

*Підкаура О.А., студент 3-го курсу
фізико-математичного факультету
Житомирський державний
університет імені Івана Франка,
Житомир*

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОЇ ІНТЕГРАЦІЇ РОЗКЛАДУ З GOOGLE КАЛЕНДАРЕМ НА БАЗІ МОВИ ПРОГРАМУВАННЯ GO

Інтеграція розкладу занять університету з Google Calendar є актуальною задачею для вищих навчальних закладів, які прагнуть забезпечити студентам та викладачам швидкий і зручний доступ до актуального розкладу занять. Автоматизоване оновлення розкладів дозволяє зменшити кількість ручної роботи, уникнути помилок і зробити інформацію про події доступною у зручному форматі, також це дозволяє ефективно організовувати час. Google Calendar API надає можливості для інтеграції сторонніх систем, що дає змогу автоматично додавати, редагувати та видаляти події.

Одним із можливих підходів до вирішення цієї задачі є розробка спеціалізованої системи, яка буде отримувати розклад із зовнішніх джерел, аналізувати його та синхронізувати з Google Calendar. Для реалізації такого рішення необхідно врахувати кілька ключових аспектів: вибір технологій, спосіб отримання даних та механізми синхронізації. Важливо розробити архітектуру системи, яка дозволить обробляти розклад у зручному форматі, кешувати дані для швидкого доступу та мінімізувати кількість звернень до API, щоб не перевищувати ліміти запитів Google.

З технічної точки зору, одним із оптимальних варіантів для розробки такої системи є використання мови програмування Go. Вона забезпечує високу швидкість компіляції та ефективну продуктивність

під час виконання, підтримує роботу з мережевими запитами та має зручні бібліотеки для роботи з API. Бібліотека golang.org/x/oauth2 може використовуватися для аутентифікації через OAuth 2.0, а google.golang.org/api/calendar/v3 – для взаємодії з Google Calendar API. Крім того, важливо обрати відповідний формат збереження даних. SQLite може бути використана як локальне сховище, оскільки вона легка та ефективна для невеликих обсягів даних, тоді як для більших масштабів може бути доцільним використання PostgreSQL чи іншої реляційної бази даних.

Сам процес інтеграції можна реалізувати через періодичне отримання розкладу із сайту або іншого джерела, його парсинг та перевірку на зміни. Це можна зробити за допомогою використання API, якщо дані доступні у форматі JSON або XML. У разі виявлення нових або змінених подій система повинна надсилати оновлення до Google Calendar API. Важливо передбачити механізм обробки помилок та логування, щоб відстежувати проблеми у процесі синхронізації, також важливо обробляти зміни в розкладі і надсилати сповіщення користувачам.

Ще одним критично важливим аспектом є планування автоматичного виконання завдань. Для цього можна використати вбудований у Go механізм `time.Ticker`, що дозволить виконувати запити з певним інтервалом, наприклад, раз на годину або кілька разів на день. Для спрощення розгортання та масштабованості системи доцільно використовувати контейнеризацію за допомогою Docker. Створення Docker-образу дозволяє легко запускати програму на різних серверах без необхідності ручного налаштування середовища. Файл `Dockerfile` міститиме всі інструкції для автоматизованої збірки Docker-образів, а запуск контейнера буде здійснюватися через `docker-compose`, інструмент що дозволяє описати та управляти багатоконтейнерними Docker-застосунками.

Створення такої системи відкриває можливості для автоматизації розкладів у університетах, гарантуючи студентам і викладачам своєчасний доступ до актуальних даних про заняття та зміни в розкладі. Наступні дослідження можуть передбачати додавання нових функцій, інтеграцію з іншими сервісами календаря, такими як Microsoft Outlook, а також поліпшення процесів обробки даних. Також може бути реалізовано REST API, який надасть зовнішнім системам можливість отримувати розклад у зручному вигляді. У майбутньому можна буде

створити мобільний додаток або веб-інтерфейс для управління розкладом і його параметрами.

Список використаних джерел

1. Patharlagadda, P. P. (2020). *Deploy a Microservice Application using Docker Compose*. *North American Journal of Engineering Research*, 1(4).
2. Шилов, Е. Ю. (2024). Вебзастосунок «Schedule manager».