WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie

PROJEKT_Szkolenie techniczne 3

Prowadzący:
Piechocki Łukasz

Student:
Artemii Sniko w68316

📘 Dokumentacja Projektu – API Serwisu Hotelowego

📑 Spis treści

---

1. 📌 Opis stosu technologicznego

Element	Opis
Język	JavaScript
Środowisko	Node.js
Framework	Express – tworzenie RESTful API
ORM	Sequelize – mapowanie obiektowo-relacyjne
Baza danych	PostgreSQL
Testowanie	Jest + Supertest (opcjonalne)
Narzędzia	Postman, curl – testowanie punktów końcowych

---

2. ⚙️ Instrukcja uruchamiania aplikacji

Wymagania:

- Node.js (>= v16)
- PostgreSQL (>= v13)
- Git

Kroki instalacji:

bash

CopyEdit

```
# 1. Klonowanie repozytorium
git clone <URL-repozytorium>
cd hotel-service-api

# 2. Instalacja zależności
npm install

# 3. Konfiguracja bazy danych
# Upewnij się, że PostgreSQL działa i utwórz bazę danych, np. hotel_db
# Skonfiguruj połączenie w pliku: models/index.js lub .env

# 4. Uruchomienie serwera
npm start
```

Po uruchomieniu:

- ✅ Połączenie z bazą danych zostanie nawiązane.
- ✅ Serwer uruchomi się na porcie 3000.

---

3. 🗂️ Diagram bazy danych (ER)

plaintext

CopyEdit

```
+-------------+          1     *          +------------------+
|  Customer   | ------------------------------> |   Reservation    |
+-------------+                          +------------------+
| id (PK)     |                          | id (PK)          |
```

| name | | roomNumber |

| email | | date |

+----------+ | status |

| customerId (FK) |

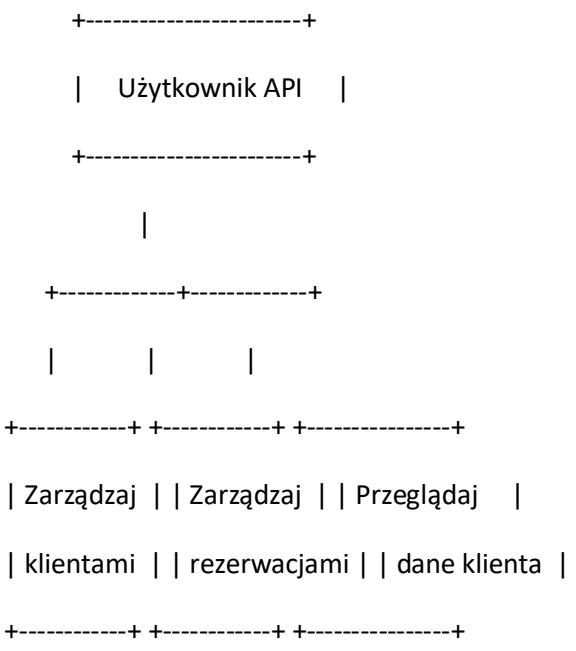
+----------+

---

4. 🧩 Diagram UML przypadków użycia

plaintext

CopyEdit



---

5. 🧱 Opis architektury back-endu

📁 Struktura katalogów:

bash

CopyEdit

hotel-service-api/

├— app.js

├— models/

```
|   ├── index.js
|   ├── customer.js
|   └── reservation.js
├── controllers/
|   ├── customerController.js
|   └── reservationController.js
├── routes/
|   ├── customerRoutes.js
|   └── reservationRoutes.js
└── test/
    └── customer.test.js
```

🔹 **Główne komponenty:**

🔸 **Serwisy:**

- **Customer Service** – zarządzanie klientami (CRUD)
- **Reservation Service** – zarządzanie rezerwacjami powiązanymi z klientem

🔸 **Modele danych:**

- **Customer** – id, name, email
- **Reservation** – id, roomNumber, date, status, customerId

🔸 **Kontrolery:**

- Obsługują logikę tworzenia, pobierania, aktualizacji i usuwania danych.

---

6. 🧪 Opis przypadków testowych (Gherkin)

Przykład: Tworzenie nowego klienta

gherkin

CopyEdit

Funkcja: Tworzenie klienta

  Aby dodać nowego klienta do bazy

Jako użytkownik API

Chcę wysłać żądanie POST z danymi klienta

Scenariusz: Poprawne dane klienta

Gdy wysyłam żądanie POST na /api/customers z imieniem i adresem e-mail

Wtedy otrzymuję odpowiedź 201 Created

I dane nowego klienta są zwracane w odpowiedzi

Scenariusz: Brak danych

Gdy wysyłam żądanie POST bez imienia

Wtedy otrzymuję odpowiedź 500 lub 400 z komunikatem o błędzie

---

7. 📚 Wykaz wykorzystanej literatury i źródeł

- Oficjalna dokumentacja Node.js
- Dokumentacja Express.js
- Dokumentacja Sequelize
- Dokumentacja PostgreSQL
- Przykłady z Postman oraz MDN Web Docs
- Kursy i materiały z platform edukacyjnych: Udemy, FreeCodeCamp