spełnienie założeń, dobrą wydajność i pozytywne doświadczenia użytkowników.

Literatura

- [1]. <https://www.raport-recommerce.pl/>
- [2] <https://www.idc.com/getdoc.jsp?containerId=prUS50565723>
- [3] <https://www.idc.com/getdoc.jsp?containerId=prUS51295423>
- [4] <https://stat.gov.pl/obszary-tematyczne/nauka-i-technikaspoleczenstwo-informacyjne/spoleczenstwo-informacyjne/spoleczenstwo-w-polsce-w-2023-roku,2,13.html>
- [5] https://www.uke.gov.pl/download/gfx/uke/pl/defaultaktualnosci/36/485/4/uke_raport_tele_2022_2.pdf
- [6] https://stat.gov.pl/download/gfx/portalinformacyjny/pl/defaultaktualnosci/5497/1/16/1/si_2022_003.pdfmln-ogloszen-w-pol-roku
- [7] https://stat.gov.pl/cps/rde/xbcr/gus/nts_spolecz_inform_w_polsce-2013.pdf

- [8] [https://learn.microsoft.com/pl-pl/previous-versions/ms533046\(v=vs.85\)?redirectedfrom=MSDN](https://learn.microsoft.com/pl-pl/previous-versions/ms533046(v=vs.85)?redirectedfrom=MSDN)
- [9] <https://learn.microsoft.com/pl-pl/dotnet/csharp/tour-of-csharp/>
- [10] <https://jwt.io/introduction>
- [11] Common web application architectures - .NET | Microsoft Learn
- [12] <https://learn.microsoft.com/pl-pl/aspnet/core/security/cors?view=aspnetcore-7.0>
- [13] <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>
- [14] <https://axios-http.com/docs/intro>
- [15] <https://learn.microsoft.com/pl-pl/dotnet/standard/serialization/>