

POLITECHNIKA _____

Wydział Informatyki — Kierunek: Informatyka

Jan Kowalski

nr albumu 000000

Projekt i implementacja aplikacji webowej do zarządzania biblioteką

Praca magisterska

napisana pod kierunkiem

dr. inż. _____

[Miasto], 2026

Jak korzystać z tego wzoru. To kompletny szkielet pracy magisterskiej z informatyki o charakterze **projektowo-implementacyjnym** (zaprojektowanie i wykonanie systemu) z przykładową treścią i komentarzami. Niebieskie adnotacje opisują funkcję każdej części — usuń je w swojej pracy. Zwróć uwagę na analizę wymagań zamiast hipotez, projekt (UML, model danych), opis implementacji i testy — to cechy charakterystyczne pracy inżynierskiej/informatycznej. Pełne poradniki: praca-magisterska.pl/poradniki/

Streszczenie

Praca dotyczy projektu i implementacji aplikacji webowej wspierającej zarządzanie biblioteką. Celem pracy jest zaprojektowanie, wykonanie i przetestowanie systemu umożliwiającego katalogowanie zbiorów, obsługę wypożyczeń oraz zarządzanie kontami czytelników. Przeprowadzono analizę wymagań funkcjonalnych i nie-funkcjonalnych, zaprojektowano architekturę i model danych, a następnie zaimplementowano aplikację w architekturze klient–serwer. System poddano testom jednostkowym, integracyjnym i wydajnościowym. Wynikiem pracy jest działający prototyp spełniający przyjęte wymagania wraz z dokumentacją i wskazaniem kierunków dalszego rozwoju.

Słowa kluczowe: aplikacja webowa, inżynieria oprogramowania, system biblioteczny, architektura klient–serwer, testowanie.

[W informatyce streszczenie opisuje cel inżynierski (co zaprojektowano i zbudowano), zastosowane podejście i wynik (działający system). Dołącz wersję angielską (Abstract), jeśli wymaga jej uczelnia.]

Spis treści

Wstęp	4
Rozdział 1. Analiza problemu i przegląd rozwiązań	6
1.1. Charakterystyka problemu	6
1.2. Przegląd istniejących rozwiązań	9
1.3. Analiza wymagań funkcjonalnych i niefunkcjonalnych	12
Rozdział 2. Projekt systemu	16
2.1. Architektura rozwiązania	16
2.2. Model danych	20
2.3. Projekt interfejsu użytkownika	24
Rozdział 3. Implementacja	27
3.1. Zastosowane technologie	27
3.2. Realizacja kluczowych funkcjonalności	30
Rozdział 4. Testowanie i ocena rozwiązania	35
4.1. Plan i rodzaje testów	35
4.2. Wyniki testów	39
Zakończenie i kierunki rozwoju	42
Bibliografia · Spis rysunków i listingów	44

Wstęp

[Akapit 1 — kontekst i uzasadnienie potrzeby zbudowania systemu.]

Cyfryzacja procesów obsługi instytucji publicznych objęła również biblioteki, w których ręczne prowadzenie ewidencji zbiorów i wypożyczeń jest czasochłonne i podatne na błędy. Dedykowana aplikacja webowa pozwala zautomatyzować te procesy i udostępnić katalog czytelnikom online.

[Akapit 2 — cel inżynierski pracy. W informatyce cel formułuje się jako zaprojektowanie i wykonanie systemu, nie jako weryfikację hipotez.]

Celem pracy jest zaprojektowanie, implementacja i przetestowanie aplikacji webowej wspierającej zarządzanie biblioteką w zakresie katalogowania zbiorów, obsługi wypożyczeń i zarządzania kontami czytelników.

[Akapit 3 — zakres i struktura pracy.]

Praca obejmuje pełny cykl wytwarzania oprogramowania: analizę wymagań, projekt, implementację i testy. Składa się z czterech rozdziałów odpowiadających kolejnym etapom tego procesu, zakończonych oceną rozwiązania i wskazaniem kierunków rozwoju.

Rozdział 1. Analiza problemu i przegląd rozwiązań

1.3. Analiza wymagań funkcjonalnych i нефункциональных

[W informatyce odpowiednikiem hipotez są WYMAGANIA. Dzieli się je na funkcjonalne (co system ma robić) i нефункциональные (jak ma działać: wydajność, bezpieczeństwo, użyteczność). To one wyznaczają zakres pracy i kryteria oceny.]

Wymagania funkcjonalne: rejestracja i logowanie użytkowników, katalogowanie pozycji, wyszukiwanie zbiorów, rejestracja wypożyczeń i zwrotów, zarządzanie kontami czytelników, generowanie raportów.

Wymagania нефункциональные: czas odpowiedzi interfejsu poniżej 1 s, obsługa co najmniej 100 równoczesnych użytkowników, szyfrowanie haseł, responsywność interfejsu, zgodność z przeglądarkami Chrome, Firefox i Edge.

Rozdział 2. Projekt systemu

2.1. Architektura rozwiązania

System zaprojektowano w architekturze klient–serwer z podziałem na warstwę prezentacji (aplikacja kliencka), warstwę logiki (API) oraz warstwę danych (relacyjna baza danych). Komunikację między klientem a serwerem oparto na interfejsie REST.

[Projekt warto zilustrować diagramami UML (przypadków użycia, klas, sekwencji) oraz diagramem architektury. W tekście opisz decyzje projektowe i ich uzasadnienie — to istota części projektowej.]

2.2. Model danych

Model danych obejmuje encje: Użytkownik, Książka, Egzemplarz, Wypożyczenie. Relacje między nimi przedstawia diagram związków encji (ERD). Tabelę kluczowych encji zawiera tabela 2.1.

Tabela 2.1. Wybrane encje modelu danych

Encja	Kluczowe atrybuty
Użytkownik	id, login, hash_hasła, rola
Książka	id, tytuł, autor, ISBN
Wypożyczenie	id, id_egzemplarza, id_uzytkownika, data_zwrotu

Rozdział 3. Implementacja

3.1. Zastosowane technologie

Warstwę kliencką zaimplementowano w bibliotece React, warstwę serwerową w środowisku Node.js (framework Express), a dane przechowywano w relacyjnej bazie PostgreSQL. Kontrolę wersji prowadzono w systemie Git, a aplikację skonteneryzowano w Dockerze.

[Opis stosu technologicznego uzasadnij — dlaczego te technologie. Kod prezentuj fragmentami (listingami) kluczowych funkcji, nie całymi plikami. Każdy listing numeruj i opatrz podpisem.]

Rozdział 4. Testowanie i ocena rozwiązania

4.2. Wyniki testów

System przetestowano na trzech poziomach. Testy jednostkowe pokryły logikę biznesową, testy integracyjne — współpracę warstw, a testy wydajnościowe sprawdziły zachowanie pod obciążeniem. Wyniki przedstawia tabela 4.1.

Tabela 4.1. Wybrane wyniki testów

Rodzaj testu	Miara	Wynik
Jednostkowe	pokrycie kodu	84%
Wydajnościowe	śr. czas odpowiedzi (100 użytk.)	320 ms
Wydajnościowe	przepustowość	280 żądań/s

Źródło: opracowanie własne.

[Ocena rozwiązania polega na zestawieniu wyników testów z WYMAGANIAMI z rozdziału 1. Pokaż, które wymagania spełniono i w jakim stopniu — to odpowiednik weryfikacji hipotez w pracach empirycznych.]

Zakończenie i kierunki rozwoju

Zaprojektowano i zaimplementowano działający prototyp aplikacji webowej do zarządzania biblioteką, który spełnił przyjęte wymagania funkcjonalne i нефункционалне. Testy potwierdziły poprawność działania i wy-

dajność w założonym zakresie. Kierunki dalszego rozwoju obejmują integrację z zewnętrznymi katalogami bibliotecznymi, aplikację mobilną oraz moduł rekomendacji oparty na historii wypożyczeń.

Bibliografia

[W informatyce często stosuje się styl IEEE (numeryczny) lub APA. Powołuj się na dokumentację technologii, normy i literaturę z inżynierii oprogramowania. Sprawdź wytyczne uczelni.]

1. Sommerville, I. (2016). *Inżynieria oprogramowania*. Warszawa: WNT.
2. Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley.
3. Dokumentacja React, <https://react.dev> [dostęp: 2026].

O tym wzorze

Wzór przygotowała redakcja serwisu **praca-magisterska.pl**. Możesz swobodnie wykorzystać jego strukturę, tabele i przypisy we własnej pracy. Pamiętaj tylko, żeby treść dostosować do wymagań swojego promotora i uczelni.

Smart-Edu.ai — pierwsza wersja pracy w kilka godzin, nie w kilka miesięcy

Podajesz temat, kierunek studiów i wytyczne promotora, a sztuczna inteligencja generuje unikalny, gotowy do edycji tekst: z poprawną strukturą rozdziałów, przypisami i bibliografią. Rejestracja jest darmowa, płacisz wyłącznie za wygenerowany tekst.

www.smart-edu.ai

Wolisz napisać pracę w pełni samodzielnie? Pomoże Ci ebook **„Jak napisać pracę magisterską. Kompletny przewodnik od A do Z”** — 250 stron praktycznych wskazówek, szablonów i checklist, z rozdziałem o etycznym korzystaniu z AI: www.praca-magisterska.pl/sklep/praca-magisterska-ebook/

*Darmowe poradniki, wzory i ponad 2500 tematów prac: **www.praca-magisterska.pl***