

Dokumentacja projektu wykonywanego w ramach zajęć BAZY DANYCH I

System zarządzania księgarnią internetową

I. Projekt koncepcji, założenia (4pkt.)

1. Zdefiniowanie tematu projektu:

Zrealizowanie bazy danych przechowującej informacje typowe dla księgarni internetowej z systemem oceniania książek przez klientów, oraz aplikacji webowej, która zasymuluje podstawowe funkcjonalności panelu administratora takiej księgarni.

2. Analiza wymagań użytkownika:

Założeniem projektu jest to, że do aplikacji webowej dostęp będzie miał tylko administrator (będzie ona hostowana lokalnie). Zarządzanie bazą danych (dostępną dla pracowników i klientów księgarni) będzie odbywać się poprzez API napisane w stylu RESTful, co w znacznym stopniu ułatwi dalszy rozwój aplikacji. Takie rozwiązanie poprawi wydajność w zarządzaniu większą ilością danych poprzez ich wizualizację oraz szybsze wykrywanie i rozwiązywanie problemów.

Administrator w podstawowej wersji aplikacji webowej będzie miał możliwość

- Bezpośredniego wglądu do „surowych danych” w każdej encji.
- Bezpośredniego wprowadzania „surowych danych” do każdej encji.
- Dodawania nowych książek wraz z autorami i wydawnictwem w specjalnie do tego przygotowanym formularzu.
- Dodawania nowych magazynów w specjalnie do tego przygotowanym formularzu.
- Dodawania książek do magazynów w specjalnie do tego przygotowanym formularzu.
- Wgląd do średniej ocen książek i autorów
- Wgląd do ilości książek sprzedanych przez dane wydawnictwo.
- Wgląd do najpopularniejszych książek.
- Wgląd do najpopularniejszych autorów.
- Wglądu do stanu magazynowego książek.
- Informację jeżeli w magazynach jest mniej niż 100 szt. danej książki oraz dane do kontaktu z wydawnictwem.

3. Zaprojektowanie funkcji:

Aby zapewnić integralność danych w tabeli zaprojektowane zostały specjalne funkcje:

- W encji klient e-mail jest wartością unikalną, umożliwia to stworzenie systemu autoryzacji w księgarni.
- W encji książka ISBN jest wartością unikalną.
- Funkcjonalności dostępne w „Zaawansowane Formularze”
 - Podczas dodawania książek do magazynu następuje sprawdzenie, czy dane książki są już przechowywane w magazynie, jeżeli tak, to zostaną one dodane do aktualnej ilości, zamiast tworzyć nowy rekord. Wykonywana jest też podstawowa walidacja danych,

jeżeli dane pole nie spełnia wymagań to odpowiednia wiadomość zostaje wysłana do klienta.

- Podczas dodawania nowego magazynu następuje sprawdzenie, czy podany magazyn, adres oraz kierownik już istnieją, jeżeli tak, to zostaną one użyte. Ma to na celu zredukowanie ilości rekordów w tabelach. Podczas dodawania następują podstawowa walidacja danych.
- Podczas dodawania książek do magazynu wyboru dokonuje się z listy. Jeżeli dana książka znajduje już się w danym magazynie, to zamiast tworzyć nowy rekord nowe książki zostaną dodane do istniejącej wartości.

II. Projekt diagramów (konceptualny) (4pkt.)

~~4. Budowa i analiza diagramu przepływu danych (DFD):~~

5. Zdefiniowanie encji (obiektów) oraz ich atrybutów:

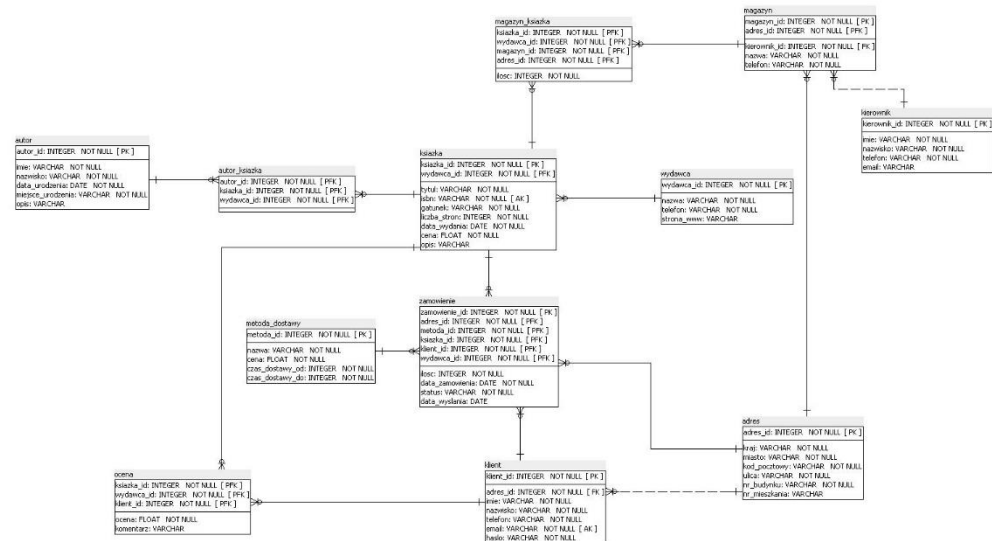
- Autor: autor_id – klucz podstawowy, imie, nazwisko, data_urodzenia, opis
- Książka: ksiazka_id – klucz podstawowy, wydawca_id – podstawowy klucz obcy, tytuł, isbn, gatunek, liczba_stron, data_wyadani, cena, opis
- Wydawca: wydawca_id – klucz podstawowy, nazwa, telefon, strona_www
- Magazyn: magazyn_id – klucz podstawowy, adres_id – podstawowy klucz obcy, kierownik_id – klucz obcy, nazwa, telefon
- Kierownik: kierownik_id – klucz podstawowy, imie, nazwisko, telefon, email
- Adres: adres_id – klucz podstawowy, kraj, miasto, kod_pocztowy, ulica, nr_budynku, nr_mieszkania
- Klient: klient_id – klucz podstawowy, adres_id – klucz obcy, imie, nazwisko, telefon, email, haslo
- Ocena: ksiazka_id – podstawowy klucz obcy, wydawca_id – podstawowy klucz obcy, klient_id – podstawowy klucz obcy, ocena, komentarz
- Zamówienie: zamówienie_id – klucz podstawowy, adres_id – podstawowy klucz obcy, metoda_id - podstawowy klucz obcy, ksiazka_id – podstawowy klucz obcy, klient_id – podstawowy klucz obcy, wydawca_id – podstawowy klucz obcy, ilość, data_zamowienia, status, data_wyslania
- Metoda Dostawy: metoda_id – klucz podstawowy, nazwa, cena, czas_dostawy_od, czas_dostawy_do

6. Zaprojektowanie relacji pomiędzy encjami:

Relacje zostały opisane w kolejnej sekcji w punkcie 7.

III. Projekt logiczny (4pkt.)

7. Projektowanie tabel, kluczy, indeksów:



Autor: posiada atrybuty opisujące oraz połączony jest relacją n:n z książką, gdyż jedna książka może mieć wielu autorów, a jeden autor może napisać wiele książek.

Książka: posiada atrybuty opisujące oraz połączona jest relacją n:n z magazynem, ponieważ w jednym magazynie może być wiele książek, a jedna książka może być w wielu magazynach rozsianych na terenie całego kraju. W relacji tej zawarta jest również informacja o ilości sztuk danej książki w magazynie. Posiada również ID wydawcy danej książki.

Wydawca: posiada atrybuty opisujące oraz połączony jest z książką relacją 1 : n, ponieważ jeden wydawca może wydać wiele książek, ale jedna książka może być wydana tylko przez jednego wydawcę.

Magazyn: posiada atrybuty go opisujące, ID adresu oraz ID kierownika. Połączony jest relacją n:n z książką (opisane wyżej).

Kierownik: posiada atrybuty go opisujące, połączony jest relacją 1:n z magazynem, ponieważ jeden kierownik może zarządzać wieloma magazynami znajdującymi się w bliskiej odległości od siebie

Adres: Posiada informacje go opisujące oraz połączony jest relacjami 1:n z magazynem, zamówieniem oraz klientem, ponieważ jeden adres może zostać użyty więcej niż jeden raz.

Klient: Posiada atrybuty go opisujące, ID adresu oraz połączony jest relacją Ocena (n:n) z Książką, ponieważ jeden klient może ocenić wiele książek, a jedna książka może zostać oceniona przez wielu klientów.

Zamówienie: Posiada atrybuty je opisujące, atrybuty opisujące zamawianą książkę, adres oraz klienta.

Metoda Dostawy: Posiada atrybuty go opisujące, oraz połączony jest relacją 1:n z zamówieniem, ponieważ jedna metoda dostawy może zostać użyta w wielu zamówieniach.

8. Słowniki danych:

- Autor:
 - autor_id – klucz podstawowy
 - imie – VARCHAR NOT NULL
 - nazwisko – VARCHAR NOT NULL
 - data_urodzenia – DATE NOT NULL
 - miejsce_urodzenia - VARCHAR NOT NULL
 - Opis - VARCHAR
- Autor_Książka:
 - autor_id – podstawowy klucz obcy
 - ksiazka_id – podstawowy klucz obcy
 - wydawca_id - podstawowy klucz obcy
- Książka:
 - ksiazka_id – klucz podstawowy
 - wydawca_id – podstawowy klucz obcy
 - tytuł – VARCHAR NOT NULL
 - isbn – VARCHAR (wersja 10 cyfrowa lub 13 cyfrowa) NOT NULL UNIQUE
 - gatunek – VARCHAR NOT NULL
 - liczba_stron – INTEGER NOT NULL
 - data_wydania – DATE NOT NULL
 - cena – FLOAT NOT NULL
 - opis - VARCHAR
- Wydawca:
 - wydawca_id - klucz podstawowy
 - nazwa – VARCHAR NOT NULL
 - telefon – VARCHAR (3cyfry-3cyfry-3cyfry) NOT NULL
 - strona_www - VARCHAR
- Magazyn_Książka:
 - ksiazka_id – podstawowy klucz obcy
 - wydawca_id – podstawowy klucz obcy
 - magazyn_id – podstawowy klucz obcy
 - adres_id - podstawowy klucz obcy
 - ilosc – INTEGER NOT NULL
- Magazyn:
 - magazyn_id – klucz podstawowy
 - adres_id – podstawowy klucz obcy
 - kierownik_id – podstawowy klucz obcy
 - nazwa – VARCHAR NOT NULL
 - telefon – VARCHAR (3cyfry-3cyfry-3cyfry) NOT NULL
- Kierownik:
 - kierownik_id – klucz podstawowy
 - imie – VARCHAR NOT NULL
 - nazwisko - VARCHAR NOT NULL

- telefon – VARCHAR (3cyfry-3cyfry-3cyfry) NOT NULL
 - email - VARCHAR
- Adres:
 - adres_id – klucz podstawowy
 - kraj - VARCHAR NOT NULL
 - miasto - VARCHAR NOT NULL
 - kod_pocztowy - VARCHAR NOT NULL
 - ulica - VARCHAR NOT NULL
 - nr_budynku - VARCHAR NOT NULL
 - nr_mieszkania - VARCHAR NOT NULL
- Klient:
 - klient_id – klucz podstawowy
 - adres_id – klucz obcy
 - imie - VARCHAR NOT NULL
 - nazwisko - VARCHAR NOT NULL
 - telefon – VARCHAR (9 cyfr) NOT NULL
 - email - VARCHAR NOT NULL UNIQUE
 - haslo - VARCHAR NOT NULL
- Ocena:
 - ksiazka_id - podstawowy klucz obcy
 - wydawca_id - podstawowy klucz obcy
 - klient_id - podstawowy klucz obcy
 - ocena - VARCHAR NOT NULL
 - komentarz - VARCHAR NOT NULL
- Zamówienie
 - zamowienie_id - podstawowy klucz obcy
 - adres_id - podstawowy klucz obcy
 - metoda_id - podstawowy klucz obcy
 - ksiazka_id - podstawowy klucz obcy
 - klient_id - podstawowy klucz obcy
 - wydawca_id - podstawowy klucz obcy
 - ilosc – INTEGER NOT NULL
 - data_zamowienia – DATE NOT NULL
 - status – VARCHAR NOT NULL („Przygotowywane” , „Wysłane”)
 - data_wyslania – DATE NOT NULL
- Metoda_dostawy:
 - metoda_id – klucz podstawowy
 - nazwa – VARCHAR NOT NULL
 - cena – VARCHAR NOT NULL
 - czas_dostawy_od – INTEGER NOT NULL
 - czas_dostawy_do – INTEGER NOT NULL

11. Zaprojektowanie operacji na danych:

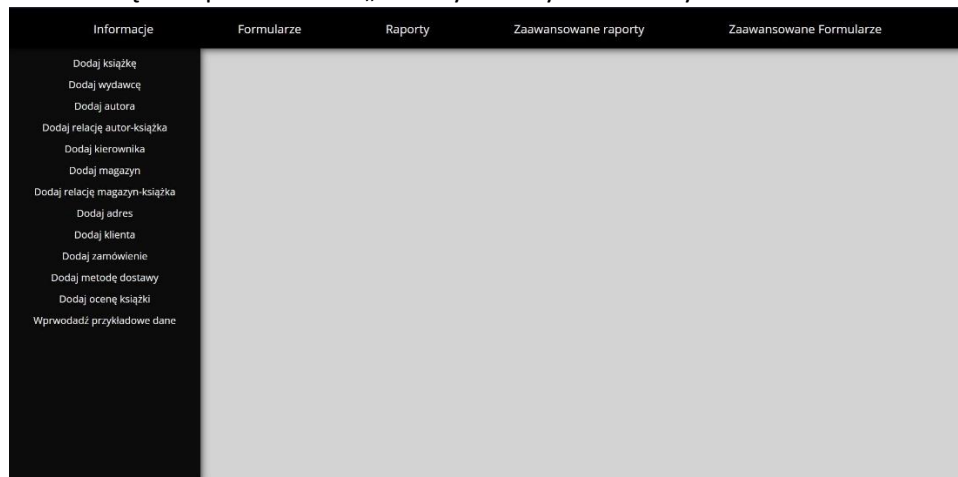
Użyte widoki oraz funkcje znajdują się w folderze SQL – plik ViewsAndFunctions.sql

IV. Projekt funkcjonalny (4pkt.)

1. Strona główna z panelem sterowania



2. Przycisk Formularze powoduje przejście do panelu nawigacyjnego z formularzami, które służą do wprowadzania „surowych danych” do bazy.



Przykładowy formularz:

This screenshot shows the 'Dodaj książkę' (Add book) form within the 'Formularze' section. The form is displayed on a light gray background. It includes the following fields and controls:

- Tytuł książki:
- Wydawca ID:
- ISBN:
- Gatunek:
- Liczba stron:
- Data wydania:
- Cena:
- Opis książki:
-

- Przycisk Raporty powoduje przejście do panelu nawigacyjnego z raportami, które wyświetlają zawartość poszczególnych tabeli w bazie.

Informacje	Formularze	Raporty	Zaawansowane raporty	Zaawansowane Formularze
Książki				
Wydawcy				
Autorzy				
Autorzy i Książki				
Kierownicy				
Magazyny				
Magazyny i Książki				
Adresy				
Klienci				
Zamówienia				
Metody dostawy				
Oceny Książek				

Przykładowy raport:

Zaawansowane raporty								
Informacje		Formularze		Raporty	Zaawansowane raporty	Zaawansowane Formularze		
	Książki	Adres ID	Kraj	Miasto	Kod pocztowy	Ulica	Numer budynku	Numer mieszkania
Wydawcy		1	Polska	Warszawa	00-012	Marszałkowska	35A	12
		2	Polska	Warszawa	00-012	Żłota	123B	5
Autorzy		3	Polska	Warszawa	00-012	Świętokrzyska	3	12
		4	Polska	Warszawa	00-012	Marszałkowska	68	
Autorzy i Książki		5	Polska	Kraków	30-007	Aleja Juliusza Słowackiego	65A	12b
		6	Polska	Kraków	30-007	Aleja Juliusza Słowackiego	65	3
Kierownicy		7	Polska	Kraków	30-007	Mazowiecka	56	
		8	Polska	Kraków	30-007	Aleja Juliusza Słowackiego	23	
Magazyny		9	Polska	Katowice	40-001	Batantów	144B	
		10	Polska	Katowice	40-001	Wietowa	23	3A
Magazyny i Książki		11	Polska	Katowice	40-001	Radockiego	76	44
		12	Polska	Wrocław	45-573	Gazowa	14	
Adresy		13	Polska	Wrocław	45-573	Kawalska	22A	
		14	Niemcy	Berlin	10178	Spandauer	10C	33
Klienci		15	Niemcy	Lipsk	04103	Numberger	33	
		16	Polska	Gdańsk	80-011	Kątowa	87	11
Zamówienia		17	Polska	Poznań	60-001	Wykopy	65	
		18	Czechy	Praga	10200	Medanikowa	432	22
Metody dostawy		19	Czechy	Brno	60200	Kotlarska	43	
		20	Niemcy	Drezno	01667	Freiberger	32	
Oceny Książek		21	Polska	Łódź	90-005	Nawrot	42H	
		22	Niemcy	Monachium	80637	Numberger	19	24
		23	Polska	Kielce	25-011	Kapitulna	56	4
		24	Niemcy	Bremn	28205	Am Hulsberg	51B	14
		25	Słowacja	Bratysława	2412	Bratka	21A	52
		26	Polska	Olsztyn	10-008	Feliksa Strabera	31	14
		27	Polska	Rzeszów	35-000	jarowa	23	32
		28	Niemcy	Hamburg	20257	Weckmannweg	22E	32
		29	Polska	Białystok	15-005	Henrika Sienkiewicza	14	44
		30	Polska	Opole	45-006	Katedralna	55	41
		31	Niemcy	Wittenber	06889	Grabo	10F	
		32	Polska	Katowice	41-000	Ulica	32	44

- Przycisk zaawansowanie raporty powoduje przejście do panelu nawigacyjnego z raportami, które wyświetlają najważniejsze informacje potrzebne administratorowi księgarni internetowej.

Informacje	Formularze	Raporty	Zaawansowane raporty	Zaawansowane Formularze
Najlepsze książki				
Najlepsi autorzy				
Najpopularniejsze książki				
Najpopularniejsi autorzy				
Zarobki wydawców				
Małe stany magazynowe				

- Przycisk Zaawansowane Formularze powoduje przejście do formularzy, które gwarantują wprowadzenie danych do bazy w sposób względnie bezpieczny. Używanie ich powoduje ograniczenie duplikacji danych w bazie, gdyż korzystają one z danych już wprowadzonych w bazie, jeżeli pokrywają się z nowymi danymi.

Informacje	Formularze	Raporty	Zaawansowane raporty	Zaawansowane Formularze
Dodaj książkę Dodaj magazyn Dodaj książki do magazynu				

6. Użytkownik końcowy jest informowany, czy dane do bazy zostały wprowadzone pomyślnie.

Informacje	Formularze	Raporty	Zaawansowane raporty	Zaawansowane Formularze
Dodaj książkę Dodaj magazyn Dodaj książki do magazynu	<div> Książka: 1: Duma i uprzedzenie, 0-2468-8405-3 Magazyn: 1: Wiatr Magazyn, 865-323-124 Ilość: 1000 Dodaj dane </div> <p>Rekord został dodany do bazy!</p>			

7. Użytkownik końcowy jest informowany, jeżeli podczas wprowadzania danych pojawił się błąd, oraz jest informowany o rodzaju błędu.

Informacje	Formularze	Raporty	Zaawansowane raporty	Zaawansowane Formularze
Dodaj książkę Dodaj magazyn Dodaj książki do magazynu	<div> Nazwa magazynu: Magazyn Telefon do magazynu: 888-888-888 Imię kierownika: Mirosław Nazwisko kierownika: Kowalski Telefon kierownika: 777-777-777 E-mail kierownika: kowalski@gmail.com Kraj: Polska Miasto: Warszawa Kod pocztowy: 00-019 Ulica: Marszałkowska Numer budynku: 56 Numer mieszkania: Dodaj dane </div> <p>Wystąpił błąd podczas dodawania rekordu: error: Magazyn pod tym adresem już istnieje!</p>			

V. Dokumentacja (4pkt.)

16. Wprowadzanie danych:

Dane do każdej tabeli, oprócz tabeli autor_książka, muszą zostać wprowadzone ręcznie.

17. Dokumentacja użytkownika:

Jak uruchomić aplikację:

1. Pierwszym krokiem jest stworzenie bazy danych PostgreSQL, której struktura znajduje się w folderze SQL w pliku Structure.sql oraz dodanie do niej widoków i funkcji z pliku ViewsAndFunctions.sql
2. Należy pobrać ze strony <https://nodejs.org/en/> node.js (najlepiej wersję 16.13.2 – na niej tworzony był projekt)
3. Z poziomu konsoli wejść do folderu backend oraz zainstalować potrzebne moduły za pomocą komendy **npm install**
4. Przed uruchomieniem serwera należy skonfigurować dane logowania do bazy PostgreSQL. Plik konfiguracyjny znajduje się w folderze backend: config.js:
dbHost – Nazwa hosta
dbPort – Port, na którym działa baza danych
dbUser – Nazwa użytkownika
dbPass: – Hasło do bazy danych
dbName: – Nazwa bazy danych
5. Kolejnym krokiem jest uruchomienie serwera: **node .\server.js**
Serwer należy uruchomić na porcie 8000, w innym wypadku należy ręcznie zmienić port we wszystkich url w części frontend-owej
6. Następnie w nowej konsoli należy wejść do folderu frontend i użyć polecenia **npm install** w celu pobrania wymaganych modułów
7. Następnie należy użyć komendy **npm start** w celu uruchomienia serwera. Zostanie on uruchomiony na porcie 3000.

Do aplikacji dołączone są przykładowe dane, które można wprowadzić bezpośrednio w bazie danych (plik Data.sql w folderze SQL) lub poprzez aplikację webową (Formularze -> Przykładowe dane). Najlepiej dane wprowadzić z poziomu aplikacji webowej w celu uniknięcia problemów z kodowaniem.

Jeżeli już raz się wprowadziło przykładowe dane nie należy robić tego ponownie! Operacja ta nie jest w żadnym stopniu zabezpieczona w aplikacji webowej.

18. Opracowanie dokumentacji technicznej:

Aplikacja została napisana w stylu RESTful, istnieje więc podział na backend, który komunikuje się z frontendem poprzez zapytania GET oraz POST.

- Frontend:

Stworzony został przy użyciu REACT'a (wersja 17.0.2). Każdy komponent jest odpowiedzialny za osobną stronę w aplikacji

- Index.js – jest to plik startowy, odpowiada za inicjalizację aplikacji
- App.js – służy jako pośrednik w nawigowaniu aplikacją

Folder layout:

- Layout.js – odpowiedzialny jest za układ strony
- MainNavigation.js – główny panel nawigacyjny aplikacji
- Information.js – strona informacyjna
- FormsNavigation.js – nawigacja po prostych formularzach, które pozwalają wprowadzić dane bezpośrednio do bazy
- ReportsNavigation.js – nawigacja po prostych raportach, które pokazują zawartość bazy danych
- AdvancedFormNavigation.js – nawigacja po zaawansowanych formularzach, dane do bazy danych są wprowadzane poprzez funkcje
- AdvancedReportsNavigation.js nawigacja po zaawansowanych raportach, dane pochodzą z widoków
- *.module.css – Pliki o tej samej nazwie jak komponenty odpowiadają za ich wygląd

Folder reports:

Każdy plik odpowiedzialny jest za wysyłanie zapytania do serwera metodą Get i wyświetlenia otrzymanych informacji w postaci graficznej.

- AddressesReport.js – „Surowe dane” z tabeli adres
- AuthorsBooksrepor.js – „Surowe dane” z tabeli autor_ksiazka
- BooksReport.js – „Surowe dane” z tabeli ksiazka
- ClientReport.js – „Surowe dane” z tabeli klient
- DeliveryMethodsReport.js – „Surowe dane” z tabeli metoda_destowy
- ManagersReport.js – „Surowe dane” z tabeli kierownik
- OrdersReport.js – „Surowe dane” z tabeli zamowienie
- PublisherReport.js – „Surowe dane” z tabeli wydawca
- ScoreReport.js – „Surowe dane” z tabeli ocena
- WarehousesBooksReport.js – „Surowe dane” z tabeli magazyn_ksiazka
- WareHousesReport.js – „Surowe dane” z tabeli magazyn
- Reports.module.css – wygląd wyświetlonych danych

Folder forms:

Każdy plik odpowiedzialny jest za wysyłanie danych do serwera metodą POST i wyświetlenia otrzymanej odpowiedzi w postaci graficznej pod formularzem.

- AddAddressForm.js – Dodanie „surowych danych” do tabeli adres
- AddAuthorBookForm.js – Dodanie „surowych danych” do tabeli autor_ksiazka
- AddAuthorForm.js – Dodanie „surowych danych” do tabeli autor
- AddBookForm.js – Dodanie „surowych danych” do tabeli ksiazka
- AddClientForm.js – Dodanie „surowych danych” do tabeli klient
- AddDeliveryMethodForm.js – Dodanie „surowych danych” do metoda_dostawy

- AddExemappleDataForm.js – Dodanie przykładowych danych do wszystkich tabel w bazie danych
- AddManagerForm.js – Dodanie “surowych danych” do tabeli kierownik
- AddOrderForm.js – Dodanie “surowych danych” do tabeli zamówienie
- AddPublisherForm.js – Dodanie “surowych danych” do tabeli wydawca
- AddScoreForm.js – Dodanie “surowych danych” do tabeli ocena
- AddWarehouseBookForm.js – Dodanie “surowych danych” do tabeli magazyn_ksiazka
- AddWarehouseForm.js – Dodanie “surowych danych” do tabeli magazyn
- AddForm.module.css – wygląd formularzy

Folder advanced-reports:

Każdy plik odpowiedzialny jest za wysyłanie zapytania do serwera metodą Get i wyświetlenia otrzymanych informacji w postaci graficznej.

- BestAuthorsReport.js – Wyświetla dane na podstawie widoku
- BestBooksReport.js – Wyświetla dane na podstawie widoku
- LowStocksReport.js – Wyświetla dane na podstawie widoku
- PopularAuthorsReport.js – Wyświetla dane na podstawie widoku
- PublishersEarningsReport.js – Wyświetla dane na podstawie widoku
- PopularBooksReport.js – Wyświetla dane na podstawie widoku
- Reports.module.css – wygląd danych
- advanced-forms:

Każdy plik odpowiedzialny jest za wysyłanie danych do serwera metodą POST i wyświetlenia otrzymanej odpowiedzi w postaci graficznej pod formularzem

- AdvancedAddBook.js – Dodanie książki wraz z jej autorem i wydawcą
- AdvancedAddBooksToWarehouse.js – Dodanie książek do magazynu
- AdvancedWarehouseForm.js – Dodanie magazynu wraz z kierownikiem i adresem
- AddForm.module.css – wygląd formularzy

- Backed:

Stworzony przy użyciu technologii node.js (wersja 16.32). Dodatkowe moduły użyte do budowy aplikacji to: cors(wersja 2.8.5), date-and-time (wersja 2.1.0), express (wersja 4.7.12), nodemon (wersja 2.0.15), pg (wersja 8.7.1). Aplikacja została podzielona na moduły

- server.js – odpowiedzialny za inicjalizację bazy danych oraz przekierowywanie zapytań do odpowiednich modułów
- config.js – przechowuje konfigurację dla bazy danych i express

- response.js – moduł odpowiedzialny za przechowywanie wiadomości wysyłanej przez serwer w ramach odpowiedzi na zapytania typu POST

W folderze routes znajdują się wyspecjalizowane moduły odpowiedzialne za odbieranie zapytań, odczytywanie danych z bazy oraz wysyłanie odpowiedzi klientowi

- address.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli adres
- advanced-forms.js – odpowiedzialny za odbieranie danych z formularzy metodą POST i zapisywanie ich do bazy danych.
- author-book.js - – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli autor_ksiazka
- author.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli autor
- book.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli ksiazka
- client.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli klient
- delivery-method.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli metoda_dostawy
- examples.js – odpowiedzialny za dodanie wcześniej przygotowanych danych przykładowych do bazy, jeżeli otrzyma odpowiednie zapytanie metodą POST
- manager.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli kierownik
- order.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli zamówienie
- publisher.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli wydawca
- score.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli ocena
- views.js – odpowiedzialny za realizację zapytań typu GET na widokach w bazie danych
- warehouse-book.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli magazyn_ksiazka
- warehouse.js – odpowiedzialny za realizację zapytań POST oraz GET, które dotyczą tylko tabeli magazyn

19. Wykaz literatury:

- <https://stackoverflow.com/>
- <https://nodejs.org/en/docs/>
- <https://docs.npmjs.com/>
- <https://pl.reactjs.org/docs/getting-started.html>
- <https://node-postgres.com/>
- <https://www.postgresql.org/docs/>