**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет**

**Кафедра радіоприймання та оброблення сигналів**

|  |  |
| --- | --- |
| «На правах рукопису»  004.432.2 | До захисту допущено:  В.о.зав. кафедри  \_\_\_\_\_ Андрій МОВЧАНЮК  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р. |

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Радіозв’язок і оброблення сигналів»**

**за спеціальністю 172 «Телекомунікації та радіотехніка»**

**на тему: «Програмне забезпечення дистанційного навчання з використанням месенджерів»**

Виконав: студент 2 курсу, групи РА-91мп

РАБОДЗЕЙ Максим Валерійович

Керівник: доцент, к.т.н. каф. РОС Мосійчук Віталій Сергійович

Рецензент: доцент, к.т.н. каф. РТПС Вишневий Сергій Валерійович

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Радіотехнічний факультет**

**Кафедра радіоприймання та оброблення сигналів**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіозв`язок і оброблення сигналів»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

\_\_\_\_ Андрій МОВЧАНЮК

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ р.

**ЗАВДАННЯ**

**на магістерську дисертацію студента**

**Рабодзея Максима Валерійовича**

1. Тема дисертаї «**Програмне забезпечення дистанційного навчання з використанням месенджерів**»

науковий керівник дисертації доцент, к.т.н. Мосійчук В. С. затверджені наказом по університету від «5» листопада 2020 р. №3223-с

2. Термін подання студентом дисертації 11 грудня 2020 року

3. Об’єкт дослідження можливості реалізації дистанційного навчання за допомогою Telegram API

4. Вихідні дані\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перелік завдань, які потрібно розробити Реалізувати функціонал для проведення дистанційного навчання за допомогою Телеграм API. Реалізувати виконання деякої рутинної роботи з підключення Google Classroom. Розробити авторизацію з використанням даних методик

6. Орієнтовний перелік ілюстративного матеріалу для написання рішення на фреймворці

7. Орієнтовний перелік публікацій \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9. Дата видачі завдання 02 вересня 2020 року

Календарний план

|  |  |  |  |
| --- | --- | --- | --- |
| №з/п | Назва етапів виконання  магістерської дисертації | Термін виконання етапів магістерської дисертації | Примітка |
| 1 | Отримання теми магістерської дисертації | 02.09.2020 | виконано |
| 2 | Розробка плану магістерської дисертації | 09.09.2020 | виконано |
| 3 | Початок збору матеріалів для досліджень | 23.09.2020 | виконано |
| 4 | Дослідження API та бібліотек | 10.10.2020 | виконано |
| 5 | Аналіз відомих фреймворків | 24.10.2020 | виконано |
| 6 | Дослідження відомих методів авторизації | 15.11.2020 | виконано |
| 7 | Розробка власного ПЗ | 27.11.2020 | виконано |
| 8 | Оформлення магістерської дисертації | 05.12.2020 | виконано |

Студент: Рабодзей М.В.



Науковий керівник: Мосійчук В.С.

# Реферат

Магістерська дисертація на тему «Програмне забезпечення дистанційного навчання з використанням месенджерів»

Обсяг пояснювальної записки 71 сторінки, вони містять 4 розділи, 24 ілюстрацій, 14 таблиці, 11 джерел в переліку посилань.

Метою даної роботи є створення програмного забезпечення для дистанційного навчання, при використанні ресурсів Google та Телеграм , що буде побудовано за допомогою мови програмування JavaScript та його модулів.

Ключові слова: фреймворк, LMS, сервер, авторизація.

# Abstract

Master's dissertation on "Distance learning software using messengers"

The volume of the explanatory note is 71 pages, they will contain 4 sections, 25 illustrations, 14 tables, 11 sources in the list of references.

The purpose of this work is to create software for distance learning using Google and Telegram resources, which will be built using the JavaScript programming language and its modules.

Keywords: framework, LMS, server, authorization

**Зміст**

[Реферат 4](#_Toc59442250)

[Abstract 5](#_Toc59442251)

[Вступ 8](#_Toc59442252)

[1. ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО НАВЧАННЯ 9](#_Toc59442253)

[1.1 Впровадження дистанційного навчання 9](#_Toc59442254)

[1.2 Системи керування вмістом для дистанційного навчання 9](#_Toc59442255)

[1.2.1 LMS Moodle 9](#_Toc59442256)

[1.2.2 Google Classroom 18](#_Toc59442257)

[1.3 Популярні месенджери та їх використання в навчальних цілях 23](#_Toc59442258)

[1.3.1 Месенджер Telegram 23](#_Toc59442259)

[1.3.2 Месенджер WhatsApp 24](#_Toc59442260)

[1.3.2 Різні типи ботів для авоматизованої взаємодії з користувачами 25](#_Toc59442261)

[1.3.3 Варіанти використання ботів у навчальних цілях. 26](#_Toc59442262)

[Висновки до розділу 1 27](#_Toc59442263)

[2. ОСНОВИ ІНТЕГРАЦІЇ СЕРВІСІВ РОЗРІЗНЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ 28](#_Toc59442264)

[2.1 Формати обміну даними розрізнених інформаційних систем 28](#_Toc59442265)

[2.1.1 Формат обміну даними JSON 28](#_Toc59442266)

[2.1.2 Формат обміну даними XML 29](#_Toc59442267)

[2.2 Відкритий програмний інтерфейс Google Clasroom 30](#_Toc59442268)

[2.2.1 Використання відкритого стандарту авторизації користувача 30](#_Toc59442269)

[2.4 Відкритий програмний інтерфейс Moodle 36](#_Toc59442270)

[2.4.1 Опис корисного функціоналу 37](#_Toc59442271)

[2.5 Відкритий програмний інтерфейс Telegram 38](#_Toc59442272)

[Висновки до розділу 2 39](#_Toc59442273)

[3. РЕАЛІЗАЦІЯ ІНТЕГРАЦІЇ СЕРВІСІВ ДИСТАНЦІЙНОГО НАВЧАННЯ З TELEGRAM 40](#_Toc59442274)

[3.1 Вибір технологій створення програмного забезпечення 40](#_Toc59442275)

[3.1.1 Cередовище для запуску JavaScript проекту 40](#_Toc59442276)

[3.2 Процедура авторизації OAuth 2.0 42](#_Toc59442277)

[3.2.1 Перенаправлення на сервер Google OAuth 2.0 44](#_Toc59442278)

[3.2.2 Згода користувача в Google на передачу персональних даних 45](#_Toc59442279)

[3.2.3 Оброблення відповіді сервера OAuth 2.0 46](#_Toc59442280)

[3.3 Отримання доступу до API Google Classroom REST за допомогою Postman 46](#_Toc59442281)

[3.3.1 Часткова відповідь 48](#_Toc59442282)

[3.3.2 Відповідь API на пакетний запит 51](#_Toc59442283)

[3.3.3 Обробка помилок API 52](#_Toc59442284)

[3.4 Отримання даних від API Moodle 53](#_Toc59442285)

[3.5 Створення бота в месенджері Telegram 54](#_Toc59442286)

[3.6 Функціонал програмного забезпечення дистанційного навчання 61](#_Toc59442287)

[Висновки до розділу 3 63](#_Toc59442288)

[4. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ 64](#_Toc59442289)

[4.1 Опис ідеї проекту (товару, послуги, технології) 64](#_Toc59442290)

[4.2 Технологічний аудит ідеї проекту 64](#_Toc59442291)

[4.3 Аналіз ринкових можливостей запуску стартап-проекту 65](#_Toc59442292)

[4.4 Розроблення ринкової стратегії проекту 68](#_Toc59442293)

[4.5 Розроблення маркетингової програми стартап-проекту 69](#_Toc59442294)

[Висновки до розділу 4 72](#_Toc59442295)

[ЗАГАЛЬНІ ВИСНОВКИ 73](#_Toc59442296)

[ПЕРЕЛІК ПОСИЛАНЬ 74](#_Toc59442297)

# Вступ

В період пандемії COVID-19 різко прийшлося адаптувати себе під новий ритм життя. Робочий процес став протікати дома і більшість повсякденних операцій відбувається в онлайн режимі: замовлення продуктів харчування, лікувальних препаратів і тому подібних речей. Навчальний процес постраждав чи не найбільше, оскільки система освіти не була готова до такого форсованого перебігу подій. В наслідок цього впровадження дистанційного навчання відбувалось хаотично та несистемно. Це призвело до того, що студенти вимушені навчатися дистанційно з використанням різних інформаційних систем одночасно, що однією з найвідчутніших болей сучасних студентів. Для вирішення цієї проблеми дистанційного навчання потрібно створити програмне забезпечення, що здатне об’єднати інформаційні системи та сервіси в єдиний інтерфейс взаємодії з студентами.

**Мета роботи** – забезпечення дистанційного навчання на основі інтеграції інформаційних систем.

Задачі дослідження:

1) Огляд сучасних інформаційних систем та сервісів для дистанційного навчання.

2) Методів інтеграції інформаційних систем та сервісів.

3) Реалізувати чат-бот месерджера з доступом до навчальних інформаційних систем

4) Розробити власний стартап проекту.

**Об’єкт дослідження** –процеси інтеграції та обміну даними між інформаційними системами.

# 1. ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

## 1.1 Впровадження дистанційного навчання

Задля отримання різних послуг, таких як служба доставки чи скористатися послугами таксопарку використовуються застосунки, які в свою чергу займають значний об’єм пам’яті на телефоні. Крім цього кожен із додатків потрібно відкривати окремо, звикати до інтерфейсу та функціоналу, що не є занадто зручним. Використовуючи месенджер більшість цих проблем зникає так як можна звести весь спектр замовлення послуг в одне єдине місце.

## 1.2 Системи керування вмістом для дистанційного навчання

### 1.2.1 LMS Moodle

Система управління навчанням Moodle (LMS) забезпечує викладачів усіх рівнів та галузей промисловості високофункціональним, гнучким та сумісним цифровим навчальним рішенням за допомогою технології з відкритим кодом.

Moodle розшифровується як Модульне об’єктно-орієнтоване динамічне навчальне середовище. Заснований і розроблений Мартіном Дугіамасом у 2002 році, Moodle був розроблений, щоб надати викладачам, адміністраторам та учням відкриту, надійну, безпечну та безкоштовну платформу для створення та забезпечення персоналізованих навчальних середовищ. Moodle - це зручна система управління навчанням (LMS), яка підтримує потреби у навчанні та навчанні для широкого кола установ та організацій по всьому світу [6].

На сьогодні Moodle є найбільш широко використовуваною системою управління навчанням у світі, в якій зареєстровано понад 100 000 реалізацій у всьому світі, що підтримують понад 150 мільйонів учнів. Проектом Moodle з відкритим кодом керує спеціалізована команда у штабі Moodle із головним офісом у Перті, Австралія та супутниковими офісами по всьому світу. Модульний характер і властива Moodle гнучкість роблять його ідеальною платформою як для академічного, так і для корпоративного рівня будь-якого розміру.



Рисунок 1.1 – Логотип Moodle

Moodle пропонує потужний набір інструментів, орієнтованих на учня, та середовищ для спільного навчання. Завдяки своїй гнучкості та масштабованості, Moodle був пристосований для використання в освіті, бізнесі, некомерційних, державних та багатьох інших системах громад будь-якого розміру.

Безкоштовно доступний для всіх користувачів, Moodle можна самостійно розмістити (якщо у вас є час, амбіції та талант) або ви можете довіритись Сертифікованому партнеру Moodle, який надасть першокласний хостинг та підтримку для вашого середовища Moodle.

Розроблений за допомогою простого та гнучкого користувальницького інтерфейсу та добре задокументованих ресурсів, Moodle може бути простим для викладачів та учнів. Moodle - це веб-версія, яка має сумісний з мобільними пристроями інтерфейс, завдяки чому вона доступна з будь-якої точки світу через різні веб-браузери чи пристрої. Moodle перекладено більш ніж на 120 мов і призначений для відповідності стандартам відкритості та доступності.

Крім того, модульна та сумісна конструкція Moodle дозволяє розробникам створювати плагіни, які можуть інтегрувати зовнішні програми для покращення можливостей системи та розширення можливостей основного продукту Moodle. У каталозі плагінів Moodle є понад 1400 плагінів, якими користується спільнота Moodle у вільному доступі, а база відкритого коду Moodle заохочує користувачів створювати власні, за потреби, для спільного використання із спільнотою з відкритим кодом.

*Moodle є відкритим кодом: що це означає і чому це важлив*

Під відкритим кодом розуміється основна технологія, на якій побудована Moodle, в першу чергу означає, що програмне забезпечення або код Moodle є у вільному доступі для завантаження, використання, налаштування, перерозподілу та модифікації.

Як проект з відкритим кодом, Moodle - це спільна робота, яка підтримується сильним світовим співтовариством, включаючи команду Moodle HQ, а також розробників, партнерів Moodle та користувачів по всьому світу.

Безкоштовно код знаходиться у вільному доступі без ліцензійної плати. Натомість ці кошти можна виділити на обслуговування та підтримку вашої системи управління вмістом. Коли ви платите лише за послугу, ви забезпечуєте найвищий рівень підтримки.

Рішення із закритим кодом часто ставлять клієнта в залежність від постачальника товарів та послуг. Якщо ваш рівень обслуговування погіршується, із закритим кодом у вас немає інших варіантів підтримки. Тільки компанія із закритим кодом може підтримувати власний продукт. Завдяки LMS із відкритим кодом ви маєте свободу не тільки налаштувати своє рішення відповідно до своїх потреб, але ви можете легко взяти свій код (який є вашим власником, а не постачальником) і перейти до іншого постачальника, якщо ви не отримуєте необхідний рівень підтримки.

*Безпека.* Поширена помилкова думка, що відкрите джерело означає меньший рівень безпеки. Навпаки, відкритий та видимий код дозволяє спільноті з відкритим кодом знаходити та відзначати будь-які проблеми безпеки. Це означає, що збої в системі безпеки розпізнаються набагато швидше, а патчі повертаються майже відразу. Тож насправді, чим більше очей буде зосереджено на коді, тим краще!

*Налаштування*. Як уже згадувалося вище, відкритий код пропонує користувачам повну гнучкість у створенні власної, спеціально розробленої платформи електронного навчання для задоволення потреб конкретної організації чи установи.

*Інновації*. Програмне забезпечення з відкритим кодом забезпечує гнучкість, необхідну для налаштування та адаптації відповідно до мінливих потреб у навчанні, що дозволяє адміністраторам залишатися на передовій сучасних технологій та найкращих практик для ефективного досягнення цілей електронного навчання.

*Масштабована*. У міру зростання та масштабу установ чи організацій важливо, щоб ваш LMS міг рости разом із вами та пристосовуватись до задоволення нових потреб. Відкритий код дозволяє легко масштабувати ваші програми навчання та навчання в Інтернеті.

*Докази на майбутнє*. Можливість масштабувати, налаштовувати та впроваджувати інновації означає, що ви також можете забезпечити свої інвестиції в LMS у майбутньому, створивши рішення електронного навчання, яке може адаптуватися до мінливих потреб. Окрім того, оскільки відкритий код є модульним за своєю суттю, ви маєте можливість інтегрувати свою систему управління вмістом із безкоштовними інструментами, технологіями та рішеннями на ринку, щоб створити надійну та сумісну екосистему електронного навчання. У цьому відношенні відкрите джерело захищає не лише вашу систему управління вмістом, але і всю вашу навчальну інфраструктуру.

*Moodle для навчання на робочому місці*

У сучасних стрімких робочих середовищах забезпечення спрощеного процесу навчання та ефективної постійної освіти є невід’ємною частиною гарної бізнес-стратегії. Добре навчені співробітники допомагають рухати бізнес вперед до досягнення цілей та завдань компанії. eLearning забезпечує підготовку працівників шляхом сприяння співпраці та надання персоналу будь-де та будь-коли доступу до освітніх ресурсів, необхідних їм для досягнення кращих позицій.

Moodle пропонує гнучке рішення для навчання, яке допомагає навчати та розширювати можливості співробітників.

*Приклади використання Moodle на робочому місці включають:*

* Відповідність
* Навчання на основі компетентності
* Рольова підготовка співробітників
* Запуски продуктів та послуг
* Спільноти практики / Культура навчання
* Включення
* Безпека на робочому місці
* Безперервна освіта та професійний розвиток
* Програми сертифікації
* Збереження та найм внутрішньо / мобільність вгору
* електронна комерція
* Розширене підприємницьке навчання

*Параметри та найкращі практики хостингу Moodle*

Оскільки код Moodle є у вільному доступі, Moodle пропонує безліч варіантів хостингу на вибір, включаючи можливість самостійного розміщення або співпраці із сертифікованим партнером Moodle (або несертифікованим постачальником).

*Самостійний Moodle*

Самостійне розміщення Moodle означає, що ваша команда повністю володіє та управляє кодом вашого веб-сайту. Ви маєте повний контроль над кожним аспектом СУО, від впровадження до дизайну до тем та плагінів, які ви вирішили встановити.

Хоча це звучить як найдоступніший та найзручніший спосіб розміщення вашого LMS, проте це, як правило, не найкращий метод, коли вам бракує пропускної здатності або досвіду. Часто потрібні розширені ресурси, щоб не лише підтримати та управляти таким складним, але потужним рішенням, але й переконатися, що воно відповідає потребам у навчанні, відображає вашу організацію або виконує те, що ви хочете. Багато організацій, які самостійно приймають Moodle, мають команду експертів з високою винагородою, що займаються підтримкою платформи.

Рішення про самостійне розміщення не слід приймати легковажно. Для більшості користувачів Moodle стане додатком на рівні підприємства з сотнями користувачів залежно від його щоденної готовності виконувати свої завдання. Без внутрішньої експертизи, здатної підтримувати, модернізувати та підтримувати кінцевих користувачів, самохостинг може швидко стати надзвичайним завданням.

*Керований хостинг Moodle / Cloud*

Як уже згадувалося, управління Moodle та використання його надійних функціональних можливостей - це не те, що кожна команда L&D має час чи досвід, щоб взяти на себе. Багато хто звертається за додатковою допомогою до хмарних хостинг-партнерів, щоб полегшити тягар LMS.

З партнером Moodle ви можете витратити менше часу, зосереджуючись на управлінні своєю системою, а більше на забезпеченні залучення електронного навчання для своїх виборців.

*Самообслуговування проти керованих послуг Moodle*

Самостійний хостинг безкоштовний; однак керовані послуги Moodle зазвичай заощаджують з точки зору персоналу та продуктивності. Для самостійного розміщення потрібні величезні внутрішні знання та робоча сила (і години роботи) для підтримки вашого веб-сайту Moodle. Партнер Moodle може полегшити навантаження, ставши продовженням вашої ІТ-команди, відповідальної за ваше обслуговування LMS, так що ваші адміністратори можуть менше зосереджуватися на технічному обслуговуванні, а більше на наданні залученого електронного навчання кінцевим користувачам.

*Захист даних*. Організації, які самостійно розміщують, часто зберігають дані, які Moodle виробляє у власних центрах обробки даних. Сюди входять інтелектуальна власність, приватні дані студентів і навіть медичні картки. Поміщення цих даних "у хмару" може здатися занадто великим ризиком. Однак робота з партнером Moodle надає вам додаткову безпеку, оскільки ваш хост матиме власні заходи безпеки навколо серверів, щоб забезпечити максимальний час безвідмовної роботи та загальний захист LMS та даних. Крім того, дослідження показали, що хмарний хостинг насправді є більш безпечним, ніж самохостинг.

*Інфраструктура масштабування*. Сертифікований партнер може допомогти з коригуваннями та вдосконаленнями, необхідними для масштабування вашої системи управління вмістом у міру зростання та розвитку вашої організації.

*Налаштування та консультації*. Співпрацюючи з експертом, ви отримуєте рекомендації щодо найкращих рішень для задоволення ваших конкретних потреб у електронному навчанні. Кожна організація відрізняється, а це означає, що кожну СУО потрібно налаштовувати з урахуванням унікальних кінцевих цілей організації.

*Інтеграції*. Організації, які самостійно приймають Moodle, часто інтегруються з іншими кампусними системами, такими як Інформаційні системи студентів (SIS), ERP, CRM та інші власні послуги, такі як Big Blue Button, Kaltura тощо. Партнер Moodle може допомогти вам інтегрувати ваш LMS за допомогою цих систем, щоб забезпечити безперебійний потік інформації між вашими найважливішими системами.

Оновлення та резервні копії. Самостійно розміщені користувачі Moodle несуть відповідальність за планування, планування та тестування своїх оновлень сайту Moodle. Багато установ також повинні планувати оновлення навколо свого напруженого академічного календаря. Партнер хостингу планує оновлення відповідно до розкладу організації, досліджує нові функції та ретельно тестує нові версії, що дозволяє спеціалістам, які навчаються, натомість зосередитися на вивченні того, як нова версія покращить методи викладання та навчання та досвід користувачів. Крім того, організації можуть бути впевнені, що резервні копії проводяться регулярно і доступні за потреби.

Будьте в курсі подій. Співпраця з партнером Moodle дозволяє вам не лише залишатися в курсі версій продуктів, керуючи оновленнями, але й полегшує вам бути в курсі найновіших тенденцій, технологій, оновлень безпеки та інновацій на ринку електронного навчання. Експерти Moodle та eLearning, хостинг-партнер, можуть допомогти вам орієнтуватися на величезний процес управління вашим LMS, а також бути в курсі останніх технологій та найкращих практик. Маючи це на увазі, ваш партнер Moodle допоможе вам створити власний LMS та надасть рекомендації щодо побудови вашої розширеної екосистеми навчання.

### 1.2.2 Google Classroom

Google Classroom – це безкоштовний онлайн-сервіс, який дозволяє викладачам та студентам легко обмінюватися файлами між собою. Викладачі можуть розміщувати завдання для учнів, а потім оцінювати їх, не вимагаючи нічого друкувати.

Це також служить способом спілкування. Вчителі можуть розміщувати оголошення та майбутні завдання, а також надсилати листи на електронну пошту як учнів, так і їх батьків.



Рисунок 1.2 – Логотип Google Classroom

Google описує Google Classroom як "контроль місій для вашого класу", і це може бути найпростіший спосіб подумати про це. Простіше кажучи, це платформа, яка пов’язує інструменти *Google G Suite* для викладачів та студентів. Він також діє як цифровий органайзер, де вчителі можуть зберігати навчальні матеріали та ділитися ними зі студентами - усі безпаперові. Звідти ви можете вибрати функції, які хочете включити. Ця гнучкість та її безперервна інтеграція з популярними інструментами Google, ймовірно, зробили Google Classroom одним із найпоширеніших інструментів EdTech сьогодні.

Технічно Google Classroom не є системою управління вмістом (Cotent Menegment Systems, CMS). Google Classroom не є окремою системою управління навчанням (LMS), системою управління курсами (Courses Menegment Systems, CMS) або інформаційною системою для студентів (SIS). Тим не менш, Google регулярно додає нові функції до Google Classroom. Наприклад, у червні 2019 року Google оголосив, що школи незабаром зможуть синхронізувати нові функції оцінювання інструменту з існуючою інформаційною системою для учнів. Оскільки Google продовжує додавати функції, цілком ймовірно, що він може почати виглядати та функціонувати, більше схожий на СУО. Але наразі найкраще думати про інструмент як про універсальний магазин для організації класів.

Google Classroom може виткористовувати будь-хто, оскільки він представлений як безкоштовний сервіс для всіх, хто має особистий обліковий запис Google, а також безкоштовний для організацій, які використовують *G Suite for Education* або *G Suite* для некомерційних організацій. У більшості випадків викладачі та студенти можуть отримати доступ до Google Classroom за допомогою облікового запису Google, наданого їхньою школою. Хоча вчителі та учні шкіл є основними користувачами Google Classroom, є також функції, якими можуть користуватися адміністратори, сім'ї та домашні школярі.

Оскільки це досить гнучка платформа, освітяни використовують її функції різними способами. За допомогою Google Classroom вчителі можуть:

Впорядковувати роботу з класами класами. Платформа інтегрується з іншими інструментами Google, такими як Документи, Диск та Календар, тому існує безліч вбудованих "ярликів" для завдань управління класом. Наприклад, якщо ви опублікуєте завдання із датою виконання, воно автоматично додається до календаря занять, щоб його бачили ваші учні.

Організовувати, розподіляти та збирати завдання, матеріали курсу та роботи студентів у цифровій формі. Викладачі можуть також розміщувати завдання в декількох класах або змінювати та повторно використовувати завдання з року в рік.

Спілкуватися з учнями про їх класні роботи. Можуть використовувати платформу для розміщення оголошень та нагадувань про доручення, і легко зрозуміти, хто виконав чи не виконав свою роботу. Ви також можете приватно зареєструватися з окремими студентами, відповісти на їхні запитання та запропонувати підтримку.

Забезпечувати студентам своєчасний зворотній зв’язок щодо своїх завдань та оцінок. В Google Classroom можна використовувати *Google Forms* для створення та обміну тестами, які автоматично оцінюються, коли учні здають їх. Ви не тільки витратите менше часу на оцінювання, але й ваші студенти отримають миттєвий відгук про свою роботу.

Не завжди. Однак Google пропонує преміум-функції *Hangouts Meet* безкоштовно викладачам та студентам, які перебувають вдома під час пандемії коронавірусу, що дозволяє проводити віртуальні зустрічі до 250 людей, а також пряму трансляцію. Окрім можливостей відеозйомки в режимі реального часу, функція запису у Hangouts Meet дає викладачам простий інструмент для створення попередньо записаних відеозаписів уроків для перегляду студентами у вільний час. Вчителі можуть розпочати роботу з *Hangouts Meet* за допомогою цих корисних відеоуроків від учительського центру *Google for Education*.

Основний процес налаштування Google Classroom досить інтуїтивний, навіть для початківців користувачів. Учительський центр Google пропонує кілька посібників для початку роботи - це найкращий вибір, якщо ви шукаєте найсвіжіші відео та інформацію. На YouTube також є безліч підручників «зроби сам», які публікують викладачі та спеціалісти з технічної інтеграції. Багато з цих відео, створених викладачами, містять практичні поради та підказки, яких вони навчились користуючись платформою у власних класах.

*Що Google робить із даними студентів*

Як викладач, захист конфіденційності та даних ваших учнів, безумовно, повинен бути важливим, коли б ви вибирали цифровий інструмент для свого класу. Кожного разу, коли інструмент може збирати дані від студентів, важливо задати питання про те, як задіяні компанії забезпечують, використовують або зберігають дані студентів. Щоб отримати додаткову інформацію, не забудьте прочитати наш повний огляд конфіденційності Google Classroom.

Google заявляє, що конфіденційність даних та безпека є пріоритетом для всіх продуктів G Suite for Education. Однак вихователям слід пам’ятати, що батьки та сім’ї мають право відмовитись, якщо вони не хочуть, щоб їхні діти користувалися продуктами Google у школі. Перш ніж запустити Google Classroom, шкільні адміністратори та вчителі можуть захотіти створити альтернативний план для учнів, які можуть відмовитись.

Раніше деякі освітяни, сім'ї та адвокати висловлювали сумніви щодо здатності Google виконувати обіцянки щодо конфіденційності та захисту даних. Більше того, популярність брендування та продуктів Google у школах викликала питання щодо компромісів, які дозволяють Google будувати свій бренд у школах. Незалежно від того, користуєтесь ви Google Classroom чи ні, важливо змусити студентів критично думати про конфіденційність даних та комерціалізацію, яку ми спостерігаємо в різних аспектах нашого життя, включаючи наші класи.

*Як Google Classroom може підтримувати диференціацію в класі*

Google Classroom може допомогти впорядкувати формувальне оцінювання, що важливо для надання допомоги студентам, яким може знадобитися додаткова підтримка чи додаткові завдання. Наприклад, ви можете використовувати платформу для швидкого створення, розповсюдження та збору цифрових квитків на вихід або автоматичного оцінювання. Певним чином, Google Classroom може спростити та швидше збирати регулярні відгуки про успіхи ваших учнів. Звичайно, існує безліч інших інструментів оціночного формування, багато з яких зараз пропонують інтеграцію з Google Classroom.

Google Classroom також спрощує налаштування завдань для окремих учнів або для невеликих груп. Це означає, що вчителі можуть давати змінені або різні завдання певним учням або групам у класі. У вас також є можливість приватно зареєструватися у студента, щоб побачити, чи є у них запитання чи потрібна додаткова допомога. Можливість робити все це в Інтернеті може зробити зусилля вчителів щодо диференціації менш помітними для класу, що може бути корисним для учнів, які можуть почуватись виділеними.

З таким інструментом, як Google Classroom, або без нього, диференціація завжди буде предметом творчого вирішення проблем, і цього не існує нікому, ні «правильному» способу. На щастя, багато вчителів діляться своїми порадами, підказками та креативними рішеннями в Інтернеті.

Платформа була оновлена досить швидко з моменту її запуску, і Google продовжує регулярно впроваджувати нові функції, часто на основі відгуків викладачів. Протягом декількох років користувачі нарікають на відсутність у Google Classroom функцій оцінювання чи інструменту для створення рубрик. Google вислухав і випускає новий інструмент збору та оцінки робіт, який називається «Завдання», наприкінці 2019-2020 навчального року. Будь-хто, хто має доступ до *G Suite for Education*, може подати заявку на тестування бета-версії Завдання безкоштовно.

## 1.3 Популярні месенджери та їх використання в навчальних цілях

Миттєві повідомлення стали одним із найбільш популярних способів спілкування. Вони надають безпосередній простір, в якому можна спілкуватися один на один

Питання, який додаток для обміну повідомленнями використовувати є потенційно складним, хоча більшість користувачів використовують WhatsApp, Facebook Messenger, Telegram.

### 1.3.1 Месенджер Telegram

Основна функціональність Telegram така ж, як і у більшості інших програм обміну повідомленнями: можна надсилати повідомлення іншим користувачам Telegram, створювати групові розмови, телефонувати контактам та надсилати файли та наклейки.

Основною функцією Telegram є конфіденційність, і для цього вона використовує наскрізне шифрування. Це те, що заважає людям поза двосторонньою розмовою - будь то компанія, уряд, хакери чи хтось інший - бачити надіслане.

### 1.3.2 Месенджер WhatsApp

Основна привабливість WhatsApp полягає в тому, що він дозволяє надсилати та отримувати дзвінки та повідомлення, використовуючи лише підключення до Інтернету, а це означає, що додаток практично безкоштовний у використанні та ідеально підходить для міжнародних дзвінків. Немає плати за реєстрацію та надбавок для планування даних, про які слід турбуватися.

На перший погляд, WhatsApp може здатися не чим іншим, як програмою обміну текстовими повідомленнями, але може зробити набагато більше.

* *Голосові та відеодзвінки:* На додаток до голосових дзвінків, WhatsApp також пропонує відеодзвінки, включаючи групову функцію, яка дозволяє до восьми учасників на одному дзвінку.
* *Голосові повідомлення:* Можна записувати та надсилати голосові повідомлення в окремі чати або групові чати.
* *Безпечний обмін повідомленнями:* WhatsApp використовує наскрізне шифрування, стандарт безпечного спілкування, де лише люди, які обмінюються повідомленнями, можуть читати повідомлення.

Отже, для пересічної людини використання Telegram не обов’язко означає, що їх повідомлення є більш приватними або захищеними, ніж при використанні WhatsApp. Насправді, якщо вони не використовують секретні чати, обмін повідомленнями Telegram технічно є більш безпечним.

Найбільш популярним методом взаємодії користувача із каналами зв’язку, які надають можливість отримання та оброблення інформації являється чат бот.

Чат бот – програма, котра використовує інтерфейс месенджера для своєї роботи. Основною функцією якої програми є розпізнавання запитів користувача та коректне реагування на них. Перевагою таких ботів є те, що вони не вимагають окремих додатків та регістрації. В будь який час користувач може легко зв’язатися с ними [5].

### 1.3.2 Різні типи ботів для авоматизованої взаємодії з користувачами

Найбільш популярним методом взаємодії користувача із каналами зв’язку, які надають можливість отримання та оброблення інформації являється чат бот.

Чат бот – програма, котра використовує інтерфейс месенджера для своєї роботи. Основною функцією якої програми є розпізнавання запитів користувача та коректне реагування на них. Перевагою таких ботів є те, що вони не вимагають окремих додатків та регістрації. В будь який час користувач може легко зв’язатися с ними [5].

Виходячи з функціональних можливостей та характеру використання, можна розділити чат-боти на 6 типів. Крім того, додано приклади в кінці кожного типу як посилання для розгляду.

* + Чат-бот на основі меню / кнопки
  + Чат-бот на основі розпізнавання ключових слів
  + Контекстний чат-бот
  + Голосовий чат-бот
  + Послуга чат-бота
  + Чат-бот для обміну повідомленнями в соціальних мережах

*1. Чат-бот на основі меню / кнопки*

Чат-бот на основі меню або кнопки - це найпростіший тип чат-бота. Доступна заздалегідь створена база знань, і вам просто потрібно вибрати заздалегідь визначені кнопки.

Цілком можливо, що ваш запит не буде в списку, і бот, отже, не може допомогти, оскільки він не має можливості вийти за межі переліченої кнопки або меню. Іншими словами, ви можете розглядати його як "скриптованого чат-бота", обмеженого певними запитаннями-відповідями.

Боти почали з’являтися після того, як Telegram оголосив про новий API у 2015 році. Він дозволяє третім сторонам створювати боти, які використовуватимуть цей додаток для обміну повідомленнями як основний інтерфейс. Список довгий: багато розробників експериментували із наданим API для створення унікальних ботів для різних цілей.

Telegram - це платформа, що розвивається, яка славиться своєю безпекою, швидкістю доставки повідомлень та безліччю переваг для розвитку, які роблять спілкування приємним. Одним із таких прикладів є чат-боти, які працюють цілодобово та без вихідних, і миттю можуть надати безліч корисної інформації.

Уже існує велика кількість ботів, які допомагають людям отримувати інформацію на різні теми. На даному етапі ми розглянемо найкращі чат-боти, які Telegram може запропонувати.

### 1.3.3 Варіанти використання ботів у навчальних цілях.

Діти ростуть із планшетами в руках, тому навчання їх новим технологіям та програмам допоможе лише підготувати їх до ширшого світу. Не кажучи вже про те, що технологія може значно полегшити роботу вчителя.

Що стосується питання про те, як - на ринку існує кілька платформ для створення чат-ботів, які пропонують освітніх ботів, призначених для залучення студентів та надання короткої та швидкої, але цінної інформації.

Ці платформи інтегровані у програми Facebook, Twitter, Skype та інші соціальні медіа, щоб зробити навчання максимально цікавим та цікавим для студентів, і в той же час усунути необхідність перемикання між ними десятків інтернет-порталів та облікових записів електронної пошти.

Студенти можуть просто взаємодіяти з чат-ботом на бажаній платформі та миттєво отримати від нього відповідь.

## Висновки до розділу 1

Проаналізувавши популярні навчальні сервіси, а саме відокремивши основні проблеми які пов’язані із доступністю відкритого коду, наявністю документації, яка в свою чергу спрощує розуміння актуальності, а також введення LMS у повноцінну реалізацію проекту, можна зробити висновок, що перевагу мають системи контролю навчальним вмістом Google Classroom та Moodle, зокрема і тим що уже існують подібні рішення реалізації робіт, які за основу взяли саме цю систему.

# 2. ОСНОВИ ІНТЕГРАЦІЇ СЕРВІСІВ РОЗРІЗНЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

## 2.1 Формати обміну даними розрізнених інформаційних систем

Розробляючи програмне забезепечення часто можна зіштовхнутись із різними типами даних, наприклад інформація про користувача чи набір певних перекладів. При роботі з таким даними потрібно зберігати їх у зручному форматі, щоб потім можна було обробити і застосувати у потрібному місці.

### 2.1.1 Формат обміну даними JSON

JSON - це скорочення від "JavaScript Object Notation" і це спосіб зберігати інформацію в організованому та легкодоступному вигляді. У двох словах, це дає нам зручний для читання збір даних, до яких ми можемо отримати доступ [8].

З ростом веб-сайтів, що працюють на AJAX, для веб-сайтів стає все більш важливим можливість швидкого та асинхронного завантаження даних у фоновому режимі, не затримуючи надання сторінок. Переключення вмісту певного елемента в наших макетах, не вимагаючи оновлення сторінки, додає фактору “вау” у програми, не кажучи вже про додаткову зручність для користувачів. Через популярність та легкість соціальних медіа багато веб-сайтів покладаються на вміст таких сайтів, як Twitter, Flickr та інші. Ці веб-сайти надають RSS-канали, які легко імпортувати та використовувати на стороні сервера, але якщо ми спробуємо завантажити їх за допомогою AJAX, натрапляємо на стіну: можемо завантажувати RSS-канал лише тоді, коли ми вимагаємо його від той самий домен, на якому він розміщений.

### 2.1.2 Формат обміну даними XML

Розширювана мова розмітки (XML) використовується для опису даних. Стандарт XML - це гнучкий спосіб створення інформаційних форматів та електронного обміну структурованими даними через загальнодоступний Інтернет, а також через корпоративні мережі.

Код XML, офіційна рекомендація Консорціуму всесвітньої павутини (W3C), подібний до мови розмітки гіпертексту (HTML). І XML, і HTML містять символи розмітки для опису вмісту сторінки чи файлу. HTML-код описує вміст веб-сторінки (переважно текстові та графічні зображення) лише з точки зору того, як він повинен відображатися та взаємодіяти.

Дані XML відомі як самоопис або самовизначення, що означає, що структура даних вбудовується в дані, отже, коли дані надходять, немає необхідності попередньо створювати структуру для зберігання даних; це динамічно розуміється в XML. Формат XML може використовувати будь-яка особа чи група осіб чи компанії, які хочуть послідовно обмінюватися інформацією. XML насправді є простішим і простішим у використанні підмножиною Стандартної узагальненої мови розмітки (SGML), яка є стандартом для створення структури документа [7].

Основним будівельним блоком документа XML є елемент, який визначається тегами. Елемент має початковий і кінцевий теги. Всі елементи у документі XML містяться в самому зовнішньому елементі, відомому як кореневий елемент. XML також може підтримувати вкладені елементи або елементи всередині елементів. Ця можливість дозволяє XML підтримувати ієрархічні структури. Назви елементів описують зміст елемента, а структура - взаємозв'язок між елементами.

## 2.2 Відкритий програмний інтерфейс Google Clasroom

### 2.2.1 Використання відкритого стандарту авторизації користувача

API Google використовує протокол OAuth 2.0 для аутентифікації та авторизації. Google підтримує загальноприйняті сценарії OAuth 2.0, такі як веб-сервер, програми на стороні клієнта, інстальовані програми та пристрої з обмеженим введенням [3].

Для початку отримуються облікові дані клієнта OAuth 2.0 із консолі Google API. Потім клієнтська програма запитує токен доступу з сервера авторизації Google, витягує токен із відповіді та надсилає його до API Google, до якого потрібно отримати доступ. Для інтерактивної демонстрації використання OAuth 2.0 з Google (включаючи можливість використовувати власні облікові дані клієнта).

Усі додатки дотримуються базової моделі при доступі до API Google за допомогою OAuth 2.0 у вигляді п’яти кроків:

***1. Отримання облікових даних OAuth 2.0 із консолі Google API****.*  Відвідується консоль Google API, щоб отримати облікові дані OAuth 2.0, такі як ідентифікатор клієнта та секрет клієнта, відомі як Google, так і програмі. Набір значень залежить від типу програми, яку ми створюємо.

***2. Отримання токен доступу на сервері авторизації Google.*** Перш ніж програма зможе отримати доступ до приватних даних за допомогою Google API, вона повинна отримати токен доступу, який надає доступ до цього API. Один токен доступу може надавати різний ступінь доступу до декількох API. Змінний параметр, що називається область, контролює набір ресурсів та операцій, дозволених токеном доступу. Під час запиту на токен доступу наша програма надсилає одне або кілька значень у параметрі сфери.

Існує кілька способів зробити цей запит, і вони залежать від типу програми, яку ми створюємо [10].

Деякі запити вимагають крок автентифікації, коли користувач входить у свій обліковий запис Google. Після входу в систему користувача запитують, чи готовий він надати один або кілька дозволів, які вимагає наша програма. Цей процес називається згодою користувача.

Якщо користувач надає принаймні один дозвіл, сервер авторизації Google надсилає нашій програмі токен доступу (або код авторизації, який наша програма може використовувати для отримання токена доступу) та список обсягу доступу, наданого цим токеном. Якщо користувач не надає дозволу, сервер видає помилку.

Як правило, найкращою практикою є запит на обсяги поступово, у той час, коли потрібен доступ, а не наперед. Наприклад, програма, яка хоче підтримати збереження події в календарі, не повинна запитувати доступ до Календаря Google, поки користувач не натисне кнопку "Додати до календаря"; див. Додаткова авторизація.

***3. Обсяги доступу, надані користувачем***. Вимикаємо будь-які функції вашого додатка, які не можуть працювати без доступу до відповідного API.

Обсяг, включений у наш запит, може не збігатися з обсягом, включеним у нашу відповідь, навіть якщо користувач надав усі запитувані області. Звертаємся до документації кожного API Google, щоб дізнатись про обсяги, необхідні для доступу. API може зіставити кілька значень рядка області дії в одну область доступу, повертаючи однаковий рядок області для всіх значень, дозволених у запиті. Приклад: API Google People може повернути область https://www.googleapis.com/auth/contacts, коли програма вимагає, щоб користувач авторизував область https://www.google.com/m8/feeds/

***4. Надсилання токен доступу API****.* Після того, як програма отримує токен доступу, вона надсилає його Google API у заголовку запиту авторизації HTTP. Можна надсилати токени як параметри рядка запиту URI, але не рекомендую цього, оскільки параметри URI можуть потрапити до файлів журналів, які не є повністю захищеними. Крім того, хороша REST практика уникати створення непотрібних імен параметрів URI.

Токени доступу дійсні лише для набору операцій та ресурсів, описаних у обсязі запиту токена. Наприклад, якщо для API Google Calendar видано токен доступу, він не надає доступу до API Google Contacts. Однак ми можемо кілька разів надсилати цей токен доступу до API Календаря Google для подібних операцій.

***5. Оновлення токен доступу, якщо це необхідно.*** Токени доступу мають обмежений термін служби. Якщо програмі потрібен доступ до API Google після закінчення терміну дії одного токена доступу, він може отримати токен оновлення. Він дозволяє програмі отримувати нові токени доступу.

Кінцева точка Google OAuth 2.0 підтримує програми, встановлені на таких пристроях, як комп’ютери, мобільні пристрої та планшети. Коли створюється ідентифікатор клієнта через консоль Google API, вказуємо, що це встановлена програма, а потім виберається тип програми Android, додаток Chrome, iOS, універсальна платформа Windows (UWP) або робочий стіл.

Результатом процесу стає ідентифікатор клієнта, а в деяких випадках і секрет клієнта, який вставляється у вихідний код своєї програми.

Your application sends a token request to the Google Authorization Server,
                  receives an authorization code, exchanges the code for a token, and uses the token
                  to call a Google API endpoint.

Рисунок 2.1 –Сценарій OAuth авторизації

Послідовність авторизації починається, коли програма переспрямовує браузер на URL-адресу Google; URL-адреса включає параметри запиту, які вказують на тип доступу, що запитується. Google здійснює автентифікацію користувача, вибір сеансу та згоду користувача. Результатом є код авторизації, який програма може обміняти на токен доступу та токен оновлення.

Додаток повинен зберігати токен оновлення для подальшого використання та використовувати токен доступу для доступу до Google API. Як тільки термін дії токена доступу закінчується, програма використовує токен оновлення для отримання нового.

Обробка токена. Його властивості

Токени можуть відрізнятися за розміром, до наступних меж:

Коди авторизації: 256 байт

Токени доступу: 2048 байт

Оновлення токена: 512 байт

Google залишає за собою право змінювати розмір токена в цих межах, і наша програма повинна підтримувати змінні розміри цих токенів відповідно.

*Оновлення терміну дії маркера*

Потрібно написати свій код, щоб передбачити можливість того, що наданий маркер оновлення може більше не працювати. Токен оновлення може перестати працювати з однієї з таких причин:

Користувач скасував доступ програми.

Токен оновлення не використовувався протягом шести місяців.

Користувач змінив паролі, а токен оновлення містить обсяги Gmail.

Обліковий запис користувача перевищив максимальну кількість наданих (активних) токенів оновлення.

В даний час обмежено 50 токенів оновлення для одного облікового запису користувача на клієнта. Якщо обмеження досягнуто, створення нового маркера оновлення автоматично робить недійсним найстаріший маркер оновлення без попередження. Це обмеження не поширюється на облікові записи служб.

Існує також більший ліміт на загальну кількість токенів оновлення, який може мати обліковий запис користувача або обліковий запис служби для всіх клієнтів. Більшість звичайних користувачів не перевищують цього обмеження, але тестовий обліковий запис розробника може.

Якщо потрібно авторизувати декілька програм, машин або пристроїв, одним із обхідних шляхів є обмеження кількості авторизованих клієнтів для одного облікового запису користувача до 15 або 20.

Використання OAuth 2.0 для клієнтських веб-додатків

У цьому потоці додаток відкриває URL-адресу Google, яка використовує параметри запиту для ідентифікації нашої програми та типу доступу до API, який потрібен додатку. можна відкрити URL-адресу у поточному вікні браузера або у спливаючому вікні. Користувач може пройти автентифікацію в Google і надати потрібні дозволи. Потім Google переспрямовує користувача назад до вашого додатка. Переспрямування включає маркер доступу, який ваш додаток перевіряє, а потім використовує для надсилання запитів API.

Увімкнення API для проекту

Будь-яка програма, яка викликає API Google, повинна вмикати ці API на консолі API.

Відкривається бібліотека API у консолі Google API.

Якщо буде запропоновано, виберається проект або створюється новий.

Бібліотека API перелічує всі доступні API, згруповані за сімейством продуктів та популярністю.

Будь-яка програма, яка використовує OAuth 2.0 для доступу до API Google, повинна мати облікові дані авторизації, які ідентифікують програму на сервері Google OAuth 2.0. Наступні кроки пояснюють, як створити облікові дані для нашого проекту.

API дозволяє програмі запитувати доступ лише до необхідних їй ресурсів, а також дозволяє користувачам контролювати, куди саме допускають нашу програму. Таким чином, може існувати зворотна залежність між кількістю запитуваних областей застосування та ймовірністю отримання згоди користувача.

## 2.4 Відкритий програмний інтерфейс Moodle

Загалом інтерфейс - це межа, через яку взаємодіють дві системи. У Moodle існує багато програмних інтерфейсів, наприклад, наприклад, інтерфейс, який модуль використовує для зв'язку з основним продуктом, або інтерфейс, який використовується для надсилання пошти.

Однак зазвичай цим терміном позначається інтерфейс між програмним забезпеченням та людьми: користувальницький інтерфейс. Наприклад, у Moodle це те, що ми бачимо і натискаємо у веб-браузері та в наших поштових програмах, і воно надає нам способи доступу, розуміння та зміни бази даних в основі будь-якого сайту Moodle.

Moodle використовує рольову модель контролю доступу. Більшість сутностей у Moodle (система, користувачі, категорії курсів, курси, модулі та блоки) представлені контекстами, розташованими у дереві, як ієрархія, що називається контекстним деревом. Роль - це набір визначень можливостей, кожна здатність зазвичай представляє здатність користувача щось робити. Ролі визначаються на найвищому рівні системного контексту. Визначення ролей можна замінити на нижчих рівнях контексту. Контроль доступу користувачів обчислюється з визначень ролей, призначених користувачам.

Усі користувачі, які ще не входили в систему, автоматично отримують роль за замовчуванням, визначену в *$CFG->notloggedinroleid*, неможливо призначити будь-яку іншу роль цьому неіснуючому ідентифікатору користувача. Існує один спеціальний обліковий запис гостьового користувача, який використовується, коли користувач входить в систему за допомогою кнопки гостьового входу або коли ввімкнено гостьовий автоматичний вхід. Знову ж таки, ви не можете призначити будь-які ролі безпосередньо гостьовому рахунку, цей рахунок автоматично отримує *$ CFG->guestroleid.* Усі інші автентифіковані користувачі отримують роль користувача за замовчуванням, зазначену в *$ CFG-> defaultuserroleid*, а в контексті передньої сторінки роль, зазначену в *$ CFG-> defaultfrontpageroleid*.

Moodle здебільшого дотримується підходу до сценарію транзакції. Тобто, у випадку переглядання форуму. URL-адреса буде ... / mod / forum / view.php? Id = 1234. і mod / forum / view.php PHP-скрипт, який генерує цю сторінку. Можна стверджувати, що сценарій транзакції не є відповідним шаблоном для такої складної програми, як Moodle. Однак це дуже природна архітектура для програми PHP, і Moodle - це сукупність багатьох різних плагінів, а не єдиний складний додаток.

За цим базовим підходом до сценарію транзакцій багато основних функціональних можливостей було перероблено на бібліотеки (переважно в папці lib). Це забезпечує елементи моделі домену.

### 2.4.1 Опис корисного функціоналу

*Отримання контексту*

У плагінах екземпляри контексту, як правило, лише створюються, оскільки вони створюються і автоматично видаляються системою.

Отримання за ідентифікатором об'єкта:

$systemcontext = context\_system::instance(); $usercontext = context\_user::instance($user->id); $categorycontext =context\_ coursecat ::instance ($category->id);

$coursecontext = context\_course::instance($course->id); $contextmodule = context\_module::instance($cm->id); $contextblock = context\_block::instance($this->instance->id)

API зовнішніх функцій дозволяє створювати повністю параметризовані методи, до яких можуть отримати доступ зовнішні програми (наприклад, API веб-служб).

Зовнішні функції розташовані під кожним простором імен компонентів \ *external* \ [необов'язково \ *sub* \] Кожна зовнішня функція реалізована всередині окремого класу і доповнена двома функціями опису:

*FUNCTIONNAME\_parameters (),* що описує параметри функцій

*FUNCTIONNAME\_returns (),* що описує повернене значення

Функції опису використовують класи *external\_description*, які були створені для цієї мети.

## 2.5 Відкритий програмний інтерфейс Telegram

Бажано використовувати відкритий вихідний код. До коду включається зразок ідентифікатора API. Цей ідентифікатор API обмежений на стороні сервера і не підходить для програм, що випускаються для кінцевих користувачів – використання його для будь-чого, крім тестування, призведе до помилки API\_ID\_PUBLISHED\_FLOOD для користувачів. Потрібно отримати власний ідентифікатор API перед публікацією програми [11].

Перш ніж використовувати API MTProto Telegram, потрібно звернути увагу, що всі клієнтські бібліотеки API суворо контролюються, щоб запобігти зловживанням.

Якщо і використовується API Telegram для затоплення, спаму, підробки підписки та лічильників переглядів каналів, доступ буде заборонений назавжди.

Через надмірне зловживання API Telegram, усі облікові записи, які реєструються або входять за допомогою неофіційних клієнтів Telegram API, автоматично перебувають під спостереженням, щоб уникнути порушень Умов надання послуг.

Важливо і те, що електронні листи перевіряє людина, тому автоматично генеровані електронні листи будуть виявлені та заборонені.

## Висновки до розділу 2

Пройшовши даний розділ, можна чітко уявити, а також ознайомитися із усіма можливими варіантами отримання даних, зробивши висновок, що потрібно працювати саме з JSON, так як це саме той формат отримання інформації який найбільш поширений серед більшості API, а також в сучасних мовах програмування під нього підібраний гнучкий функціонал для оброблення цих даних.

# 3. РЕАЛІЗАЦІЯ ІНТЕГРАЦІЇ СЕРВІСІВ ДИСТАНЦІЙНОГО НАВЧАННЯ З TELEGRAM

## 3.1 Вибір технологій створення програмного забезпечення

Так як працюємо з backend-розробкою необхідно вибрати середовище у якому власне і будемо працювати.

Node.js (Node) - це платформа розробки з відкритим кодом для виконання коду JavaScript на стороні сервера. Node корисний для розробки програм, які потребують постійного з'єднання браузера з сервером, і часто використовується для додатків у режимі реального часу, таких як чат, стрічки новин та веб-сповіщення [2].

Node.js призначений для роботи на виділеному сервері HTTP та використання одного потоку з одним процесом за раз. Програми Node.js засновані на подіях і працюють асинхронно. Код, побудований на платформі Node, не відповідає традиційній моделі отримання, обробки, надсилання, очікування, отримання. Натомість Node обробляє вхідні запити у постійному стеку подій і надсилає невеликі запити один за одним, не чекаючи відповідей.

### 3.1.1 Cередовище для запуску JavaScript проекту

При роботі з Node.js для пришвидшення, а також спрощення роботи зазвичай підключають фреймворк, щоб розуміти який із можливих варіантів найбільш оптимальний, буде проведено порівняльний аналіз.

Таблиця 3.1 – Порівняння JavaScript фреймворків

|  |  |  |  |
| --- | --- | --- | --- |
|  | Express.JS | Meteor.JS | Koa.JS |
| Переваги | Швидка розробка додатків  Вимагає введення/ виводу  Спільнота з відкритим кодом  Простий у вивченні | Набір пакетів та бібліотек  Інструмент для постійного тестування  Налагодження  Послідовне спілкування клієнт-сервер | Код набагато розумніший  Легкий  Резервне копіювання для синхронізації |
| Недоліки | Кодова організація  Логіка модулів  Зворотні колбеки | Немає візуалізації на стороні сервера  Немає вбудованої підтримки прогресивних веб-програм  Відсутність власної бібліотеки віджетів  Відсутність підтримки MongoDB | Невелика громада для підтримки  Не сумісний з проміжним програмним забезпеченням Node.js |

Проаналізувавши вище згадані фреймворки можна прийти до висновку що найбільш підходящим буде *Express.JS,* так як під нього зроблена достатньо розширена документація. Окрім цього він має велике ком’юніті, тобіш при виникненні проблем з розробкою можна буде швидко знайти оптимальне рішення.

## 3.2 Процедура авторизації OAuth 2.0

При використанні бібліотеки API Google для обробки потоку Oauth 2.0, перший крок - це налаштування об’єктів *gapi.auth2* та *gapi.client*. Ці об’єкти дозволяють нашій програмі отримувати авторизацію користувача та робити авторизовані запити API.

Клієнтський об'єкт визначає сфери дії, на які наша програма вимагає дозволу на доступ. Ці значення повідомляють про згоду, яку Google відображає користувачеві:

* клієнтська бібліотека JavaScript спрощує численні аспекти процесу авторизації;
* вона створює URL-адресу для переадресації для сервера авторизації Google і забезпечує спосіб спрямування користувача на цю URL-адресу;
* обробляє перенаправлення з цього сервера назад у нашу програму;
* перевіряє токен доступу, повернений сервером авторизації.

Окрім цього зберігає токен доступу, який сервер авторизації надсилає нашій програмі, і отримує його, коли ваш додаток згодом робить авторизовані виклики API.

Фрагмент коду нижче - це уривок із повного прикладу, показаного далі в цьому документі. Цей код ініціалізує об'єкт *gapi.client*, який наш додаток згодом використовуватиме для здійснення викликів API. Коли цей об’єкт створюється, об’єкт *gapi.auth2*, який наш додаток використовує для перевірки та контролю стану авторизації користувача, також ініціалізується.

*Виклик* ***gapi.client.init*** *визначає такі поля:*

Значення *apiKey* та *clientId* визначають облікові дані авторизації нашої програми. Ці значення можна отримати в консолі API. *СlientId* необхідний, якщо наша програма робить авторизовані запити API. Додатки, які роблять лише несанкціоновані запити, можуть просто вказати ключ API.

Поле сфери дії визначає обмежений пробілом список областей доступу, які відповідають ресурсам, до яких програма могла отримати доступ від імені користувача. Ці значення повідомляють екран згоди, який Google відображає користувачеві.

Запитуючи доступ до даних користувачів у контексті, шляхом додаткової авторизації, ми допомагаєте користувачам легше зрозуміти, чому нашій програмі потрібен доступ, який вона вимагає.



Рисунок 3.1 – Клієнтський об'єкт

Поле *discoveryDocs* визначає список документів API Discovery, які використовує наша програма. Документ *Discovery* описує поверхню API, включаючи схеми ресурсів, а клієнтська бібліотека JavaScript використовує цю інформацію для генерації методів, які можуть використовувати програми. У цьому прикладі код отримує документ виявлення для версії 3 API Google Drive.

Після завершення виклику *gapi.client.init* код встановлює змінну *GoogleAuth* для ідентифікації об’єкта *Google Auth*. Нарешті, код встановлює слухач, який викликає функцію, коли змінюється статус входу користувача.

### 3.2.1 Перенаправлення на сервер Google OAuth 2.0

Щоб запитати дозвіл на доступ до даних користувача, перенаправляємо користувача на сервер Google OAuth 2.0.

На практиці програма може встановити логічне значення, щоб визначити, чи слід викликати метод *signIn ()* перед спробою здійснити виклик API.

Фрагмент коду нижче демонструє, як ініціюється потік авторизації користувачів. Звертаємо увагу на такі моменти щодо фрагмента:

Об’єкт *GoogleAuth*, на який посилається код, збігається з глобальною змінною, визначеною у фрагменті коду на попередньому кроці.

Функція *updateSigninStatus* - це прослуховувач, який прослуховує зміни стану авторизації користувача.

На практиці програма може встановити логічне значення, щоб визначити, чи слід викликати метод *signIn ()* перед спробою здійснити виклик API.

Фрагмент коду нижче демонструє, як ініціюється потік авторизації користувачів. Звертаємо увагу на такі моменти щодо фрагмента:

Об’єкт *GoogleAuth*, на який посилається код, збігається з глобальною змінною, визначеною у фрагменті вище.

Функція *updateSigninStatus* - це прослуховувач, який прослуховує зміни стану авторизації користувача.



Рисунок 3.2 – Перенаправлення на Google OAuth 2.0

### 3.2.2 Згода користувача в Google на передачу персональних даних

На цьому кроці користувач вирішує, чи надати вашій програмі запитуваний доступ. На цьому етапі Google відображає вікно згоди, яке відображає ім’я нашої програми та сервісів Google API, які вона запитує дозвіл на доступ за допомогою облікових даних авторизації користувача та короткий опис обсягу доступу, який потрібно надати. Потім користувач може дати згоду на надання доступу до одного або декількох обсягів, запитаних нашим додатком, або відхилити запит.

На цьому етапі нашій програмі не потрібно нічого робити, оскільки вона чекає відповіді від сервера OAuth 2.0 від Google із зазначенням, чи був наданий доступ.

### 3.2.3 Оброблення відповіді сервера OAuth 2.0

Клієнтська бібліотека JavaScript обробляє відповідь сервера авторизації Google. Якщо встановити прослуховувач для моніторингу змін поточного стану входу користувача, ця функція викликається, коли користувач надає запитаний доступ до програми.

## 3.3 Отримання доступу до API Google Classroom REST за допомогою Postman

*Postman* - це середовище розробки API, яке допомагає людям створювати, тестувати, документувати, контролювати та публікувати документацію для своїх API.

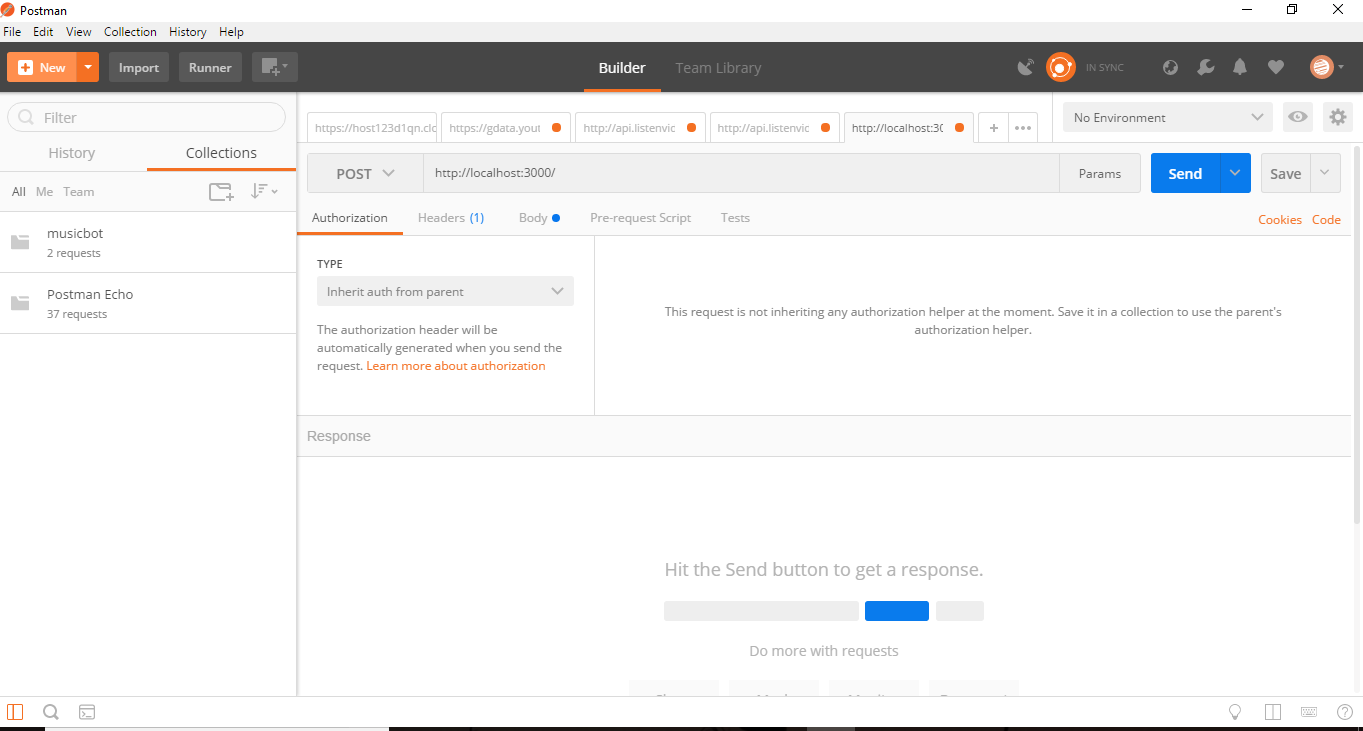


Рисунок 3.3 – Інтерфейс Postman

Основними рисами програмного забезепечення являється - надсилання запитів (з підтримкою різних схем автентифікації, файлів cookie, сертифікатів, заголовків, параметрів запиту, тіла запиту та SOAP із / без WSDL) та налагодження і збереження відповідей.

Для отримання запиту необхідно вписати url адресу з якої власне і буде діставатися інформація а також правильно вказати всі параметри, які додаються в адресі запиту.

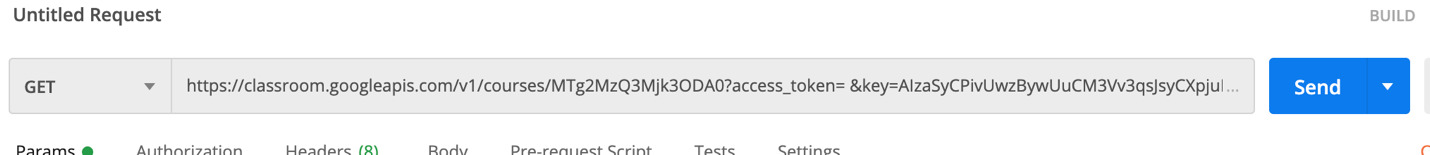


Рисунок 3.4 – URL запиту

Встановлюються для *Body* значення *x-www-form-urlencoded* та надаються наступні значення за допомогою *JSON* або *Credential manger* у консолі:

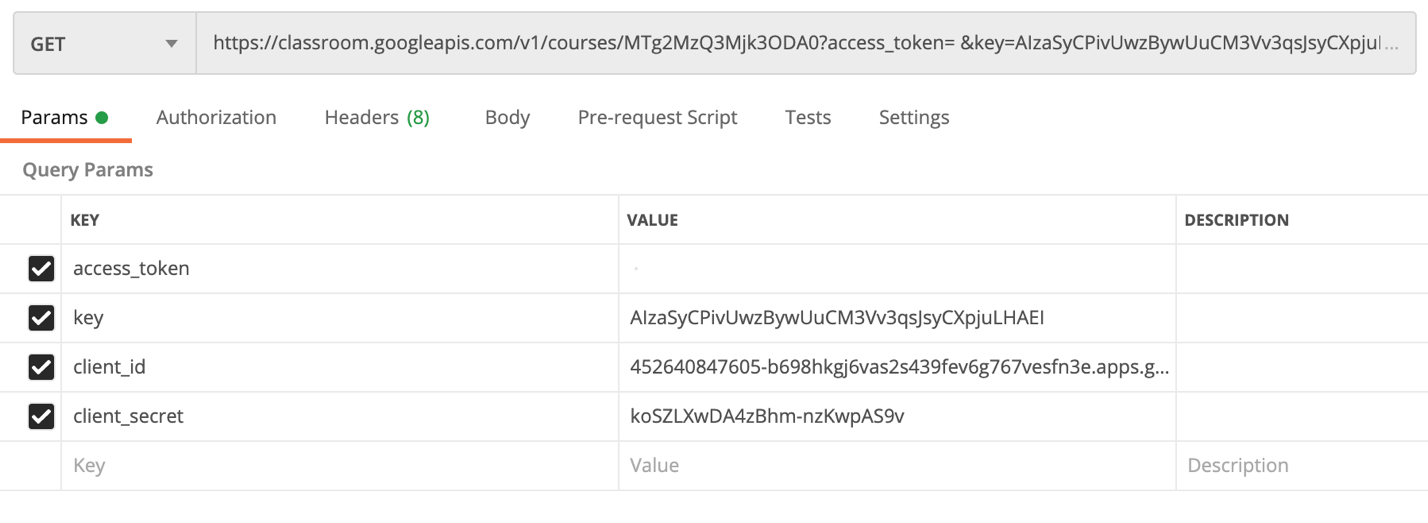
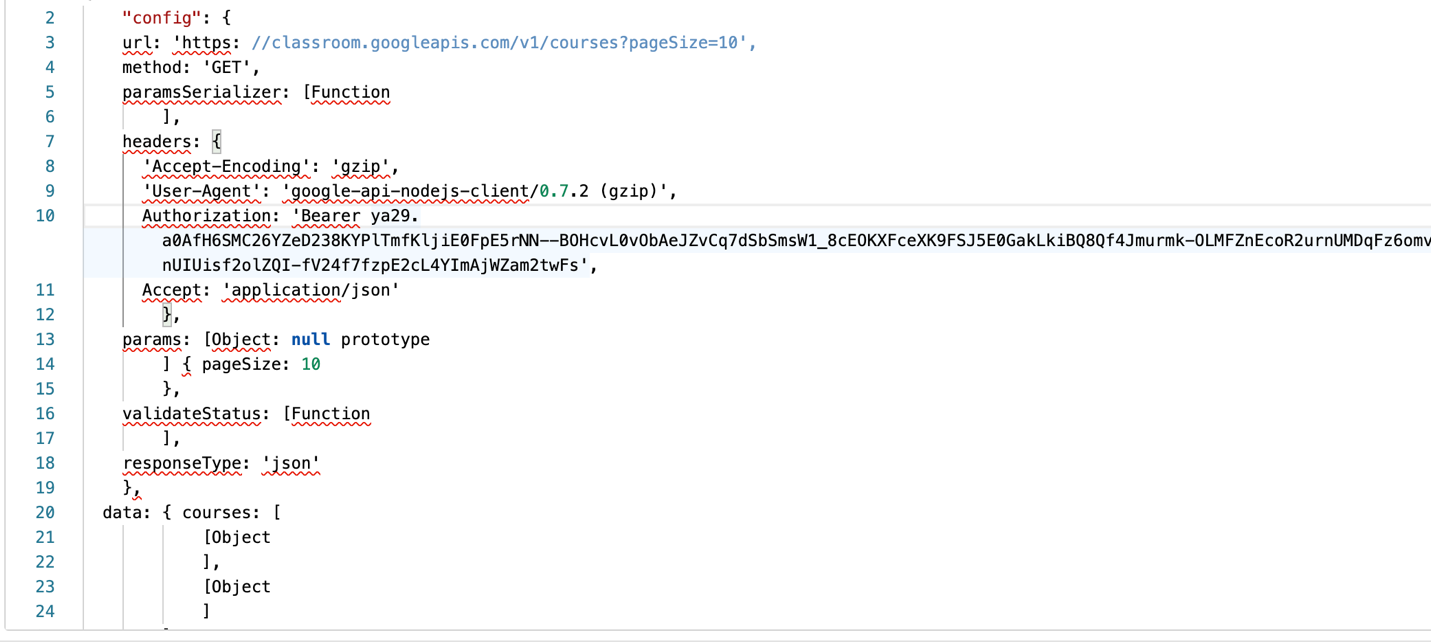


Рисунок 3.5 – Параметри для отримання запиту

Отримавши відповідь можна її переглнути, зрозумівши чи було отримано очікуваний результат



*Робота з частковими ресурсами*

Інший спосіб покращити ефективність викликів API - це запит лише тієї частини даних, яка безпосередньо цікавить. Це дозволяє програмі уникати передачі, аналізу та зберігання непотрібних полів, тому вона може використовувати ресурси, включаючи мережу, центральний процесор, і пам'ять ефективніше.

### 3.3.1 Часткова відповідь

За замовчуванням сервер надсилає назад повне представлення ресурсу після обробки запитів. Для кращої продуктивності можна попросити сервер надіслати лише ті поля, які дійсно потрібні, і замість цього отримати часткову відповідь.

Щоб надіслати запит на часткову відповідь, використовують параметр request поля, щоб вказати поля, які потрібно повернути. Можна використовувати цей параметр із будь-яким запитом, який повертає дані відповіді.

*Приклад:*

У наведеному нижче прикладі показано використання параметра field із загальним (вигаданим) API "Demo".

*Простий запит:* Цей запит HTTP GET опускає параметр fields і повертає повний ресурс.

https://www.googleapis.com/demo/v1

*Повна відповідь ресурсу*: Повні дані про ресурси включають наступні поля, а також багато інших, які були пропущені для стислості.

{  
  "kind": "demo",  
  "items": [  
  {  
    "title": "First title",  
    "comment": "First comment.",  
    "characteristics": {  
      "length": "short",  
      "accuracy": "high",  
      "followers": ["Jo", "Will"],  
    },  
    "status": "active",  
  },  
  {  
    "title": "Second title",  
    "comment": "Second comment.",  
    "characteristics": {  
      "length": "long",  
      "accuracy": "medium"  
      "followers": [ ],  
    },  
    "status": "pending",  
  },  
  ]  
}

Запит на часткову відповідь: Наступний запит на цей самий ресурс використовує параметр fields для значного зменшення обсягу повернутих даних.

https://www.googleapis.com/demo/v1?**fields=kind,items(title,characteristics/length)**

*Часткова відповідь:* У відповідь на запит, наведений вище, сервер надсилає відповідь, що містить лише інформацію про вид, а також масив елементів, що зменшується, що включає лише інформацію про заголовки та довжину в кожному елементі.

{  
  "kind": "demo",  
  "items": [{  
    "title": "First title",  
    "characteristics": {  
      "length": "short"  
    }  
  }, {  
    "title": "Second title",  
    "characteristics": {  
      "length": "long"  
    }  
  },  
  ]  
}

### 3.3.2 Відповідь API на пакетний запит

Відповідь сервера – це одна стандартна відповідь HTTP із багатокомпонентним / змішаним типом вмісту; кожна частина є відповіддю на один із запитів у пакетному запиті в тому ж порядку, що і запити.

Як і частини запиту, кожна частина відповіді містить повну відповідь HTTP, включаючи код стану, заголовки та тіло. І, як і частини в запиті, перед кожною частиною відповіді стоїть заголовок *Content-Type*, який позначає початок частини.

Якщо дана частина запиту мала заголовок *Content-ID*, то відповідна відповідm має заголовок *Content-ID*, з вихідним значенням, перед яким стоїть рядок *response-,* як показано в наступному прикладі.

Наступний приклад демонструє використання пакетного використання з API Classroom

POST https://classroom.googleapis.com/batch HTTP/1.1

Authorization: Bearer your\_auth\_token

Content-Type: multipart/mixed; boundary=batch\_foobarbaz

Content-Length: total\_content\_length

--batch\_foobarbaz

Content-Type: application/http

Content-Transfer-Encoding: binary

MIME-Version: 1.0

Content-ID: <item1:12930812@classroom.example.com>

PATCH /v1/courses/134529639?updateMask=name HTTP/1.1

Content-Type: application/json; charset=UTF-8

Authorization: Bearer your\_auth\_token

{

"name": "Course 1"

}

--batch\_foobarbaz

Content-Type: application/http

Content-Transfer-Encoding: binary

MIME-Version: 1.0

Content-ID: <item2:12930812@classroom.example.com>

PATCH /v1/courses/134529901?updateMask=section HTTP/1.1

Content-Type: application/json; charset=UTF-8

Authorization: Bearer your\_auth\_token

{

"section": "Section 2"

}

--batch\_foobarbaz—

### 3.3.3 Обробка помилок API

API Classroom повертає два рівні інформації про помилки: Об’єкт JSON у тілі відповіді з додатковими деталями

Можна використовувати об’єкт відповіді для налагодження. У деяких випадках відповідь може містити додаткові деталі помилок. Можна використовувати ці деталі, щоб усунути причину помилки та надати користувачам інформацію або направити їх на відповідні дії.

*Структура повідомлення про помилку*

Якщо вони доступні, додаткові деталі помилок включаються в поле повідомлення про помилку, відформатоване як @, а потім код помилки.

Наприклад, якщо запит не вдався з помилкою ClassroomApiDisabled, відповідь JSON буде таким:

{  
  "error": {  
    "code": 403,  
      "message": "@ClassroomApiDisabled The user is not permitted to access the Classroom API.",  
      "errors": [  
        {  
          "message": "@ClassroomApiDisabled The user is not permitted to access the Classroom API.",  
          "domain": "global",  
          "reason": "forbidden"  
        }  
      ],  
      "status": "PERMISSION\_DENIED"  
  }  
}

## 3.4 Отримання даних від API Moodle

Бібліотека яку рекомендують розробники для підключення програмного інтерфейсу https://github.com/llagerlof/MoodleRest:

$url = "http://iot.kpi.ua/lms/webservice/rest/server.php";

$token = "xxx84a93d836f8f2xxx8d97e4ca65bxxx";

//get user\_id by token

$parameters = array();

$site\_info = new \MoodleRest($url, $token);

$site\_info->setReturnFormat(\MoodleRest::RETURN\_ARRAY);

$arr = $site\_info->request('core\_webservice\_get\_site\_info',$parameters);

//var\_dump($arr["userid"]);

//get courses by user\_id

$parameters = array('userid' => $arr["userid"]);

$enrol = new \MoodleRest($url, $token);

$enrol->setReturnFormat(\MoodleRest::RETURN\_ARRAY);

$arr2 = $enrol->request('core\_enrol\_get\_users\_courses',$parameters);

foreach($arr2 as $key => $course){

var\_dump($course ["shortname"]);

echo "<br/>";

}

## 3.5 Створення бота в месенджері Telegram

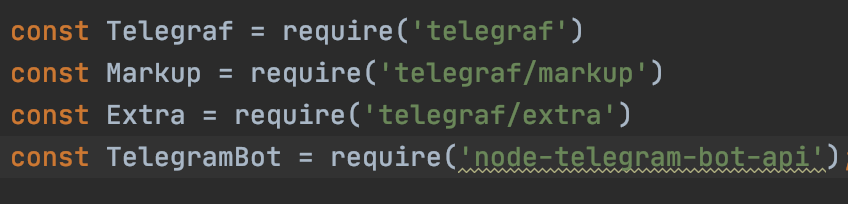
****

Рисунок 3.6 – Модулі для взаємодії з Telegram

Для того, щоб успішно взаємодіяти із користувачами телеграма необхідно підключити певні модулі, які дозволять нам з легкістю оброблювати отримані повідомлення, а також реалізувати функціонал зворотнього зв’язку

Щоб використовувати API Telegram Bot, спочатку потрібно отримати обліковий запис бота, поспілкувавшись з *BotFather* [1].

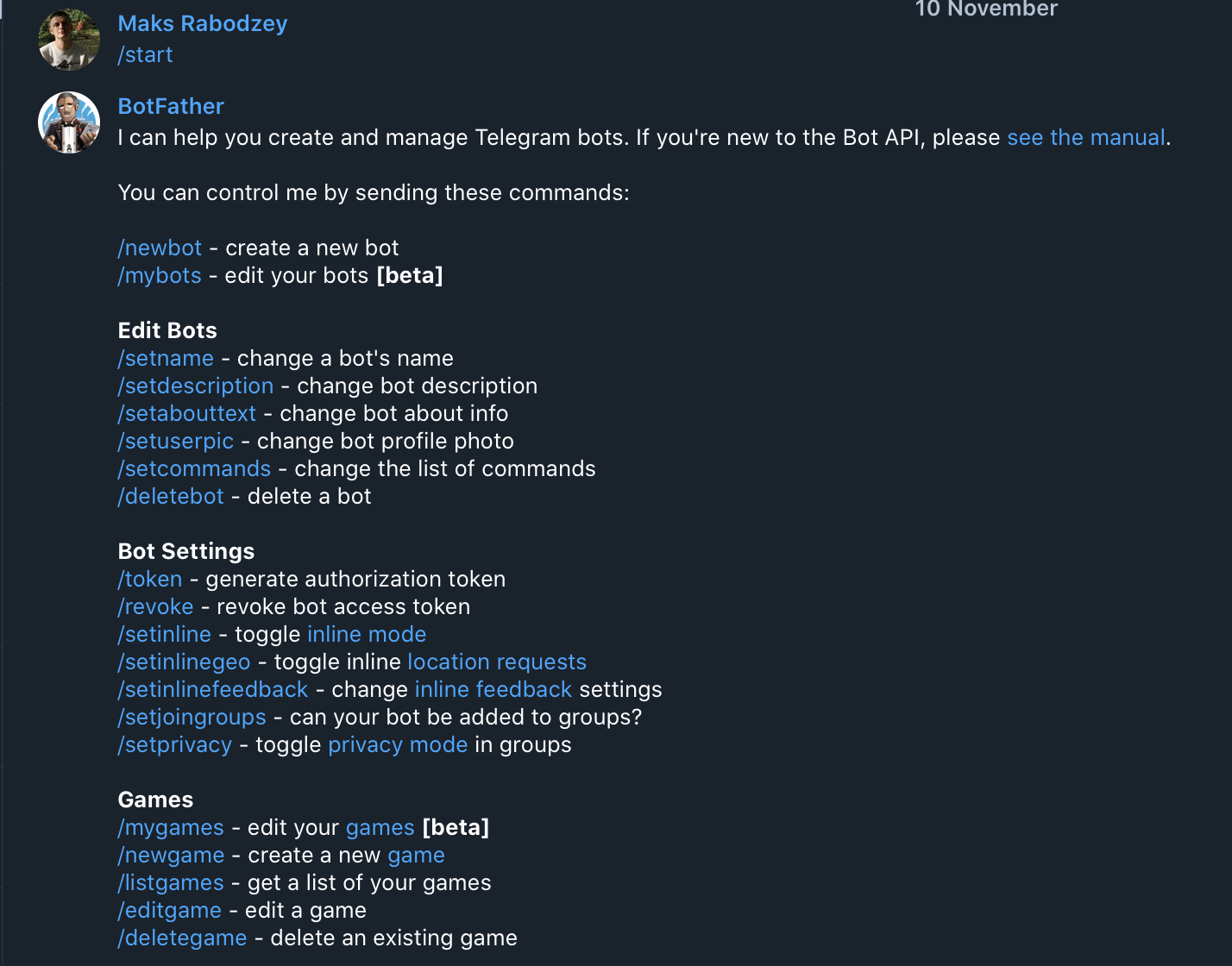


Рисунок 3.7 – BotFather

*BotFather* дасть вам маркер, приблизно на зразок 123456789: AbCdfGhIJKlmNoQQRsTUVwxyZ.

Щоб розпочати взаємодію із ботом вам необхідно прописати команду */start,* після чого і відбудеться початок роботи. Після цього відкриється перелік доступних команд, які запрацюють у відповідності з тим яка команда буде вибрана, або ж прописана вручну.

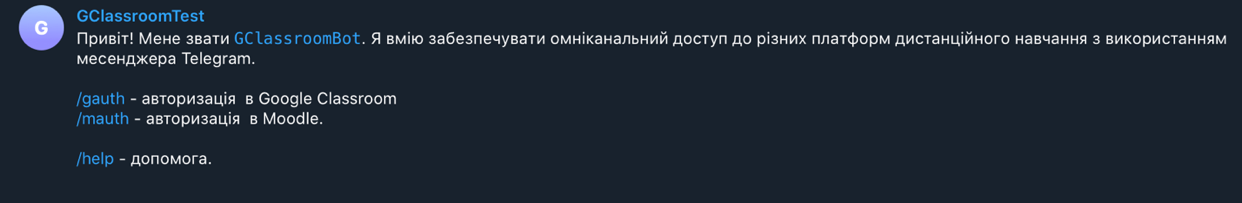


Рисунок 3.8 – Стартові команди бота

Вибравши один із пунктів можна виконати авторизацію в одній із LMS

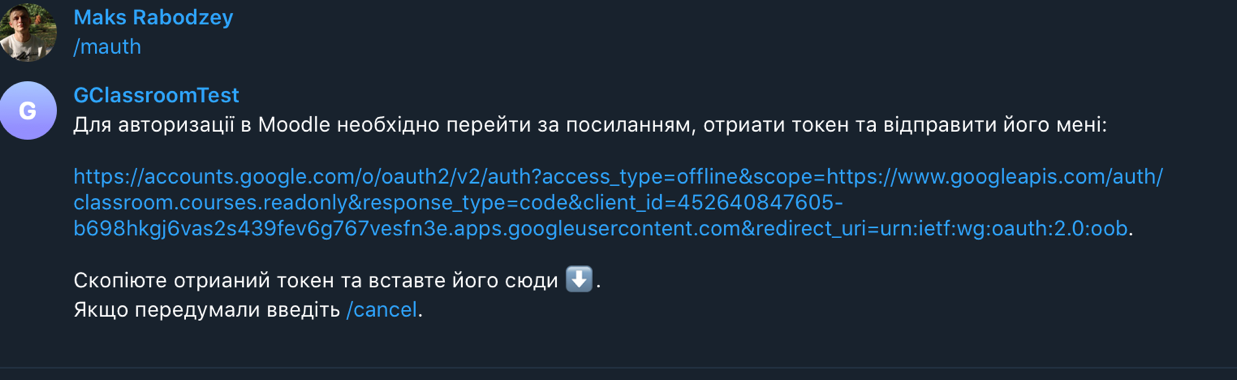


Рисунок 3.9 – Команда на авторизацію

Відкривши меню, стає відкритим додатковий функціонал, зокрема і те, що можна перейти до розділу із курсами

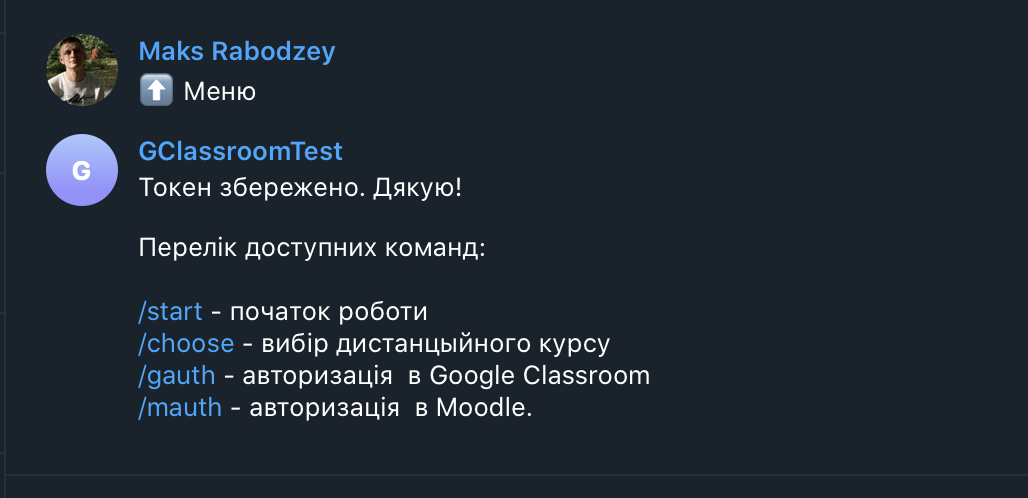


Рисунок 3.10 – Меню бота

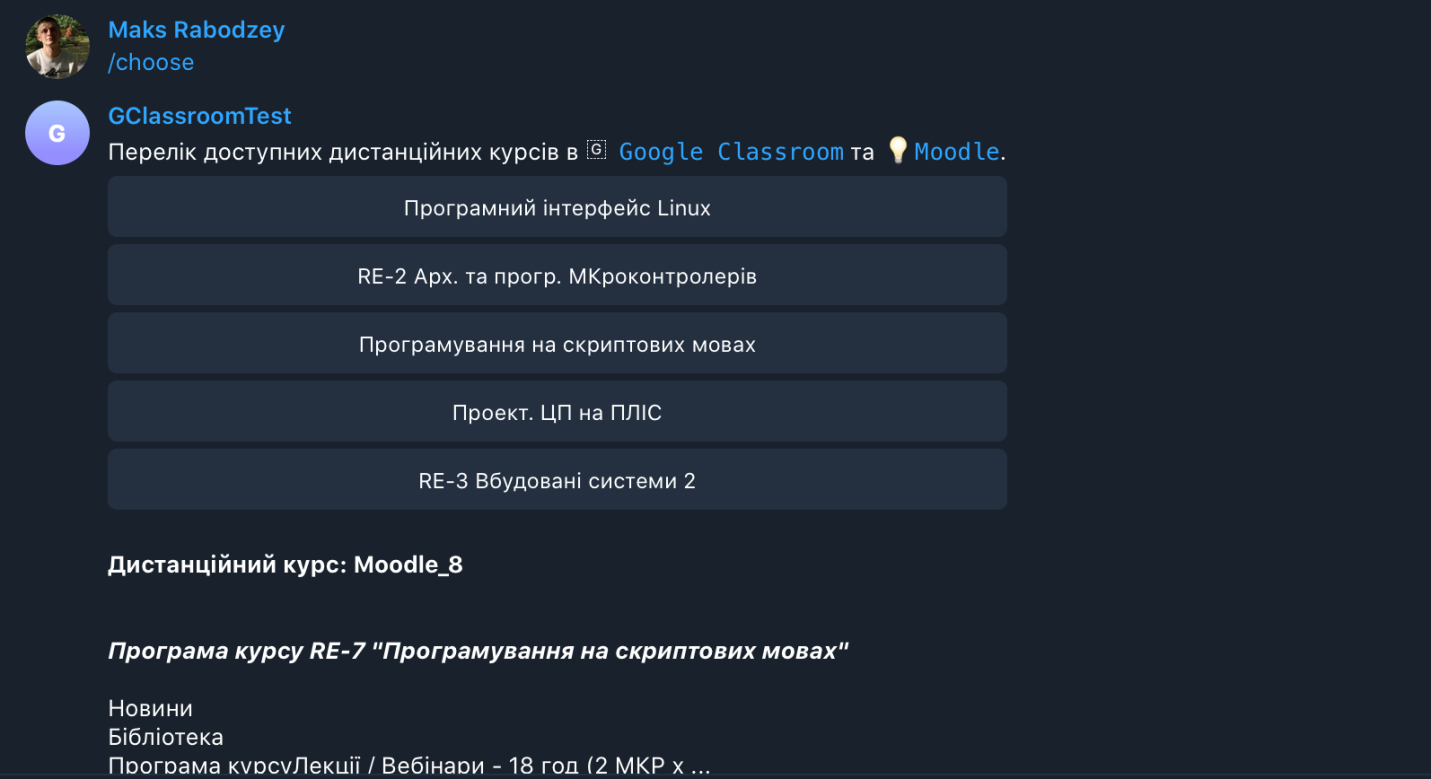


Рисунок 3.11 – Перелік курсів за командою

Бот Telegraf - це об'єкт, що містить масив проміжних програм, які складаються та виконуються у формі стека за запитом. Подібний до багатьох інших систем проміжного програмного забезпечення.

*Проміжне програмне забезпечення*

Проміжне програмне забезпечення є важливою частиною будь-якого сучасного середовища. Це дозволяє нам змінювати запити та відповіді, коли вони проходять між Telegram та нашим ботом [4].

Проміжне програмне забезпечення зазвичай приймає два параметри (ctx, next), *ctx* - це контекст для одного оновлення Telegram, наступне - це функція, яка викликається для виконання проміжного програмного забезпечення, що стоїть нижче. Він повертає *Promise* з функцією *then* для запуску коду після завершення.

*Відправка повідомлення*

bot.action('not bad', (ctx) => {  
 ctx.editMessageText('<span>Have a nice day </span>',  
 Extra.HTML())  
})  
bot.action('all right', (ctx) => {  
 ctx.editMessageText('<i>May happiness be with you 🙏</i>',  
 Extra.HTML())  
})

*Обробка помилок*

За замовчуванням Telegraf надрукує всі помилки в *stderr* і повторно виведе помилку.

Щоб виконати власну логіку обробки помилок, використовуємо такий фрагмент:

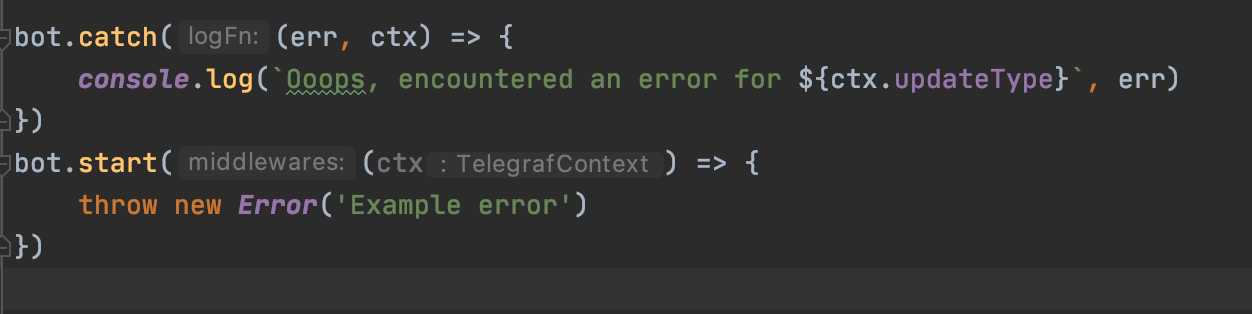


Рисунок 3.12 – Оброблення помилок

*Отримуємо курс за допомогою нашого боту*

Так як у якості сервера в нас виступає власний ПК, виконуємо запуск головного файла

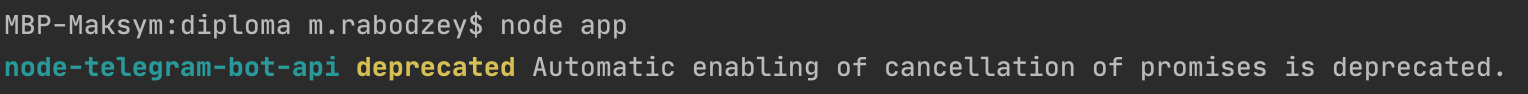


Рисунок 3.13 – Запуск *app.js*

Пишемо боту повідомлення, у відповідь отримуємо посилання.

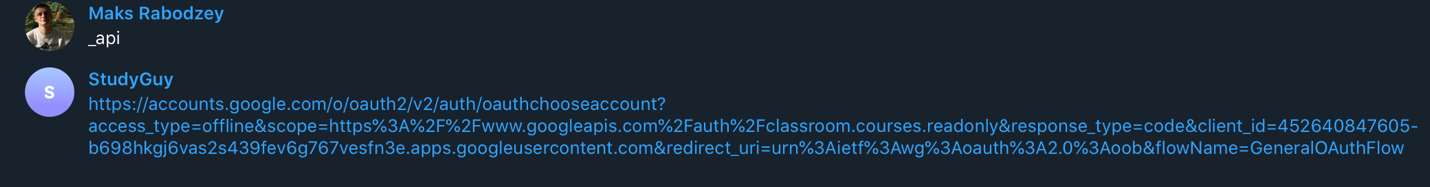


Рисунок 3.14 – Отримання *url* для авторизації

Відкриваємо лінк і проходимо авторизацію

1. Перед вами відкривається список доступних акаунтів, відповідно до цього ми маєте можливість вибрати саме той, що нам потрібен. При цьому важливо не забувати, що під цим акаунтом ми будете отримувати важливі нам дані, тому потрібно бути обережним із вибором

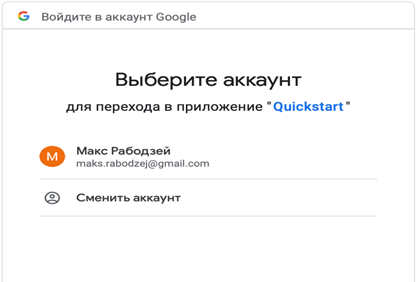


Рисунок 3.15 – Перший крок авторизації

Надаємо доступ до сервісу *Google Classroom*

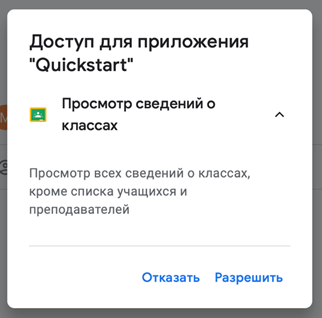


Рисунок 3.16 – Другий крок авторизації

1. На цьому етапі необхідно ознайомитися, що ми надаємо доступ до перегляду інформації, яка безпосередньо пов’язана з нашими класами, а це значить, що Google буде володіти цими данними

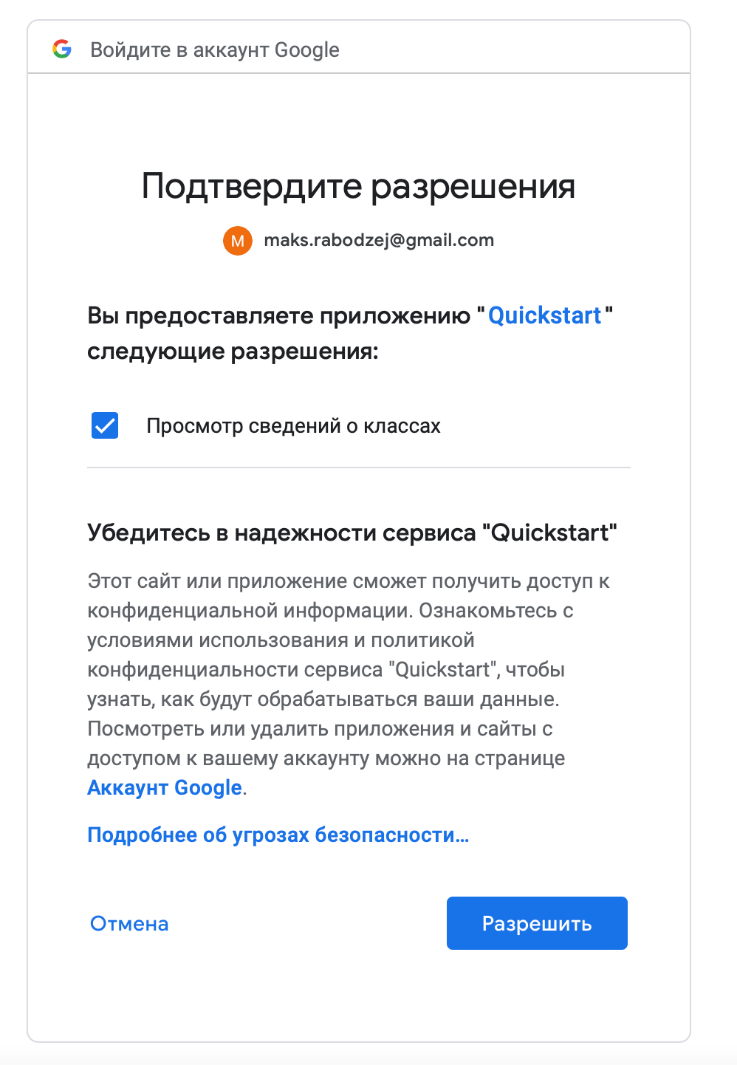


Рисунок 3.17 – Третій крок авторизації

В завершення отримуємо токен, у якому закладаються всі ключі доступа, які необхідні для подальшої роботи

*4/1AY0e-g41ZZC3eZ-oULO1ijdm7urt7uak0OxXLYSzV7slnCfXuepMx-jdaFo*

З його допомогою виконуємо вхід:

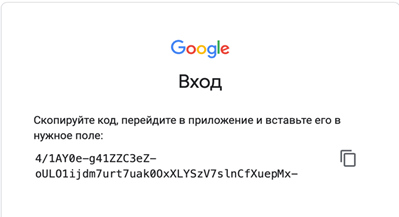


Рисунок 3.18 – Отримали токен

1. Тепер можемо отримати наприклад назву курса



Рисунок 3.19 – Відповідь на запит

Щоб розмістити завдання курсу та оцінки у певній темі необхідно до неї звернутися. Щоб отримати список тем та відповідні ідентифікатори, необхідно виконати наступний фрагмент коду.

## 3.6 Функціонал програмного забезпечення дистанційного навчання

*Список отриманих топіків*

function getTopicList(){  
 const courseID = "83743632546";  
 console.log(getTopics(courseID));

*Отримання топів по даному ID*

function getTopics(COURSE\_ID){  
 const topics = Classroom.Courses.Topics.list(COURSE\_ID).topic;  
 const topicItems = topics.map(item => `${item.name} : ${item.topicId}`);  
 return topicItems;  
};

Коли викликається *getTopicList()* можна буде спостерігати ось такі результати:

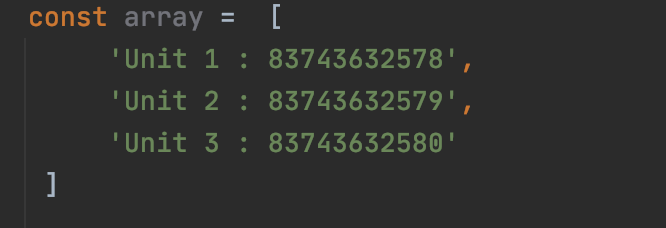


Рисунок 3.20 – Приклад отриманого результату

*Створення курсового завдання.* Для створення курсового проекту в Google Classroom, можна прописати наступну механіку:

*Приклад отриманого результату*

function createCourseWorkItem(){  
 const COURSE\_ID = "83743632546";// <<< UPDATE THIS  
 const TOPIC\_ID = "83743632579";// <<< UPDATE THIS  
 const COURSE\_INFO = {  
 "assigneeMode": "ALL\_STUDENTS",  
 "associatedWithDeveloper": true,  
 "description": "describe what your assignment is about here.",// <<< UPDATE THIS  
 "maxPoints": 40, // <<< UPDATE THIS  
 "state": "PUBLISHED",  
 "submissionModificationMode": "SUBMISSION\_MODIFICATION\_MODE\_UNSPECIFIED",  
 "title": "Unit 2 - External Grade import", // <<< UPDATE THIS  
 "workType": "ASSIGNMENT",  
 "topicId":TOPIC\_ID  
 }  
 const COURSEWORK\_ID = createCourseWork\_(COURSE\_ID,COURSE\_INFO);  
 console.log(COURSEWORK\_ID)  
}



Рисунок 3.21 – Перезапис курсу

При необхідності можна дописати функціонал, який дозволить отримувати більшу кількість інформації за категоріями. Наприклад, додаткове інфо по курсу або ж розклад і тому подібне.

## Висновки до розділу 3

У цьому розділі реалізовано програмне забезпечення для дистанційного навчання, застосувавши при цьому ресурси  Google Classroom API, а також Телеграм API. При цьому було проведено тестове отримання запитів, що і показала працездатність отримання даних. Також реалізовано сам бот через який і проходить весь функціонал.

# 4. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

Розділ має на меті проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Етапи маркетингового аналізу стартап-проекту:

## 4.1 Опис ідеї проекту (товару, послуги, технології)

Таблиця 4.1 – Опис ідеї стартап-проекту

|  |  |  |
| --- | --- | --- |
| Зміст ідеї | Напрямки застосування | Вигоди для користувача користувача |
| Створення чат-боту за допомогою Telegram API | 1. Навчальний | Мобільність використання |
| 1. Комерційний |

1.2) Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів.

## 4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

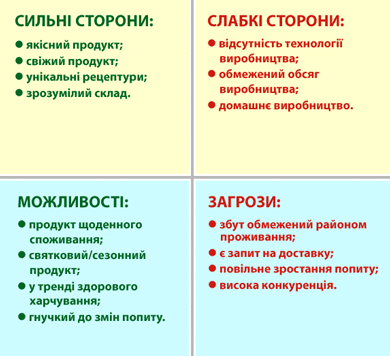


Рисунок 4.1 – Сильні та слабкі сторони проекту

## 4.3 Аналіз ринкових можливостей запуску стартап-проекту

3.1) Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.2).

Таблиця 4.2 – Попередня характеристика потенційного ринку

|  |  |
| --- | --- |
| Показники стану ринку (найменування) | Характеристика |
| Кількість головних гравців, од. | 0 |
| Загальний обсяг продаж, грн/ум.од | - |
| Динаміка ринку (якісна оцінка) | Існує попит |
| Наявність обмежень для входу | Значні інвестиції |
| Специфічні вимоги до стандартизації та сертифікації | Так |
| Середня норма рентабельності в галузі (або по ринку), % | - |

Ринок є привабливим для входження за попереднім оцінюванням.

3.2) Надалі визначаються потенційні групи клієнтів, характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 4.3).

Таблиця 4.3 – Характеристика потенційних клієнтів стартап-проекту

|  |  |  |  |
| --- | --- | --- | --- |
| Потреба, що формує ринок | Цільова аудиторія | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
| Програмне забезпечення дистанційного навчання з використанням месенджерів | МОН | Невідомі | Надійсність, компактність |

3.3) Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: (табл. № 4.4-4.5).

Таблиця 4.4 – Фактори загроз

|  |  |  |
| --- | --- | --- |
| Фактор | Зміст загрози | Можлива реакція компанії |
| Появлення вигіднішого конкуренту | Відсутність продаж | Вдосконалення функціоналу |

Таблиця 4.5. – Фактори можливостей

|  |  |  |
| --- | --- | --- |
| Фактор | Зміст можливості | Можлива реакція компанії |
| Поява нової технології | Простіше виготовляти | Переобладнання виробництва |

3.4) Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку

За результатами аналізу таблиці продукт можна випускати на ринок, не зважаючи на існування конкурентів, але необхідно забезпечити запуск виробництва протягом року та з низькою ціною продажу.

3.6) На основі аналізу конкуренції, визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 5.10

Таблиця 4.6 – Обґрунтування факторів конкурентоспроможності

|  |  |
| --- | --- |
| Фактор конкурентоспроможності | Обґрунтування |
| Ціна | Затрати на матеріали та виготовлення |
| Простота | Технологія виготовлення |
| Надійсність | Технологія виготовлення |

1. 7) Фінальним етапом ринкового аналізу можливостей впровадження проекту є виконання SWOT-аналізу

## 4.4 Розроблення ринкової стратегії проекту

1) Розроблення ринкової стратегії першим кроком передбачає ви- значення стратегії охоплення ринку: опис цільових груп потенційних споживачів

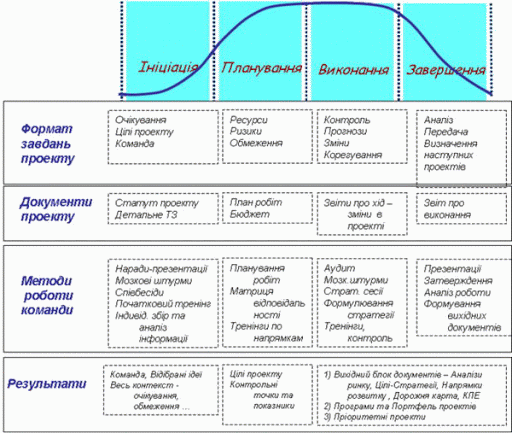


Рисунок 4.1 – Створення стратегії проекту

2) Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.7).

Таблиця 4.7 – Визначення базової стратегії розвитку

|  |  |  |
| --- | --- | --- |
| Стратегія охоплення ринку | Ключові конкурентос-  проможні позиції | Базова стратегія розвитку |
| Диференційований маркетинг | Ціна, простота, надійність | Стратегія спеціалізації |

3) Наступним кроком є вибір стратегії конкурентної поведінки

## 4.5 Розроблення маркетингової програми стартап-проекту

1) Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.8 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.8. Визначення ключових переваг концепції потенційного товару

|  |  |  |
| --- | --- | --- |
| Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити |
| Програмне забезпечення дистанційного навчання з використанням месенджерів | Виграш у ціні та простоті | Простота і ціна |

2) Надалі розробляється трирівнева маркетингова модель товару: (табл. 4.9).

Таблиця 4.9 – Опис трьох рівнів моделі товару

|  |  |
| --- | --- |
| Рівні товару | Сутність та складові |
| І. Товар за задумом | Програмне забезпечення дистанційного навчання з використанням месенджерів |
| ІІ. Товар у реальному виконанні | Властивості |
| 1. Ціна  2. Простота |
| Якість: тестування макету |
| Пакування: відсутнє |
| Марка: відсутня |
| ІІІ. Товар із підкріпленням | До продажу: невідомо |
| Після продажу: невідомо |
| Потенційний товар буде захищено від копіювання через патент | |

3) Наступним кроком є визначення цінових меж (табл. 4.10). Аналіз проводиться експертним методом.

Таблиця 4.10 – Визначення меж встановлення ціни

|  |  |  |  |
| --- | --- | --- | --- |
| Рівень цін на товари замінники | Рівень цін на товари аналоги | Рівень доходів цільової групи споживачів | Верхня та нижня межі встановлення ціни на товар/послугу |
| 70-100 тис. ум. од. | До 5 тис. ум. од. | високий | Попитом |

4) Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.11).

Таблиця 4.11 – Формування системи збуту

|  |  |  |  |
| --- | --- | --- | --- |
| Специфіка закупівельної поведінки цільових клієнтів | Функції збуту, які має виконувати постачальник товару | Глибина каналу збуту | Оптимальна система збуту |
| Невідома | Вільний доступ до товару | невідома | Вільний доступ до товару |

4.5) Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 5.19).

Таблиця 4.12. – Концепція маркетингових комунікацій

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові пози ції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
| Невідома | Інтернет, наукові публікації, прес-служби | Можливості проекту | Розповісти про можливості проекту | Переконання сильними сторонами проекту |

## Висновки до розділу 4

Існує можливість ринкової комерціалізації проекту, адже на ринку є запит на вирішення проблеми диверсифікованих каналів дистанційного навчання. Є перспективи впровадження програмного забезпечення для дистанційного навчання через месенджери з огляду на потенційні групи клієнтів, бар’єри входження, стан конкуренції, конкурентоспроможність проекту. Подальша імплементація проекту можлива, при знаходженні інвесторів.

# ЗАГАЛЬНІ ВИСНОВКИ

В даній магістерській дисертації проведено дослідження особливостей застосування чат ботів месенджерів в якості інтерфейсу взаємодії учня та студентів з різними навчальними системами та сервісами, зокрема інтеграції чат ботів месенджерів з системами керування навчальним вмістом (LMS) для забезпечення вирішення завдань дистанційного навчання.

Проведено аналіз сучасних чат-ботів та встановлено, що серед учнів та студентів найбільш перспективним є застосування месенджера Telegram внаслідок його популярності та багатофункціонального відкритого програмного інтерфейсу (API).

Розглянуто можливості та формати даних для автоматизованого обміну інформацією між розподіленими інформаційними системами, зокрема XML та JSON., формування запитів та оброблення відповідей. Розглянуто питання тестування відкритих прграмних інтерфейсів перед їх використанням в програмному забезпеченні.

Для демонстрації можливостей інтеграції різнопланових інформаційних сервісів та систем реалізовано власний Telegram Bot з авторизацією в Google API Client та підключенням до LMS Moodle, реалізовано базовий інтерфейс взаємодії бота з користувачем.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Телеграм API [Електронний ресурс]. — <http://core.telegram.org/bots>. – Назва з екрану

2. Node.JS Frameworks [Електронний ресурс].https://habr.com/ru/post/  
486886. **—** Назва з екрану.

3. Google API [Електронний ресурс]. —https://developers.google.com/  
identity/sign-in/android/sign-in. — Назва з екрану.

4. Телеграф API [Електронний ресурс]. — https://telegraf.js.org/#/ — назва з екрану

5. Telegram Bots [Електронний ресурс]. [https://www.bitdegree.org/  
tutorials/telegram-bots/#:~:text=By%20using%20bots%2C%20people%20  
can,into%20your%20chat%20or%20channel](https://www.bitdegree.org/tutorials/telegram-bots/#:~:text=By%20using%20bots%2C%20people%20can,into%20your%20chat%20or%20channel). — Назва з екрану

6. What is Moodle [Електронний ресурс] [https://ethinkeducation.com/  
what-is-moodle-guide](https://ethinkeducation.com/what-is-moodle-guide). — Назва з екрану.

7. XML (Extensible Markup Language) [Електронний ресурс]. <https://whatis.techtarget.com/definition/XML-Extensible-Markup-Language> – Назва з екрану.

8. JSON: What It Is, How It Works, & How to Use It [Електронний ресурс]. <https://www.copterlabs.com/json-what-it-is-how-it-works-how-to-use-it>. – Назва з екрану.

9. Introducing Bot API 2.0 [Електронний ресурс]. <https://core.telegram.org/bots/2-0-intro/>. — Назва з екрану

10. Virtual classroom [Електронний ресурс]. [https://opensource.com/  
article/20/4/open-source-remote-teaching-tools](https://opensource.com/article/20/4/open-source-remote-teaching-tools). — Назва з екрану.

11. Getting started with Google Classroom [Електронний ресурс]. <https://edu.gcfglobal.org/en/google-classroom/getting-started-with-google-classroom/1/> – Назва з екрану.

