przedmiotu: Programowanie Aplikacji Internetowych

Tytul projektu: Aplikacja do rezerwacji sal konferencyjnych z integracją z Microsoft Outlook

Cel projektu

- Celem projektu było stworzenie aplikacji webowej umożliwiającej przeglądanie i rezerwowanie sal konferencyjnych w firmie, przy jednoczesnej synchronizacji danych z kalendarzem Microsoft Outlook. System łączy się z kalendarzami sal (resource mailboxes) poprzez Microsoft Graph API.

Główne funkcjonalności aplikacji

Przegląd sal konferencyjnych:	•Lista sal powiązana z danym budynkiem •Informacja o dostępności, pojemności i nazwie sali
Wyświetlanie kalendarza sali:	•Dynamiczny kalendarz (FullCalendar.js) z aktualnymi wydarzeniami pobieranymi z Microsoft Outlook •Widok dzienny, tygodniowy, miesięczny
Tworzenie rezerwacji:	•Formularz rezerwacji: nazwa, czas rozpoczecia i zakończenia, e-mail organizatora •Tworzenie wydarzenia w kalendarzu Outlook danego zasobu (sali) z użyciem Microsoft Graph API •Zapisywanie danych rezerwacji lokalnie w bazie MySQL
Usuwanie rezerwacji:	•Usuwanie możliwe tylko przez organizatora wydarzenia •Synchronizacja usunięcia z kalendarzem Outlook (Graph API)
Autoryzacja użytkownika:	•Logowanie przez Microsoft Azure (OAuth2 z MSAL) •Obsługa tylko delegowanych uprawnień
Responsywność:	•Aplikacja dostosowana do urządzeń mobilnych (tablety w salach konferencyjnych)

Architektura aplikacji

Frontend:

HTML, CSS, JavaScript

Biblioteka: FullCalendar.js

Backend:

Java, Spring Boot

Spring Security + OAuth2 Login (MSAL)

Spring Data JPA + Hibernate

Baza danych:

MySQL

Zewnętrzne API:

Microsoft Graph API (kalendarz Outlook, informacje o wydarzeniach)

Github

Zmiana repozytorium w związku ze zmianą logiki aplikacji

- W trakcie rozwoju projektu aplikacji do rezerwacji sal konferencyjnych podjęto decyzję o gruntownej przebudowie logiki działania systemu. W związku z tym projekt został przeniesiony do nowego repozytorium, a wcześniejsze rozwiązania zastąpiono uproszczoną i bardziej spójną architekturą.

Repozytorium pierwotne: **https://github.com/jbialek99/ConferenceHallPc.git**

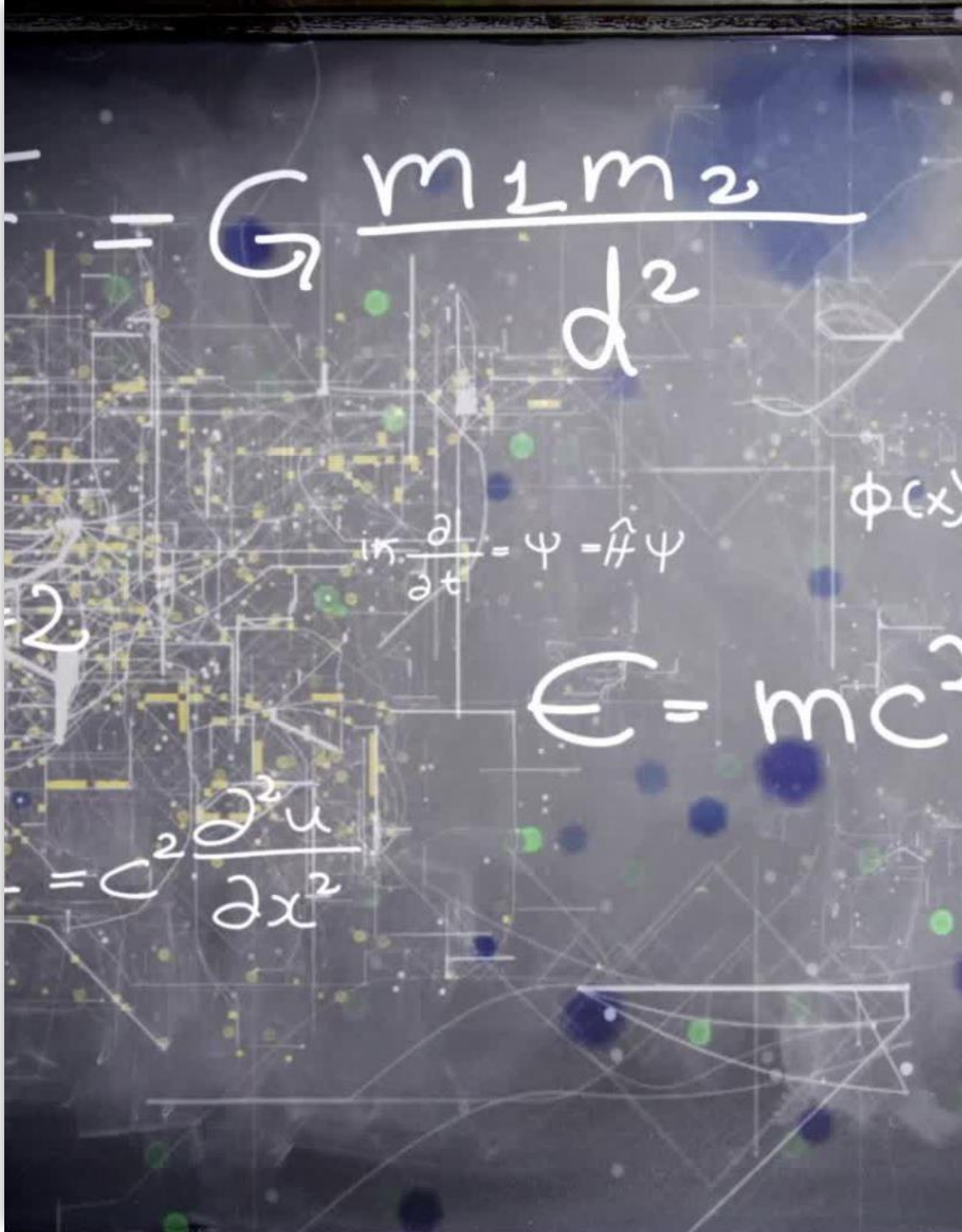
- Zawiera pierwszą wersję projektu, rozwijaną z wykorzystaniem wielu gałęzi (branchy) i commitów. Testowano tam różne koncepcje działania aplikacji, w tym m.in. sposoby obsługi rezerwacji oraz integracji z kalendarzem Microsoft Outlook przy użyciu Graph API.

Repozytorium aktualne: **https://github.com/jbialek99/ReservtionSystem.git**

- To aktualna i docelowa wersja aplikacji, w której zastosowano uproszczoną oraz zoptymalizowaną architekturę. Ujednolicono sposób integracji z Microsoft Graph API, usprawniono obsługę kalendarzy zasobów (resource mailboxes), a logika tworzenia oraz synchronizacji rezerwacji została znacznie uproszczona i ustabilizowana.
- Zmiana repozytorium była niezbędna, by uporządkować kod, zwiększyć jego przejrzystość i umożliwić dalszy rozwój aplikacji w oparciu o stabilne, dobrze zaprojektowane rozwiązania technologiczne.

Proces realizacji projektu

- Etap 1: Wstępna analiza i wybór technologii
- Etap 2: Budowa modelu danych (rezerwacje, sale, budynki)
- Etap 3: Implementacja backendu z integracją MS Graph
- Etap 4: Tworzenie frontendowej części aplikacji z kalendarzem
- Etap 5: Testy lokalne i integracyjne
- Etap 6: Wdrożenie funkcji logowania i obsługi błędów (np. kolizje rezerwacji)



Moduły aplikacji

Moduł zarządzania salami

Pobieranie sal z bazy

Filtrowanie sal po budynku

Moduł rezerwacji

Tworzenie/edycja/usuwanie rezerwacji

Weryfikacja właściciela (organizatora)

Moduł integracji z Outlook (Graph API)

Tworzenie wydarzeń w kalendarzach sal (resource mailbox)

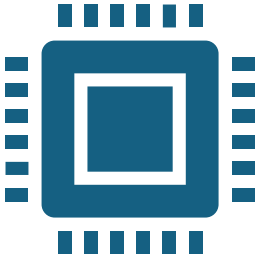
Usuwanie tylko przez organizatora

Moduł frontendowy

Widok kalendarza sali (FullCalendar.js)

Dynamiczna obsługa wyboru sali

Instrukcja instalacji



Wymagania:

Java 17+, Maven

MySQL

Konto Microsoft Azure z zarejestrowaną aplikacją (Graph API)

Krok po kroku:

Skonfiguruj bazę danych MySQL i wypełnij dane sal

Uzupełnij application.yml danymi aplikacji z Azure

Uruchom backend: mvn spring-boot:run

Otwórz przeglądarkę: http://localhost:8081

Instrukcja użytkowania

- Zaloguj się przez konto Microsoft
- Wybierz salę z listy przypisanej do budynku
- Przejdź do widoku kalendarza sali
- Dodaj nową rezerwację podając temat, czas i swój e-mail
- Edytuj lub usuń tylko swoje rezerwacje

Podsumowanie

- Projekt stanowi praktyczne zastosowanie integracji aplikacji webowej z zewnętrznym systemem jakim jest Microsoft Outlook. Pozwala na zarządzanie rezerwacjami w sposób zsynchronizowany, intuicyjny i przyjazny użytkownikowi. Aplikacja może być wdrażana w środowiskach firmowych jako lokalny system rezerwacji sal konferencyjnych.

- Główne tematy do poruszenia w prezentacji

1. **Rejestracja aplikacji w Microsoft Azure (Azure AD App Registration)**

- **Cel:** umożliwienie aplikacji autoryzowanego dostępu do danych użytkownika oraz kalendarzy sal w Microsoft Graph API.
- **Kroki:**
- Zalogowanie się do Azure Portal.
- Przejście do **Azure Active Directory → App registrations → New registration**.
- Podanie:
  - Nazwy aplikacji (np. *RezerwacjaSal*).
  - Typu konta (np. „Accounts in this organizational directory only”).
  - Adresu URI przekierowania (np. `https://localhost:8081/login/oauth2/code/azure` lub Twój adres z ngrok).
- Zapisanie **Client ID** i **Tenant ID**.
- **Dalsza konfiguracja:**
- Przejście do zakładki **API permissions**:
  - Dodanie delegowanych uprawnień do: `Calendars.ReadWrite`, `User.Read`, `email`, `offline_access`, `openid`, `profile`.
- Wygenerowanie **Client Secret** (Certificates & secrets → New client secret).
- **Efekt:** Aplikacja może logować użytkownika przez Microsoft i uzyskać dostęp do kalendarza Outlook.

2. **Utworzenie konta administratora i zakup licencji**

Cel: zapewnienie pełnego dostępu do zasobów organizacyjnych oraz utworzenie konta technicznego do integracji z Microsoft Graph API.

Kroki:

Utworzenie konta administratora w domenie Microsoft 365 (np. admin@twojadomena.onmicrosoft.com).

Przejście do **Microsoft 365 Admin Center** i zakup licencji typu **Microsoft 365 Business Standard / E3 / E5** (ważne: zawiera dostęp do Exchange i kalendarzy).

Przypisanie licencji administratorowi oraz utworzonym kontom typu **resource mailbox** (sala1, sala2 itd.).

Efekt: Aplikacja i konta zasobów (sala1@…) mają aktywne skrzynki pocztowe, które mogą być używane do rezerwacji sal przez kalendarz Outlook.

3. **Utworzenie sal jako resource mailboxes (pokoje/sale konferencyjne)**

Cel: zarejestrowanie sal w organizacji jako specjalne skrzynki pocztowe, które automatycznie odbierają zaproszenia i pojawiają się w kalendarzu.

Kroki:

Zalogowanie się do Exchange Admin Center (EAC).

Przejście do zakładki Resources.

Utworzenie nowej skrzynki typu Room mailbox:

Nazwa: Sala 1

E-mail: sala1@twojadomena.onmicrosoft.com

Typ: Room

Powtórzenie dla pozostałych sal (sala2, sala3...).

Opcjonalnie: Można dostosować ustawienia autoodpowiedzi, zatwierdzania zaproszeń itp.

Efekt: Sale stają się pełnoprawnymi zasobami, które można rezerwować z poziomu Outlooka i zintegrowanych aplikacji.

4. **Konfiguracja Azure AD i ról dostępowych**

- **Cel:** zapewnienie właściwych poziomów dostępu użytkownikom i aplikacjom.
- **Kroki:**
- Przejście do **Azure Active Directory → Users & Groups**.
- Utworzenie grupy np. `Użytkownicy aplikacji rezerwacyjnej`.
- Przypisanie użytkowników (np. `jakub@...`) do tej grupy.
- W razie potrzeby przypisanie ról (np. Application Administrator, Exchange Administrator) dla kont administracyjnych.
- Jeśli aplikacja ma działać w trybie **application permissions** (client credentials flow) — przypisanie zgód administratora do uprawnień w App Registration → API permissions → Grant admin consent.
- **Efekt:** Zarządzanie dostępem jest scentralizowane, a aplikacja ma autoryzowany dostęp do kalendarzy organizacyjnych.

5. Integracja z Microsoft Graph API

- **Cel:** pobieranie i zapisywanie wydarzeń w kalendarzach Outlooka przypisanych do sal.
- **Ważne funkcjonalności:**
- Tworzenie wydarzeń: `POST /me/events` lub `POST /users/{roomEmail}/calendar/events`
- Pobieranie rezerwacji: `GET /users/{roomEmail}/calendarview`
- Usuwanie wydarzeń: `DELETE /users/{roomEmail}/events/{eventId}`
- **Używane dane:**
- `access_token` z logowania MSAL
- Adres e-mail sali
- Obiekt `event` zawierający datę, czas, temat, lokalizację, organizatora
- **Efekt:** Pełna synchronizacja danych między aplikacją a kalendarzem Outlook — rezerwacje tworzone z aplikacji pojawiają się natychmiast w kalendarzu sali.

Działanie aplikacji przez Ngrok – uwagi konfiguracyjne

- Aplikacja działa w środowisku testowym z wykorzystaniem tunelu **Ngrok**, co umożliwia tymczasowy, publiczny dostęp do lokalnie uruchomionej aplikacji.
- 🔧 **Ważna uwaga dotycząca konfiguracji Azure AD (Entra ID):**
- Po każdorazowym **uruchomieniu nowego tunelu Ngrok**, generowany jest nowy adres URL (np. `https://abcd1234.ngrok-free.app`).
- Aby zapewnić poprawne działanie logowania i autoryzacji przez Microsoft (OAuth2), należy **zaktualizować redirect URI** w ustawieniach zarejestrowanej aplikacji w Azure.
- 📌 Aktualizacja konfiguracji w Azure jest wymagana **tylko wtedy**, gdy zakończono poprzednią sesję Ngrok i rozpoczęto nową (z nowym adresem URL).