

Dokumentacja projektu wykonanego w ramach zajęć Bazy Danych I

Hotel

Mateusz Cichostępski

10 stycznia 2023

Spis treści

1	Projekt koncepcji, założenia	3
1.1	Temat projektu	3
1.2	Analiza wymagań użytkownika	3
1.3	Zaprojektowanie funkcji	3
2	Projekt diagramów	4
2.1	Zdefiniowanie encji oraz ich atrybutów	4
2.2	Zaprojektowanie relacji pomiędzy encjami	5
3	Projekt logiczny	5
3.1	Projektowanie tabel, kluczy, indeksów	5
3.2	Słowniki danych	6
3.3	Zaprojektowanie operacji na danych	7
3.3.1	Wyzwalacze	7
3.3.2	Widoki	8
4	Projekt funkcjonalny	8
4.1	Interfejsy do prezentacji, edycji i obsługi danych	8
4.1.1	Użytkownik - gość	8
4.1.2	Administrator	12
4.1.3	Pracownik	19
4.2	Wizualizacja danych	20
4.3	Zdefiniowanie panelu sterowania aplikacji	20
4.4	Makropolecenia	21
5	Dokumentacja	21
5.1	Wprowadzanie danych	21
5.2	Dokumentacja użytkownika	21
5.3	Wykaz literatury	21

1 Projekt koncepcji, założenia

1.1 Temat projektu

Celem projektu jest stworzenie bazy danych do ułatwienia obsługi hotelu. Ma pomóc w wyborze pokoju przez użytkownika oraz rezerwacji przez niego wybranego terminu. Oraz pomóc w zarządzaniu rezerwacji przez obsługę hotelu, a także w zarządzaniu pracownikami hotelu.

1.2 Analiza wymagań użytkownika

W projekcie są trzy kategorie użytkowników.

- Gość
 - może zarejestrować się do serwisu
 - może zalogować się do serwisu
 - po zalogowaniu do serwisu:
 - * może przeglądać dostępne pokoje w hotelu
 - * może przeglądać szczegóły danego pokoju
 - * może wyświetlić dostępne dni do rezerwacji danego pokoju
 - * może zarezerwować dany pokój wybierając datę oraz liczbę gości
 - * może przeglądać swoje rezerwacje
 - * może edytować swoje rezerwacje zmieniając datę i/lub liczbę gości
 - * może anulować rezerwację, której data rozpoczęcia jest nie mniejsza niż jeden dzień w przód
- Administrator
 - po poprawnym zalogowaniu:
 - * może przeglądać kategorie pokoi
 - * może edytować kategorie pokoi, zmiana ceny i/lub nazwy
 - * może przeglądać pokoje w hotelu
 - * może przeglądać szczegółowe informacje danego pokoju
 - * może edytować informacje o pokoju
 - * może dodać nowy pokój
 - * może przeglądać listę pracowników
 - * może przeglądać obowiązki danego pracownika
 - * może dodać obowiązki do pracownika
 - * może dodać (zarejestrować) nowego pracownika
 - * może zakwaterować użytkownika
 - * może zaksięgować płatność
 - * może przeglądać listę najlepszych gości
- Pracownik
 - po poprawnym zalogowaniu:
 - * może przeglądać przypisane do siebie pokoje i obowiązki związane z nim

1.3 Zaprojektowanie funkcji

- Rejestracja i logowanie użytkownika
- Dodawanie rezerwacji do pokoju
- Dodawanie pokoi z określoną kategorią cenową
- Automatyczne obliczenie i dodanie niezaksięgowanej płatności rezerwacji

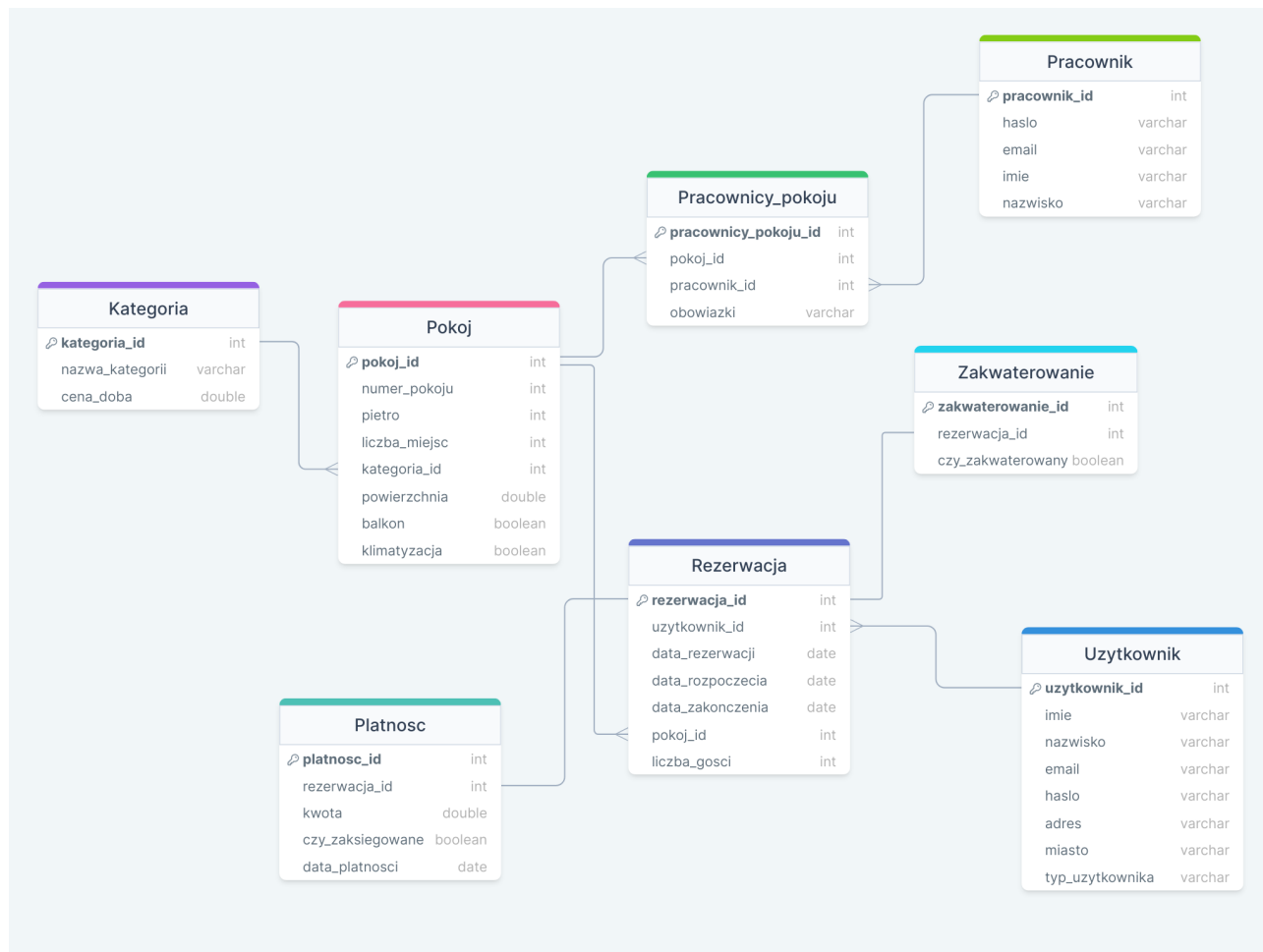
- Automatyczne dodanie rekordu z zakwaterowaniem do rezerwacji
- Edycja pokoju
- Weryfikacja poprawności wprowadzanych danych
- Sprawdzanie dostępności pokoju
- Rekalkulacja kwoty po edycji rezerwacji
- Weryfikacja anulowania rezerwacji
- Anulowanie rezerwacji i informacji z nim związanych

2 Projekt diagramów

2.1 Zdefiniowanie encji oraz ich atrybutów

- **Uzytkownik** - uzytkownik_id, imie, nazwisko, email, haslo, adres, miasto, typ_uzytkownika
- **Rezerwacja** - rezerwacja_id, uzytkownik_id, data_rezerwacji, data_rozpoczecia, data_zakonczenia, pokoj_id, liczba_gosci
- **Pokoj** - pokoj_id, numer_pokoju, pietro, liczba_miejsc, kategoria_id, powierzchnia, balkon, klimatyzacja
- **Pracownik** - pracownik_id, imie, nazwisko, email, haslo
- **Kategoria** - kategoria_id, nazwa_kategorii, cena_doba
- **Platnosc** - platnosc_id, rezerwacja_id, kwota, czy_zaksiegowane, data_platnosci
- **Zakwaterowanie** - zakwaterowanie_id, rezerwacja_id, czy_zakwaterowany
- **Pracownicy_pokoju** - pracownicy_pokoju_id, pokoj_id, pracownik_id, obowiazki

2.2 Zaprojektowanie relacji pomiędzy encjami



Rysunek 1: Diagram relacji pomiędzy encjami

3 Projekt logiczny

3.1 Projektowanie tabel, kluczy, indeksów

- Uzytkownik

- `uzytkownik_id` *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje uzytkownika

- Rezerwacja

- `rezerwacja_id` *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje rezerwacje
- `uzytkownik_id` *INTEGER FOREIGN KEY* - identyfikuje uzytkownika rezerwacji, relacja *N:1*
- `pokoj_id` *INTEGER FOREIGN KEY* - identyfikuje pokój rezerwacji, relacja *N:1*

- Pokoj

- `pokoj_id` *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje pokój
- `kategoria_id` *INTEGER SERIAL FOREIGN KEY* - identyfikuje kategorie pokoju, relacja *N:1*

- Pracownik

- `pracownik_id` *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje pracownika

- Kategoria

- `kategoria_id` *INTEGER PRIMARY KEY* - jednoznacznie identyfikuje kategorie pokoju

- **Platnosc**

- **platnosc_id** *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje płatność
- **rezerwacja_id** *INTEGER FOREIGN KEY* - identyfikuje płatność rezerwacji, relacja 1:1

- **Zakwaterowanie**

- **zakwaterowanie_id** *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje zakwaterowanie
- **rezerwacja_id** *INTEGER FOREIGN KEY* - identyfikuje zakwaterowanie rezerwacji, relacja 1:1

- **Pracownicy_pokoju**

- **pracownicy_pokoju_id** *INTEGER SERIAL PRIMARY KEY* - jednoznacznie identyfikuje przypisanie pracownika do pokoju
- **pokoj_id** *INTEGER FOREIGN KEY* - identyfikuje przypisanie pokoju, relacja N:1
- **pracownik_id** *INTEGER FOREIGN KEY* - identyfikuje przypisanie pracownika, relacja N:1

3.2 Słowniki danych

- **Uzytkownik**

- **uzytkownik_id** *INTEGER SERIAL NOT NULL*
- **imie** *VARCHAR NOT NULL*
- **nazwisko** *VARCHAR NOT NULL*
- **email** *VARCHAR NOT NULL*, unikalny adres email, na potrzeby testowania brak sprawdzania poprawności formy (możliwość pominięcia znaku @)
- **haslo** *VARCHAR NOT NULL*
- **adres** *VARCHAR NOT NULL*
- **miasto** *VARCHAR NOT NULL*
- **typ_uzytkownika** *VARCHAR NOT NULL*

- **Rezerwacja**

- **rezerwacja_id** *INTEGER SERIAL NOT NULL*
- **uzytkownik_id** *INTEGER NOT NULL*
- **data_rezerwacji** *DATE NOT NULL*
- **data_roz poczeczia** *DATE NOT NULL*, mniejsza od daty zakończenia i większa bądź równa od dzisiejszej daty
- **data_zakonczenia** *DATE NOT NULL*
- **pokoj_id** *INTEGER NOT NULL*
- **liczba_gosci** *INTEGER NOT NULL*

- **Pokoj**

- **pokoj_id** *INTEGER SERIAL NOT NULL*
- **numer_pokoju** *INTEGER NOT NULL*, unikalny numer pokoju
- **pietro** *INTEGER NOT NULL*
- **liczba_miejsc** *INTEGER NOT NULL*
- **kategoria_id** *INTEGER NOT NULL*
- **powierzchnia** *DOUBLE NOT NULL > 0*
- **balkon** *BOOLEAN NOT NULL*
- **klimatyzacja** *BOOLEAN NOT NULL*

- **Pracownik**

- pracownik_id *INTEGER SERIAL NOT NULL*
- imie *VARCHAR NOT NULL*
- nazwisko *VARCHAR NOT NULL*
- email *VARCHAR NOT NULL*, unikalny email pracownika, na potrzeby testowania brak sprawdzania poprawności formy (możliwość pominięcia znaku @)
- haslo *VARCHAR NOT NULL*

- **Kategoria**

- kategoria_id *INTEGER SERIAL NOT NULL*
- nazwa_kategorii *VARCHAR NOT NULL*, unikalna nazwa kategorii
- cena_doba *DOUBLE PRECISION NOT NULL*

- **Platnosc**

- platnosc_id *INTEGER SERIAL NOT NULL*
- rezerwacja_id *INTEGER NOT NULL*
- kwota *DOUBLE PRECISION NOT NULL*
- czy_zaksiegowane *BOOLEAN NOT NULL*
- data_platnosci *DATE*

- **Zakwaterowanie**

- zakwaterowanie_id *INTEGER SERIAL NOT NULL*
- rezerwacja_id *INTEGER NOT NULL*
- czy_zakwaterowany *BOOLEAN NOT NULL*

- **Pracownicy_pokoju**

- pracownicy_pokoju_id *INTEGER SERIAL NOT NULL*
- pokoj_id *INTEGER NOT NULL*
- pracownik_id *INTEGER NOT NULL*
- obowiazki *VARCHAR NULL*

3.3 Zaprojektowanie operacji na danych

Polecenia budujące oraz realizujące projekt znajdują się w folderze *SQL/*

3.3.1 Wyzwalacze

- **dodajPlatnoscDoRezerwacji** - wyzwała się po wprowadzeniu danych do tabeli *Rezerwacja*. Oblicza kwotę do zapłaty (liczba dni pomnożona przez cenę kategorii), dodaje rekord do tabeli *Platnosc* z daną rezerwacją, kwotą do zapłaty, ustawioną wartością zaksiegowania na *FALSE* oraz datą płatności na *NULL*.
- **dodajZakwaterowanieDoRezerwacji** - wyzwała się po wprowadzeniu danych do tabeli *Rezerwacja*. Dodaje rekord do tabeli *Zakwaterowanie* z daną rezerwacją oraz wartością *czy_zakwaterowany* na *FALSE*.
- **edytujRezerwacje** - wyzwała się przed edycją danych w tabeli *Rezerwacja*. Sprawdza czy edycja nie zachodzi na starej lub aktualnej rezerwacji. Oblicza zmienioną liczbę dni rezerwacji, aktualizuje wartość kwoty w rekordzie w tabeli *Platnosc*.
- **sprawdzNowyPokoj** - wyzwała się przed wprowadzeniem lub edycją danych do tabeli *Pokoj*. Sprawdza poprawność danych (powierzchnia większa od 0, unikalny numer pokoju).

- **sprawdzNowaRezerwacje** - wyzwała się przed dodaniem lub edycją danych na tabeli *Rezerwacja*. Sprawdza poprawność podanych danych (poprawność początku i końca, odpowiednią liczbę gości dla danego pokoju, dostępność pokoju w podanym okresie czasu).
- **sprawdzNowegoUzytkownika** - wyzwała się przed dodaniem lub edycją w tabeli *Uzytkownik*. Sprawdza unikalność adresu email.
- **sprawdzNowegoPracownika** - wyzwała się przed dodaniem lub edycją w tabeli *Pracownik*. Sprawdza unikalność adresu email.
- **sprawdzDodaniePracownikaDoPokoju** - wyzwała się przed dodaniem lub edycją danych w tabeli *Pracownicy_pokoju*. Sprawdza czy dany pracownik nie jest już przypisany do pokoju.
- **sprawdzUsuniecieRezerwacji** - wyzwała się przed usunięciem danych z tabeli *Rezerwacje*. Blokuje możliwość usunięcia jeżeli rezerwacja jest z przeszłości, jest aktywna lub jeżeli rezerwacja rozpoczyna się za mniej niż 24h. W przeciwnym wypadku usuwa także odpowiednie rekordy z tabel *Platnosc* oraz *Zakwaterowanie*.
- **sprawdzUsunieciePokoju** - wyzwała się przed usunięciem danych z tabeli *Pokoj*. Blokuje usunięcie pokoju jeżeli posiada aktywne lub przyszłe rezerwacje. W przeciwnym wypadku usuwa także rekordy z tabel *Pracownicy_pokoju* oraz *Rezerwacje*.
- **sprawdzUsunieciePracownika** - wyzwała się przed usunięciem danych z tabeli *Pracownik*. Usuwa odpowiednie rekordy z tabeli *Pracownicy_pokoju*.

3.3.2 Widoki

- **ranking_uzytkownikow** - zwraca listę dziesięciu najlepszych użytkowników, takich z największą ilością rezerwacji. Brane są pod uwagę tylko rezerwacje już odbyte.
- **rezerwacje_bez_zakwaterowania** - zwraca listę przyszłych rezerwacji, których goście nie zostali jeszcze zakwaterowani.
- **platnosci_bez_zaksiegowania** - zwraca listę rezerwacji, których płatność nie została jeszcze zaksięgowana.
- **lista_pokoi** - zwraca listę pokoi, oraz dodatkową kolumnę z informacją czy pokój posiada przyszłe rezerwacje. Widok używany do sprawdzenia czy dany pokój można usunąć.

4 Projekt funkcjonalny

Aplikacja została napisana z wykorzystaniem biblioteki *React* oraz frameworku *Next.JS* jako strona internetowa.

Strona serwerowa natomiast została napisana z użyciem środowiska *Node.JS*.

Zależnie od typu użytkownika są różne poziomy dostępu do podstron.

4.1 Interfejsy do prezentacji, edycji i obsługi danych

4.1.1 Użytkownik - gość

- **Niealogowany użytkownik** <http://127.0.0.1:3000/>
Niealogowany użytkownik po wejściu na stronę widzi stronę główną, z której może przejść do formularzy logowania lub rejestracji.



Rysunek 2: Strona główna

- **Rejestracja** *<http://127.0.0.1:3000/userRegister>*

Po wejściu na podstronę rejestracji użytkownikowi ukazuje się formularz do rejestracji. Wszystkie pola są wymagane, dokładna weryfikacja danych zachodzi po stronie serwera i bazy danych.

The image shows a web browser window with a white background. At the top left, there is a horizontal bar containing two blue buttons with white text: 'Zaloguj' and 'Zarejestruj'. In the center of the page, there is a registration form. The form has the following fields: 'Imię' with the value 'Jan', 'Nazwisko' with the value 'Kowalski', 'Email' with the value 'email@zzz.xx', 'Password' with the value 'xxx', 'Adres' with the value 'Ul. Długa 13', and 'Miasto' with the value 'Ul. Długa 13'. At the bottom right of the form, there is a blue button with white text: 'Zarejestruj'.

Rysunek 3: Formularz rejestracyjny

- **Logowanie** *<http://127.0.0.1:3000/userLogin>*

Po wejściu na podstronę logowania użytkownikowi ukazuje się formularz do logowania. Wszystkie pola są wymagane, dokładna weryfikacja danych zachodzi po stronie serwera i bazy danych.

The screenshot shows a login interface. At the top left, there are two buttons: 'Zaloguj' (Login) and 'Zarejestruj' (Register). In the center, there is a form with two input fields: 'Email użytkownika' (User email) containing 'email@zzz.xx' and 'Password' containing 'xxxx'. Below these fields is a blue 'Zaloguj' button. At the bottom center, there is a blue button labeled 'Zaloguj jako pracownik' (Login as employee).

Rysunek 4: Formularz logowania

- **Panel użytkownika** <http://127.0.0.1:3000/userDashboard>

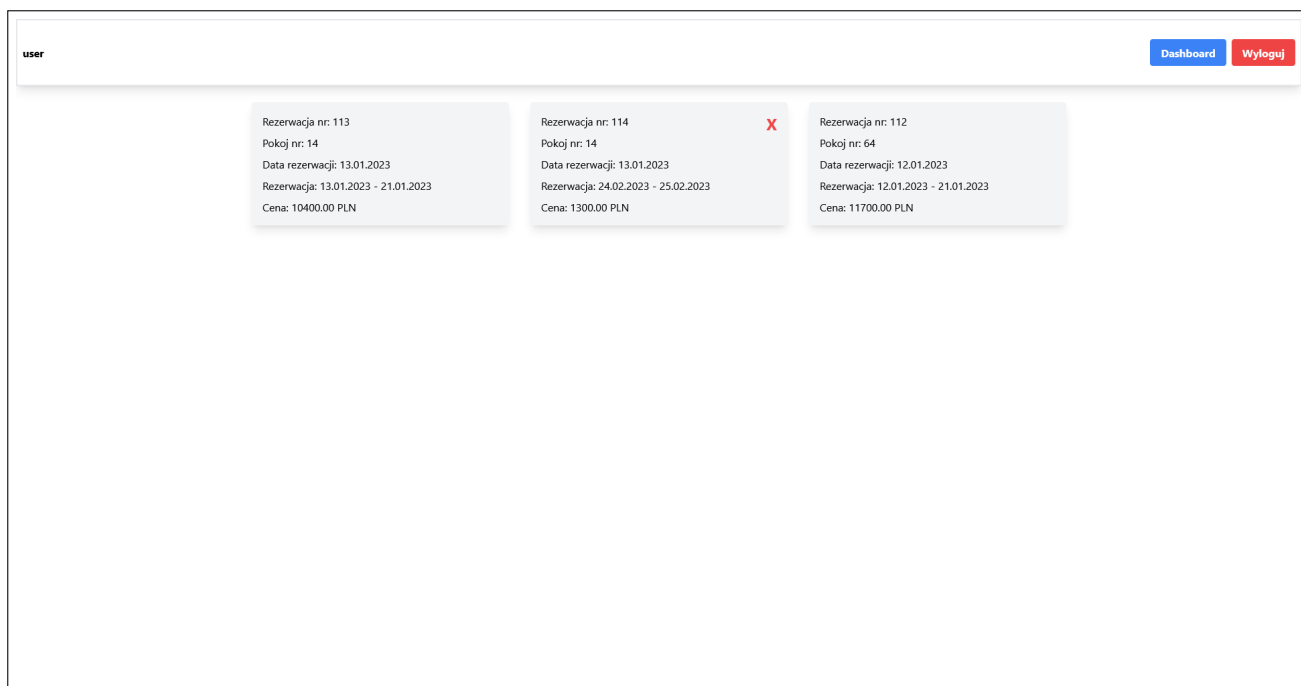
Po zalogowaniu się użytkownik zostaje przeniesiony do panelu użytkownika, z którego może wejść na stronę ze swoimi rezerwacjami lub stronę z listą dostępnych pokoi. Na górze strony po lewej stronie widnieje email aktualnego użytkownika, po prawej przyciski powrotu do panelu głównego oraz wylogowania.

The screenshot shows a user dashboard. At the top left, the word 'user' is displayed. At the top right, there are two buttons: 'Dashboard' and 'Wyloguj' (Logout). In the center, there are two blue buttons stacked vertically: 'Twoje rezerwacje' (Your reservations) and 'Przeglądaj pokoje' (Browse rooms).

Rysunek 5: Panel użytkownika

- **Lista rezerwacji** <http://127.0.0.1:3000/userDashboard/bookings>

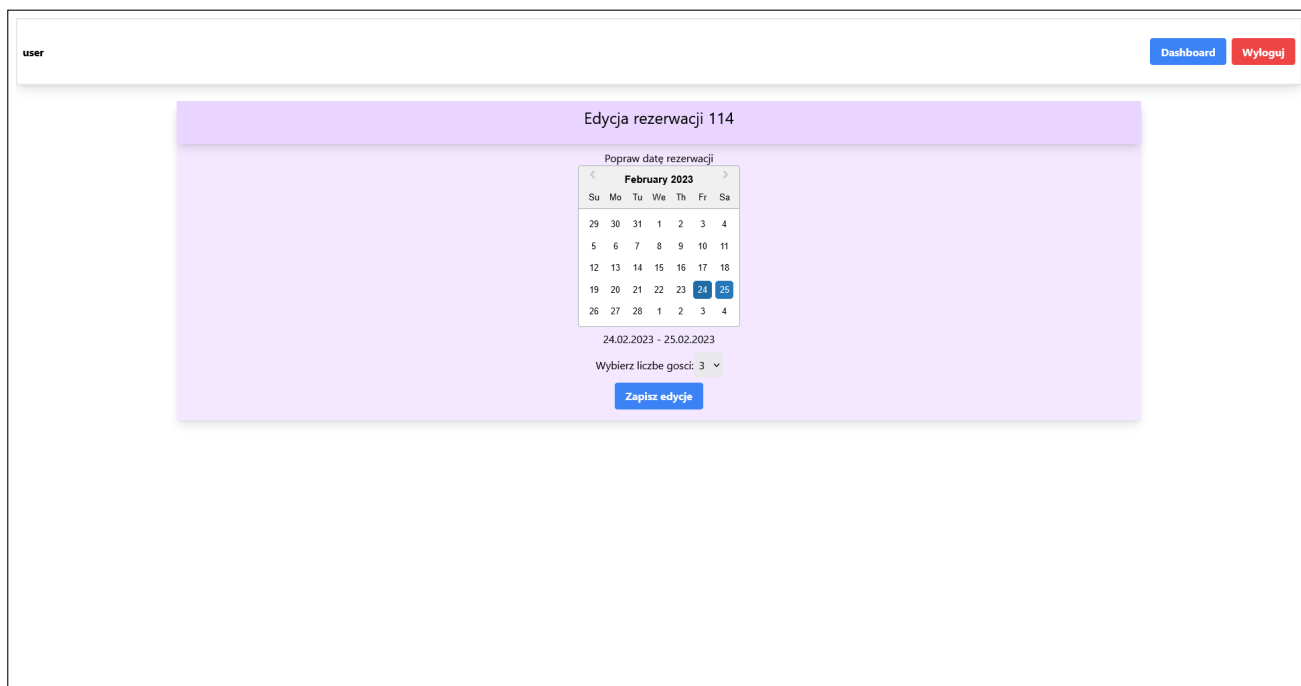
Użytkownik ma wgląd do swoich rezerwacji. Przy rezerwacjach, których data rozpoczęcia jest większa niż jeden dzień w przód widnieje czerwony przycisk do anulowania rezerwacji. Przyszłe rezerwacje są także dostępne do edycji, kliknięcie ich powoduje przejście na stronę edycji.



Rysunek 6: Lista rezerwacji użytkownika

- **Lista edycji rezerwacji** <http://127.0.0.1:3000/userDashboard/bookings/<id>>

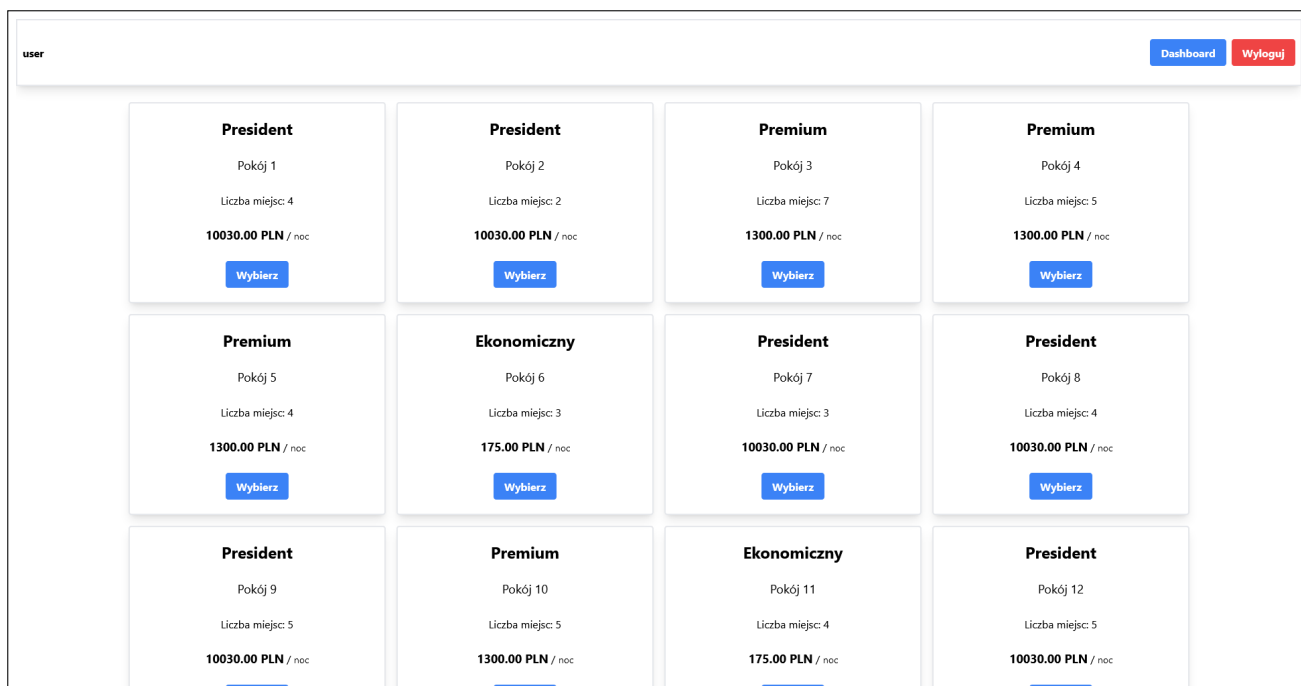
Użytkownik na stronie edycji może edytować swoją rezerwację. Po poprawnym zapisie edycji pod przyciskiem wyświetli się odpowiedni komunikat.



Rysunek 7: Edycja rezerwacji użytkownika

- **Lista dostępnych pokoi** <http://127.0.0.1:3000/userDashboard/rooms>

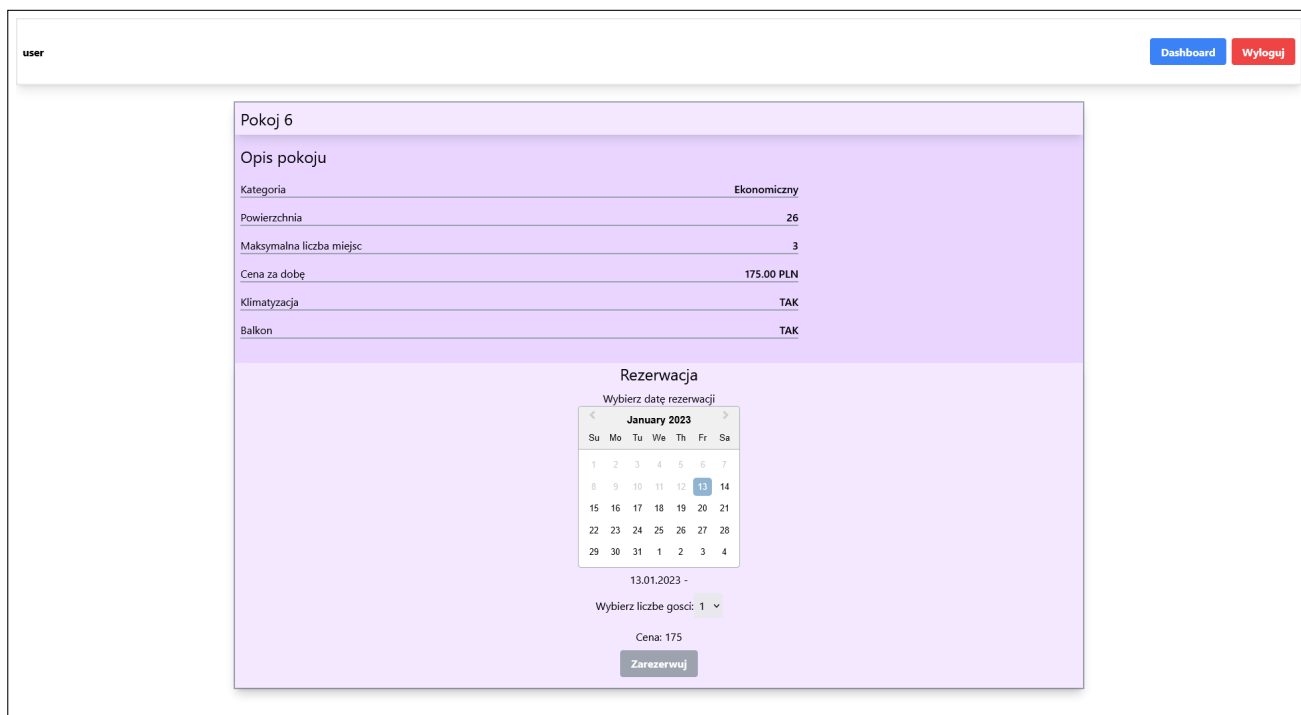
Użytkownik na stronie z listą pokoi może wybrać dogodny dla siebie pokój klikając w przycisk *Wybierz*. Pokoje są posortowane przez numer pokoju.



Rysunek 8: Lista pokoi

- **Rezerwacja wybranego pokoju** <http://127.0.0.1:3000/userDashboard/rooms/<id>>

Użytkownik po wybraniu pokoju ma wgląd do jego szczegółów. Poniżej jest kalendarz, na którym użytkownik może wybrać dogodny dla siebie okres dni rezerwacji. Niedostępne dni są w szarym kolorze. Użytkownik może wybrać liczbę gości. Po zatwierdzonej rezerwacji pojawi się napis *Dodano rezerwację*, która będzie także widoczna na podstronie z rezerwacjami.



Rysunek 9: Lista pokoi

4.1.2 Administrator

- **Niezałogowany administrator** <http://127.0.0.1:3000/userLogin>

Niezałogowany administrator, tak jak użytkownik, loguje się na stronie z formularzem logowania.

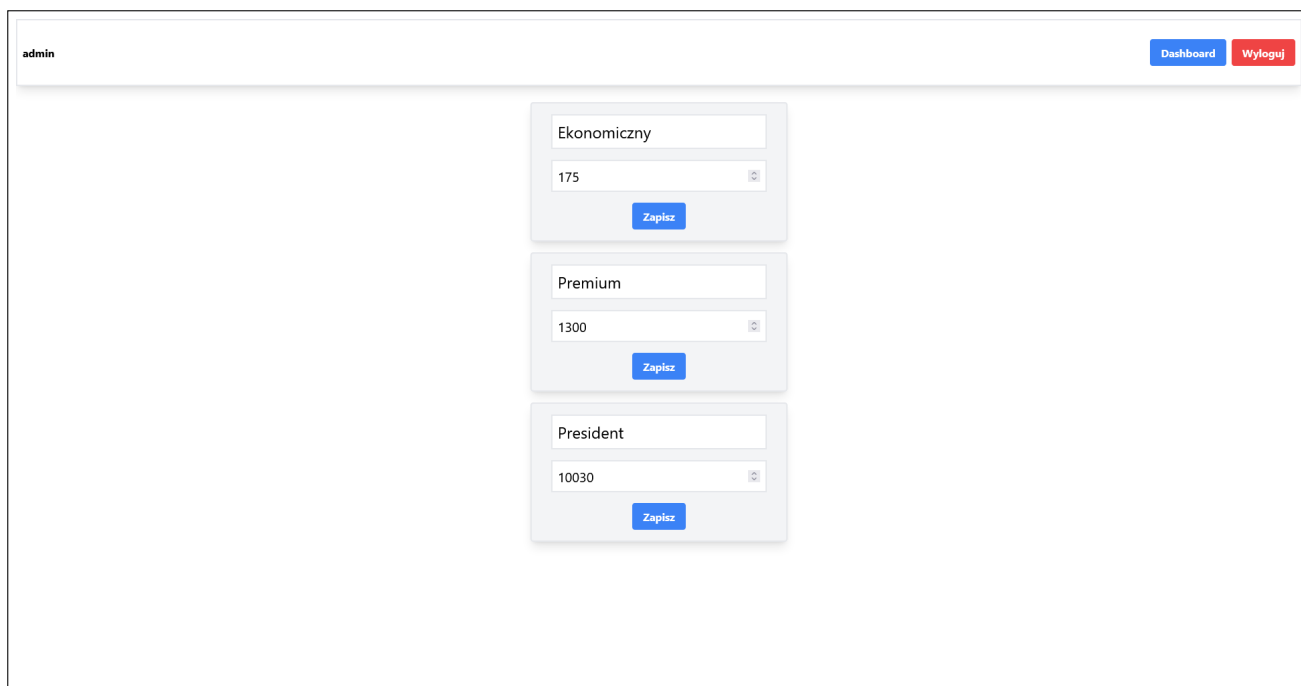
Rysunek 10: Formularz logowania administratora

- **Panel administratora** <http://127.0.0.1:3000/adminDashboard>

Po zalogowaniu jako administrator, zostanie otworzona strona z panelem administratora. Stąd może przejść do wszystkich podstron administracyjnych.

Rysunek 11: Panel administratora

- **Edycja kategorii** <http://127.0.0.1:3000/adminDashboard/categories>
Administrator może zmienić nazwy lub/i ceny za dobę danej kategorii.



admin Dashboard Wyloguj

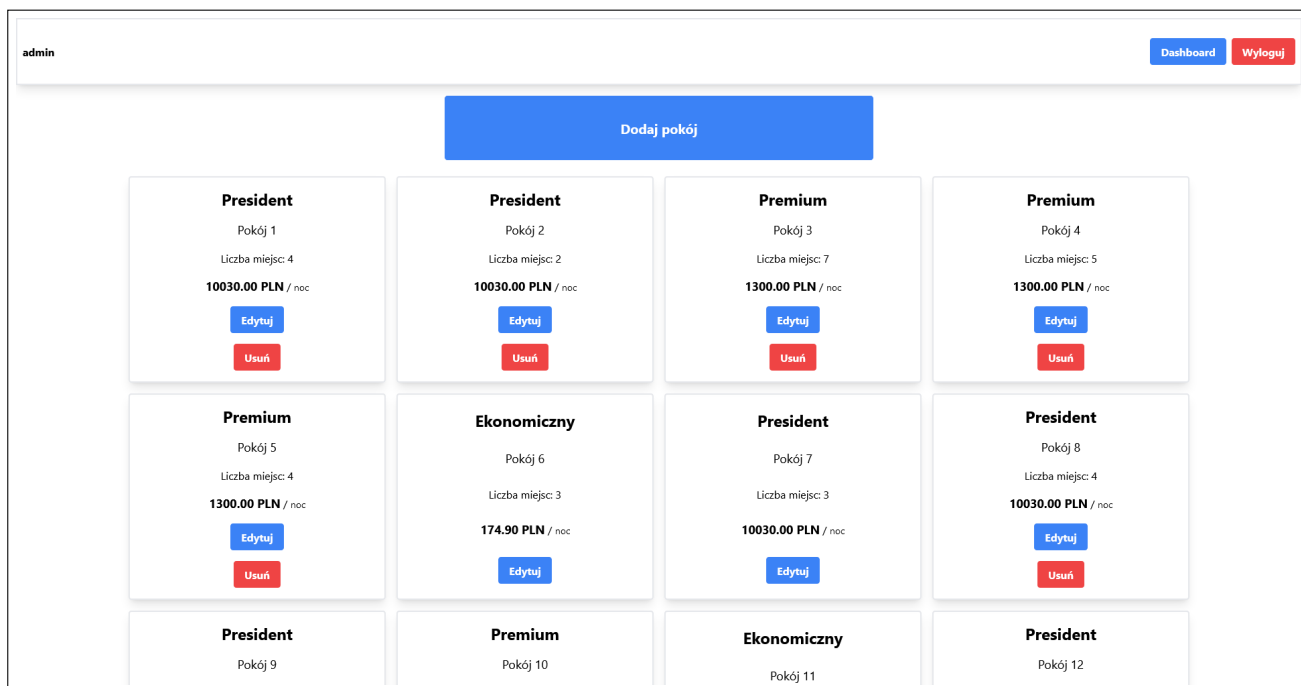
Ekonomiczny
175
Zapisz

Premium
1300
Zapisz

President
10030
Zapisz

Rysunek 12: Edycja kategorii

- **Przegląd pokoi** <http://127.0.0.1:3000/adminDashboard/rooms>
Administrator ma wgląd do wszystkich pokoi dostępnych w hotelu. Może edytować wybrany pokój, usunąć pokój nieposiadający przyszłych rezerwacji, dodać nowy pokój do hotelu.



admin Dashboard Wyloguj

Dodaj pokój

President Pokój 1 Liczba miejsc: 4 10030.00 PLN / noc Edytuj Usuń	President Pokój 2 Liczba miejsc: 2 10030.00 PLN / noc Edytuj Usuń	Premium Pokój 3 Liczba miejsc: 7 1300.00 PLN / noc Edytuj Usuń	Premium Pokój 4 Liczba miejsc: 5 1300.00 PLN / noc Edytuj Usuń
Premium Pokój 5 Liczba miejsc: 4 1300.00 PLN / noc Edytuj Usuń	Ekonomiczny Pokój 6 Liczba miejsc: 3 174.90 PLN / noc Edytuj	President Pokój 7 Liczba miejsc: 3 10030.00 PLN / noc Edytuj	President Pokój 8 Liczba miejsc: 4 10030.00 PLN / noc Edytuj Usuń
President Pokój 9	Premium Pokój 10	Ekonomiczny Pokój 11	President Pokój 12

Rysunek 13: Edycja kategorii

- **Edycja pokoju** <http://127.0.0.1:3000/adminDashboard/rooms/<id>>
Administrator może zmienić wszystkie szczegóły dotyczące pokoju. Po poprawnym zapisie pokaże się odpowiednia wiadomość.

admin
Dashboard
Wyloguj

Pokoj 11

Edytuj pokój

Numer pokoju

11

Kategoria

Ekonomiczny

Powierzchnia

38

Maksymalna liczba miejsc

4

Pietro

10

Cena za dobe

174.9

Klimatyzacja

TAK

Balkon

NIE

Zapisz

Rysunek 14: Edycja pokoju

- **Dodawanie nowego pokoju** <http://127.0.0.1:3000/adminDashboard/rooms/add>

Administrator może dodać nowy pokój z wybranymi atrybutami. Numer pokoju musi być unikalny. Po dodaniu pokaże się odpowiednia wiadomość, a nowy pokój będzie widoczny w liście pokoi.

admin
Dashboard
Wyloguj

Dodaj pokój

Numer pokoju

0

Kategoria

Ekonomiczny

Powierzchnia

0

Maksymalna liczba miejsc

1

Pietro

0

Cena za dobe

174.9

Klimatyzacja

NIE

Balkon

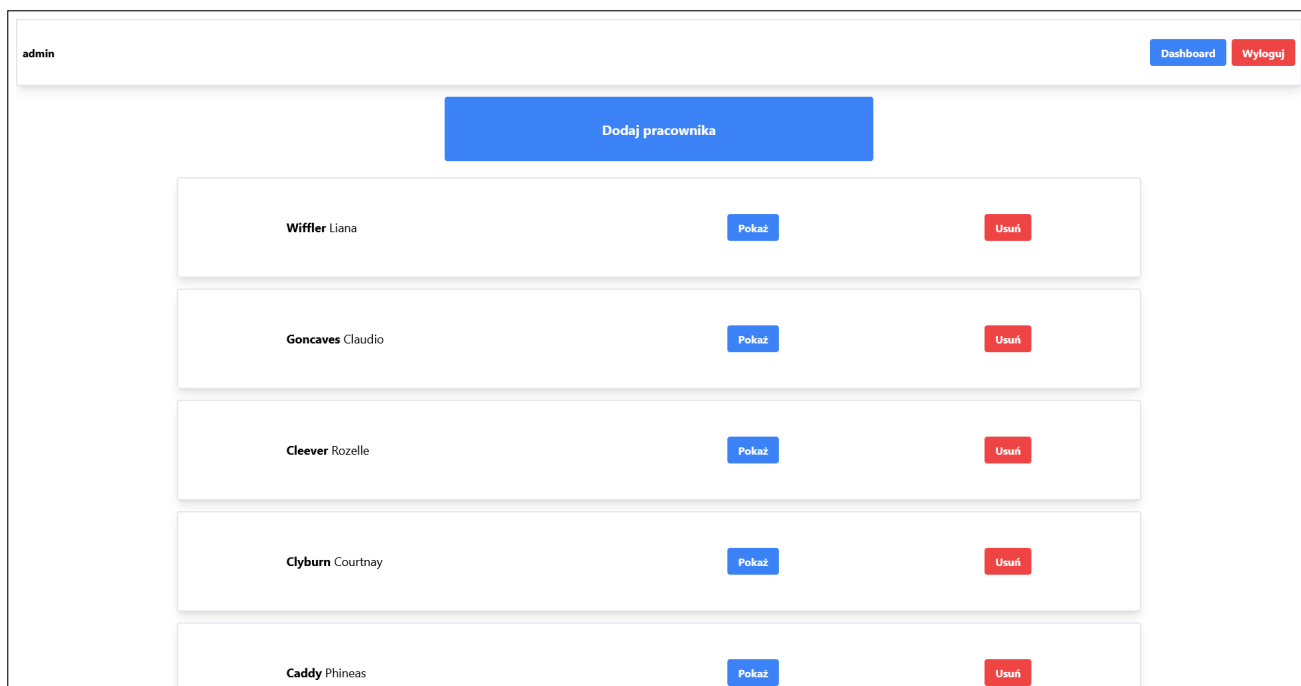
NIE

Dodaj

Rysunek 15: Dodawanie pokoju

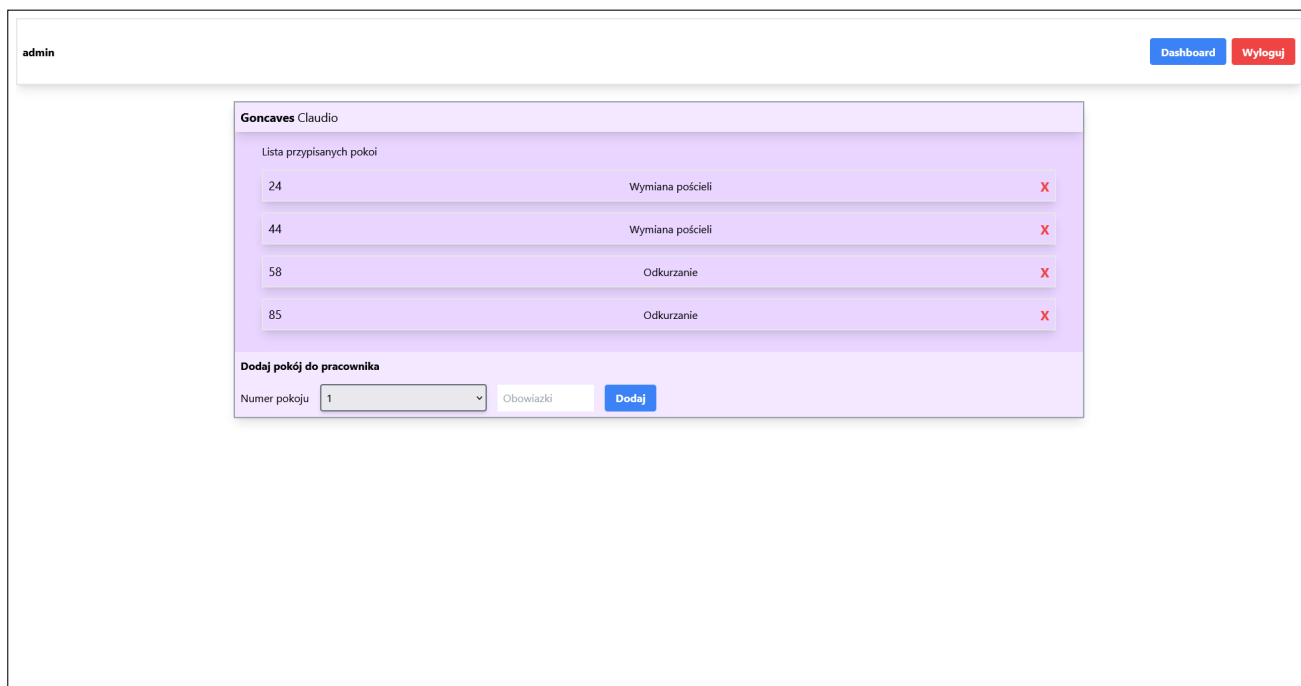
- **Przegląd pracowników** <http://127.0.0.1:3000/adminDashboard/employees>

Administrator ma wgląd do wszystkich pracowników hotelu. Może wyświetlić szczegóły wybranego pracownika, usunąć pracownika i dodać nowego pracownika.



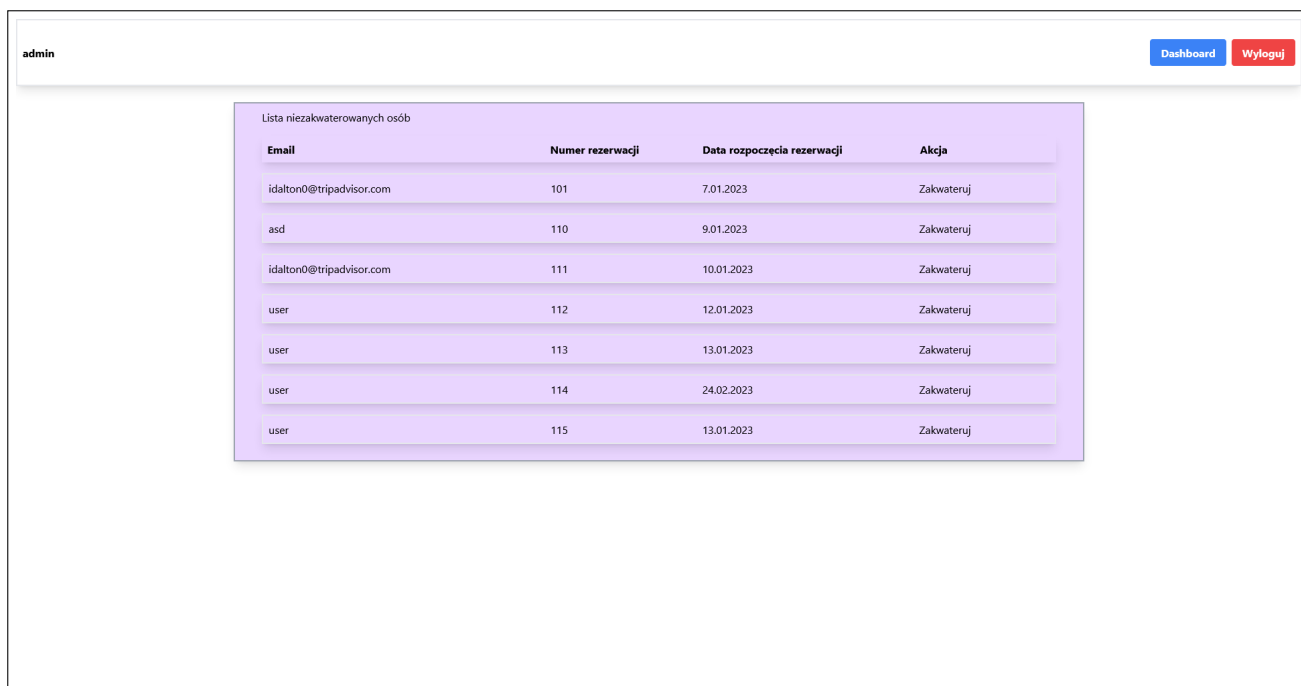
Rysunek 16: Lista pracowników

- **Szczegóły pracownika** <http://127.0.0.1:3000/adminDashboard/employees/<id>>
Administrator ma wgląd do listy przypisanych pokoi i obowiązków danego pracownika. Może dodać nowy pokój i obowiązek oraz usunąć wybrane przypisanie pokoju.



Rysunek 17: Szczegóły pracownika

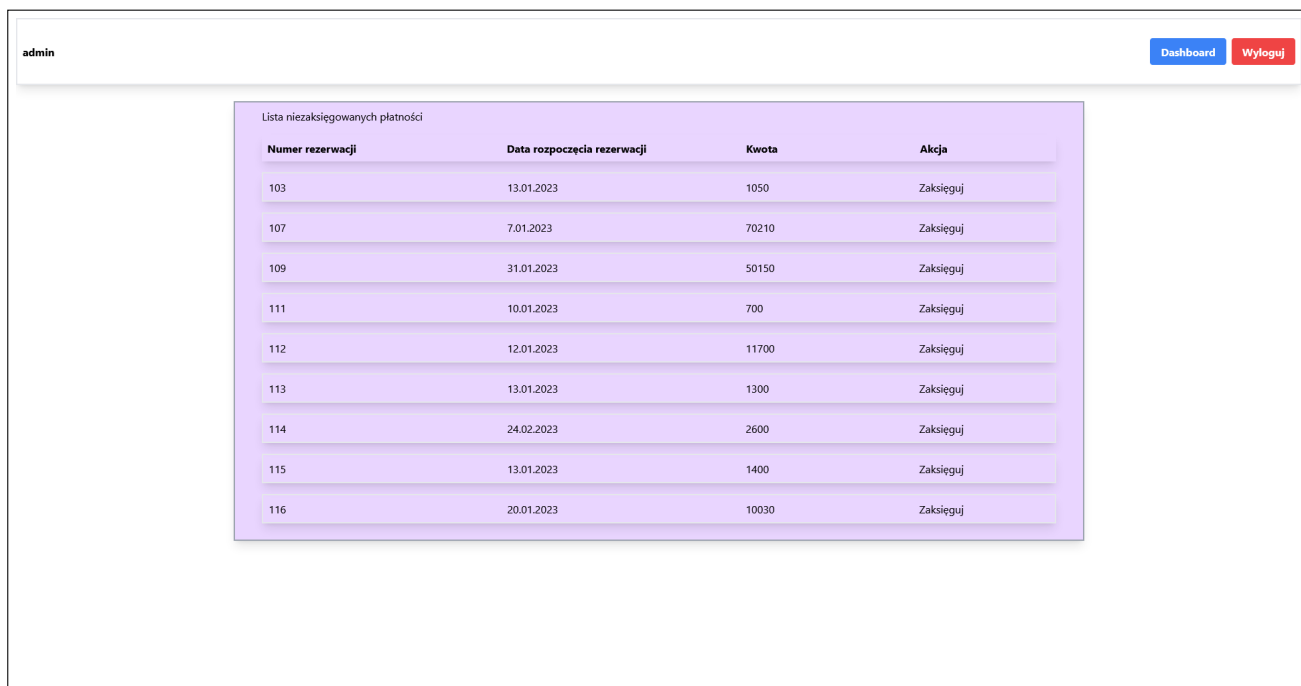
- **Dodawanie zakwaterowania** <http://127.0.0.1:3000/adminDashboard/addAccommodation>
Administrator ma wgląd do listy przyszłych rezerwacji, których goście nie zostali jeszcze zakwaterowani. Może ich zakwaterować klikając przycisk *Zakwateruj*.



Rysunek 18: Dodawanie zakwaterowania

- **Zaksięgowanie płatności** <http://127.0.0.1:3000/adminDashboard/payments>

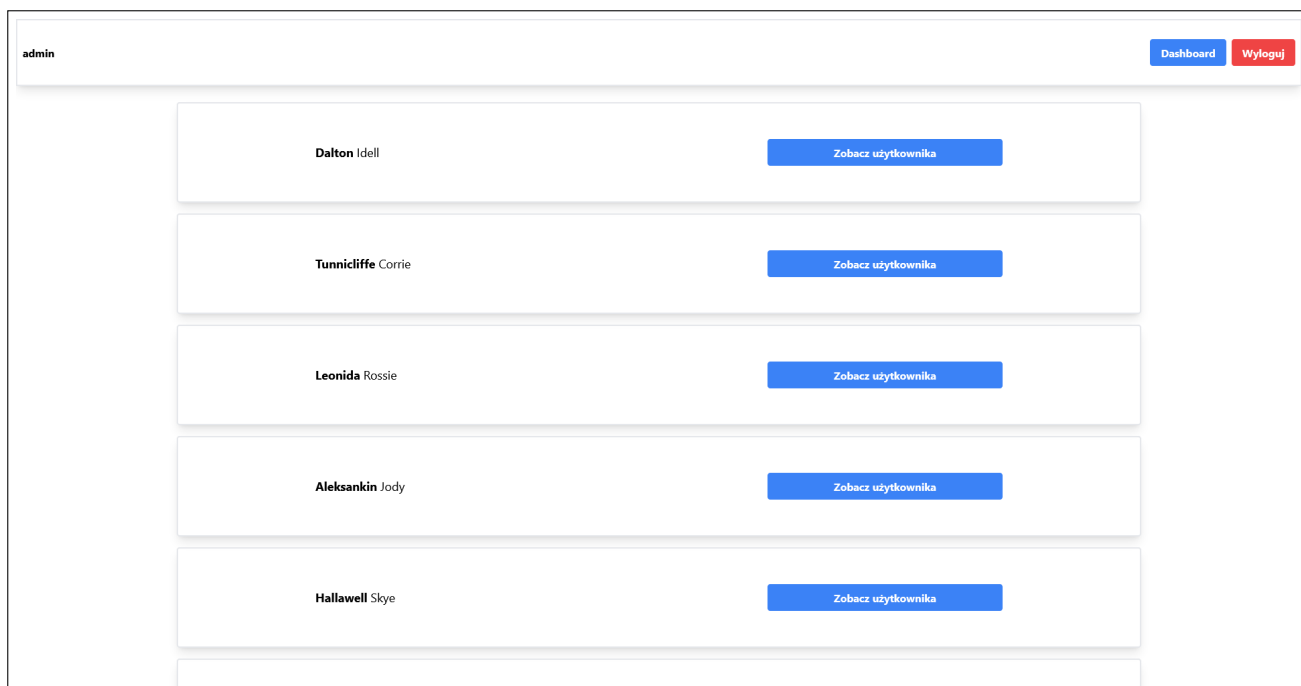
Administrator ma wgląd do listy niezaksięgowanych płatności. Może je zaksięgować klikając przycisk *Zaksięguj*.



Rysunek 19: Zaksięgowanie płatności

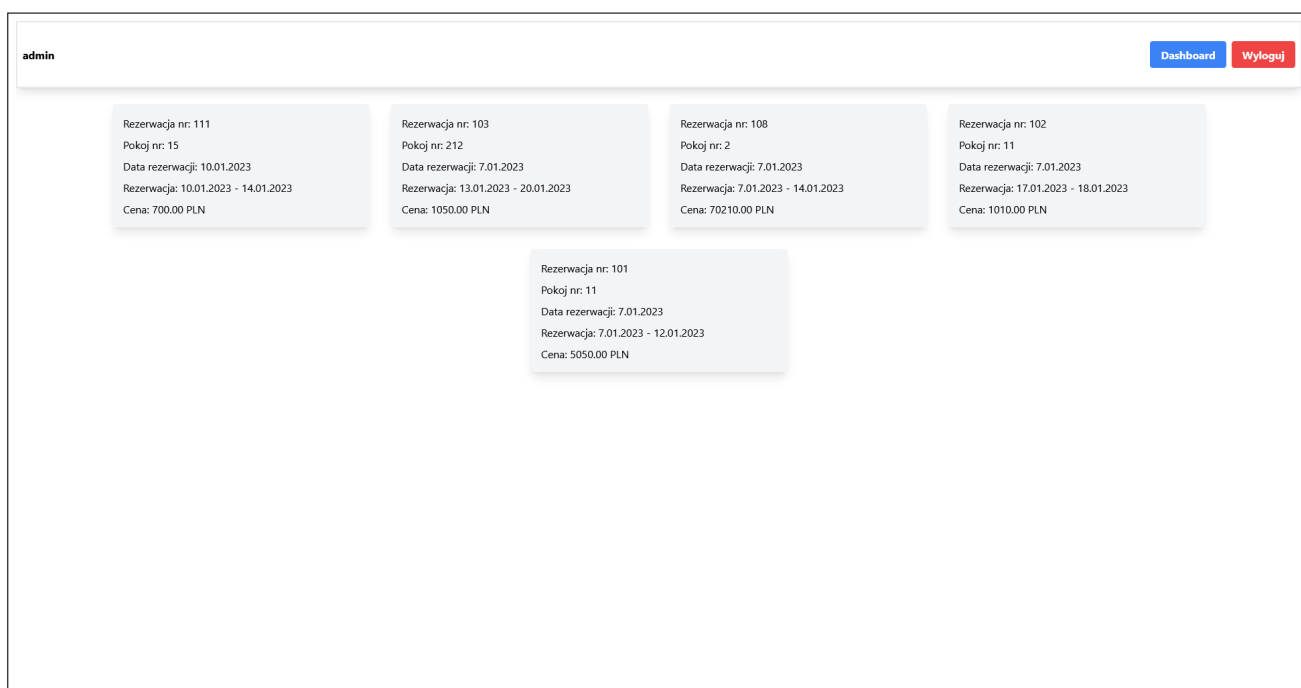
- **Przegląd użytkowników** <http://127.0.0.1:3000/adminDashboard/users>

Administrator ma wgląd do listy wszystkich zarejestrowanych gości hotelu. Po kliknięciu przycisku *Zobacz użytkownika* ma wgląd do jego rezerwacji.



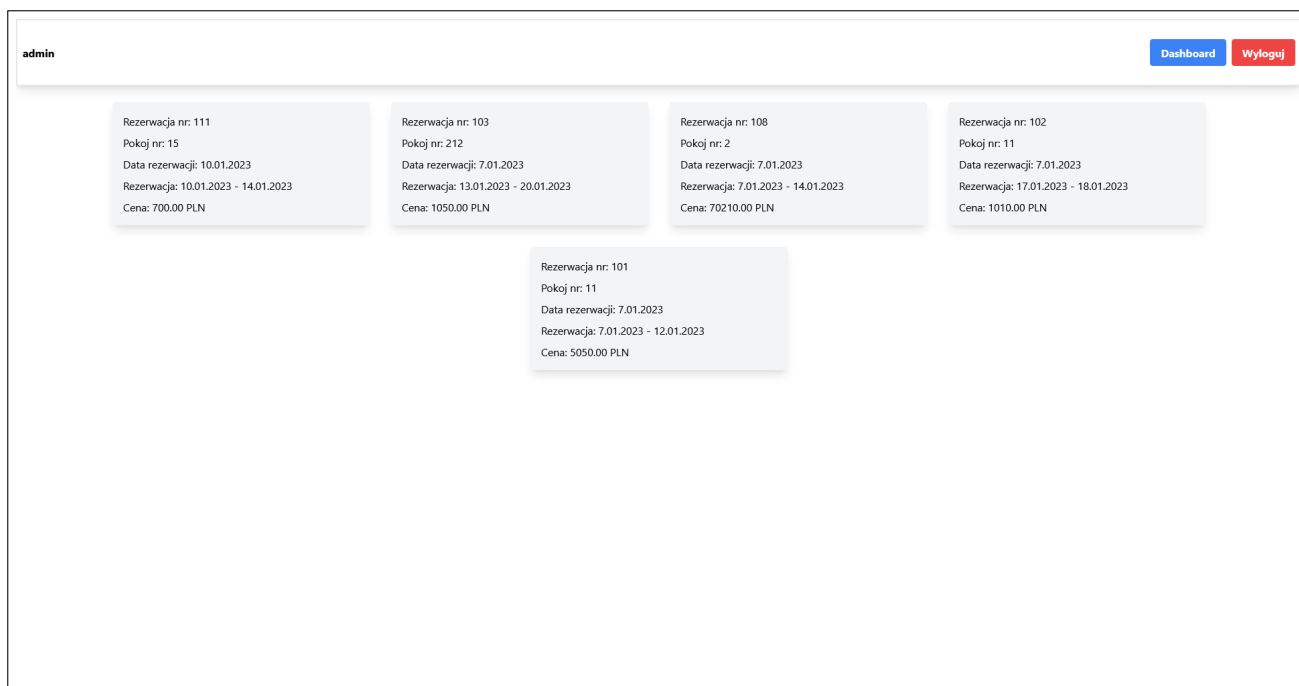
Rysunek 20: Lista użytkowników

- **Rezerwacje użytkownika** <http://127.0.0.1:3000/adminDashboard/users/<id>>
Administrator ma dostęp do przeglądu rezerwacji wybranego użytkownika.



Rysunek 21: Lista rezerwacji użytkownika

- **Rezerwacje użytkownika** <http://127.0.0.1:3000/adminDashboard/users/<id>>
Administrator ma dostęp do przeglądu rezerwacji wybranego użytkownika.



Rysunek 22: Lista rezerwacji użytkownika

- **Lista najlepszych gości** <http://127.0.0.1:3000/adminDashboard/bestUsers>
Administrator ma dostęp do przeglądu listy najlepszych gości hotelu.

The screenshot shows an admin dashboard with a header bar containing the username 'admin' and two buttons: 'Dashboard' (blue) and 'Wyloguj' (red). Below the header, there is a table listing the best guests based on the number of reservations.

Email	Ilość rezerwacji
idalton0@tripadvisor.com	5
user	3
mchimenti6@edublogs.org	1
stourne26@jiggy.com	1
oheislere@bandcamp.com	1
dniset11@mit.edu	1
ctunncliffe1@yandex.ru	1
mhusthwaite1g@go.com	1
brides9@cdbaby.com	1
bsleford19@homestead.com	1

Rysunek 23: Lista najlepszych gości

4.1.3 Pracownik

- **Niezałogowany pracownik** <http://127.0.0.1:3000/employeeLogin>
Niezałogowany pracownik może się zalogować do aplikacji przez specjalny formularz logowania dla pracowników. Na stronie dla logowania użytkowników (Rysunek 4) wymagane jest kliknięcie przycisku *Zaloguj jako pracownik*.

Rysunek 24: Formularz logowania pracownika

- **Panel pracownika** <http://127.0.0.1:3000/employeeDashboard>

Po zalogowaniu pracownik ma wgląd do swoich przypisać pokoi i obowiązków z nimi związanych.

Lista twoich przypisanych pokoi	
1	-
5	-
7	-
15	Sprzątanie
39	Odkurzanie

Rysunek 25: Panel pracownika

4.2 Wizualizacja danych

Aplikacja wyświetla dane pobrane z bazy danych głównie za pomocą struktury języka HTML.

4.3 Zdefiniowanie panelu sterowania aplikacji

Sterowanie aplikacją zapewniają elementy języka HTML, tj. przyciski, listy rozwijane, formularze. Wykorzystano zewnętrzny komponent z kalendarzem *react-datepicker* do ułatwienia wybierania dat.

4.4 Makropolecenia

Do ułatwienia obsługi wykorzystano listy rozwijane wyświetlające odpowiednie informacje i ułatwiające wybór opcji. Formularze posiadają zabezpieczenia przed wprowadzeniem błędnych danych do bazy.

5 Dokumentacja

5.1 Wprowadzanie danych

Zostały wprowadzone wygenerowane automatycznie dane na potrzeby testowania i udoskonalania projektu. Dane znajdują się w pliku *seed.sql*.

W tabeli *Kategoria* liczba rekordów jest stała. Możliwa jest tylko ich edycja nazwy i ceny.

Rekordy do tabel *Pracownicy_pokoju* są wprowadzane automatycznie i nie są dostępne do ich modyfikacji. Możliwe jest tylko usunięcie.

Do tabel *Zakwaterowanie* oraz *Platnosc* dane są wprowadzane automatycznie za pomocą odpowiednich wyzwalaczy.

5.2 Dokumentacja użytkownika

Aplikacja kliencka uruchamia się komendą *npm run dev*, po wykonaniu której projekt strony zostanie skompilowany i najprawdopodobniej uruchomiony pod adresem *http://127.0.0.1:3000* (możliwy jest inny numer portu). Do uruchomienia potrzebne są wszystkie biblioteki z pliku *package.json*, które można zainstalować komendą *npm install*.

Strona serwerowa aplikacji jest uruchamiana komendą *node index.js*. Do uruchomienia potrzebne są wszystkie biblioteki z pliku *package.json*, które można zainstalować komendą *npm install*.

Poszczególne podstrony i ich funkcjonalności są opisane w punkcie 4.1.

Aplikacja posiada testowych użytkowników z poniższymi danymi logowania:

- Zwykły użytkownik: login: *user* hasło: *user*
- Administrator: login: *admin* hasło: *admin*
- Pracownik (logowanie na podstronie dla pracowników): login: *prac* hasło: *prac*

5.3 Wykaz literatury

- <https://nextjs.org/docs/getting-started>
- <https://reactdatepicker.com/>
- <https://expressjs.com/en/guide/routing.html>
- <https://github.com/brianc/node-postgres/tree/master/packages/pg-pool>
- <https://tailwindcss.com/docs/installation>
- <https://www.mockaroo.com/>
- <https://github.com/pmndrs/zustand>
- <https://axios-http.com/docs/intro>
- <https://react-hook-form.com/get-started>