2024 International Conference on Intelligent and Innovative Computing Applications (ICONIC)
7th-9th November 2024, Pearle Beach Resort & Spa, Flic en Flac, Mauritius
https://doi.org/10.59200/ICONIC.2024.009

Enhancing Online Class Engagement through Generative AI: A Research using Microsoft Graph and Gemini APIs

Makhulu Relebogile Langa and Michael Nthabiseng Moeti
Tshwane University of Technology
Polokwane, South Africa
relebogilelanga@gmail.com, moetimn@tut.ac.za

Abstract
The education landscape has undergone a significant transformation, with online learning emerging as the preferred mode of instruction for many students. Remote learning required a variety of digital tools and technologies that not all students and teachers could use. As some students struggled to adjust to an unfamiliar environment, attendance and engagement varied. Various methodologies were employed to enhance student involvement, classified into the domains of Virtual Reality, Augmented Reality, Internet of Things, and Artificial Intelligence (AI). The application of AI in this particular area demonstrated encouraging outcomes compared to other domains. Therefore, this paper incorporates Generative AI to enhance the existing AI literature. Multiple APIs were utilized, and their configuration procedures were thoroughly described. The findings in this paper show that the utilization of Microsoft Graph and Gemini APIs facilitated the gathering of transcripts/recordings and the creation of questions based on the content that was presented. The Streamlit web application employs these questions and presents them to students in order to evaluate their degree of engagement. The aim of this paper is to propose a solution for fostering engagement in online classes by encouraging students to consistently interact with the presented content through generated questions.

Keywords: Online learning, Student engagement and participation, Generative AI, Transcripts, Question generation.

1. Introduction

The substantial growth rates of online enrolments in comparison to overall higher education enrolments are indicative of the growing demand for online learning (Lee et al., 2021). There has been a greater increase in the number of students enrolling in one or more online courses than in the overall student population of higher education institutions (Adedoyin and Soykan, 2020) (Bashir et al., 2021). In addition, the COVID-19 pandemic has accelerated the adoption of online learning, as educational institutions were compelled to move to remote instruction to maintain the continuity of education during the global health crisis (Coman et al., 2020) (Bashir et al., 2021). The transition to remote learning required the implementation of a variety of digital tools and technologies, which not all students and teachers were equally prepared to navigate. However, online learning has been instrumental in expanding educational opportunities and access for a diverse student body, despite these obstacles (Hafeez et al., 2022). With the ongoing evolution of online learning, there is an increasing focus on issues such as students' motivation, involvement, and attitude towards online learning (Mahani, 2023). This led to inconsistencies in attendance and participation, as certain students encountered difficulties in adjusting to the new learning environment (Mahani, 2023). These difficulties include:

- Digital competence (Ratni, 2024)

This refers to inabilities to navigate online platforms and effectively using digital tools.

- Pedagogical adaptation (Lei et al., 2021)

This refers to facilitators not adjusting their pedagogical practices for online environments in consideration of student engagement, content delivery, and assessment methods.

- Student motivation and engagement (Tiwow et al., 2023)

This includes inability to sustain student motivation and engagement in digital learning environments, thus, leading to poor communication and interaction with lecture content presented.

The higher education sector acknowledges the significance of improving student engagement in online learning environments as it shifts towards a more digitally focused learning landscape (Ferrer et al, 2020) (Abdallah et al., 2020). Addressing the challenges of adapting to digital environments requires a comprehensive approach that considers the diverse needs of educators and learners in digital learning settings, which this paper attempts to provide. The incorporation of technology-driven strategies has become a widespread method in educational environments to improve student attendance and involvement. Some of these approaches are described below in Table 1:

Table 1. Approaches currently deployed to improve student engagement in online lessons.

Approach	**Description**
Online polling platforms (Malekjafarian, 2024)	These systems facilitate instructors in collecting instantaneous feedback from students, promoting active engagement and motivating learners to remain engaged throughout the learning process.
Multimedia integration (Zimu, 2024)	The incorporation of multimedia presentations, internet conversations, and virtual simulations in technology-Improved learning environments can promote student engagement by increasing motivation and interest in the subject matter.
Active learning and interactive features (Azizan, 2023)	The implementation of interactive elements and active learning approaches has been proven to enhance cognitive involvement in online education.
Online course design (Tualaulelei et al., 2021)	The strategic planning of the systematic design of online courses has also been emphasized as a method to optimize student engagement and learning outcomes.

Numerous innovative strategies have been investigated in the field of technology-based strategies to improve student engagement in a variety of domains, including AI, IoT, and other comparable technologies (Almusaed et al., 2023). The integration of context-based teaching methods with innovative technology has been recommended as a means of fostering sustainability in the learning process and effectively engaging students (Huda et al.,2022). These strategies emphasise using innovative technology to create immersive, interactive, and personalised learning experiences for students in diverse educational sectors. However, such strategies have drawbacks like poor online class attendance, lack of attention and engagement, distractions. These challenges form theoretical foundations of engagement in online learning environments, and we utilize generative artificial intelligence to supplement the existing works in improving student engagement in online learning environments to positively impact student attendance rates, which ultimately lead to improved success rates.

GenAI is a specialised field within artificial intelligence that specifically aims to generate new content that closely resembles that of humans (Stokel-Walker and Noorden, 2023). GenAI models are trained on massive amounts of data to analyse patterns and relationships to generate outputs similar to the training data. The purpose of this paper is to suggest a method for enhancing online class engagement through GenAI using critical tools such as Microsoft Graph API for collecting transcripts and Gemini API to generate questions based on the students' learning style and transcripts.

This method aims to assess students' level of involvement during an online class in order to determine whether a student is present and actively participating rather than simply being present. To achieve this, at the conclusion of the session, after recording and transcription have ceased, students are given a survey consisting of closed-ended questions related to the lesson content. This survey is administered for a limited time to assess the students' level of engagement during the session. By utilising the attendance register, assessors and lecturers can accurately determine which students were actively engaged in the class and distinguish them from those who were merely physically present. Implementing this measure will guarantee that students remain attentive at all times, thereby actively participating in the lecture and interacting with the presented material.

In the next section, the related works pertaining to the use of various technologies and tools to improve attitudes and participation in online learning environments are discussed. Section 3 describes the methodology used to develop the proposed solution for enhancing students' engagement in online learning environments. Section 4 describes the experimentation of the proposed solution and the associated results. In Section 5, the results from the experiments are evaluated to determine whether the proposed solution can identify actively participating students and simply present students. Lastly, Section 6 concludes the paper, with recommendations and future work.

2. Related Works

This section discusses students' participation in online learning environments and how different technologies and tools are applied to promote engagement and in turn promote attendance rates. In a quasi-experimental study by Li et al. (2019), feedback design was investigated as to how it might impact student's motivation in participating in online learning environments. This suggests that students are encouraged to actively engage in online learning environments due to the timely feedback they receive (Meşe and Sevilen, 2021). Therefore, we investigate the works that emphasise methods and technologies that encourage interactive learning to improve engagement in online classes in addition to those identified in Section 1.

Walker (2022) explored the use of Virtual Reality (VR) technology to create interactive content for students. Virtual Reality (VR) is particularly effective in its capacity to vividly represent abstract ideas, enabling users to visualise and interact with complex scientific phenomena, historical events, and artistic movements. This immersive experience enhances understanding and facilitates a more thorough comprehension.

In an Internet of Things (IoT) approach, Abuatiq et al. (2022) suggested that Smartwatches and fitness trackers can be used with educational software to make learning fun. Learning becomes more exciting and dynamic when students track their progress, compete in learning challenges, and receive real-time feedback.

Kurniawan (2024) proposed an Augmented Reality (AR) strategy which offers a unique way to boost student engagement in online learning environments by overlaying digital elements onto the real world a student inhabits. AR can change textbooks and learning materials during online classes to allow students to enhance learning by scanning pages or objects to activate animations, 3D models, or more information.

Sadler et al., (2024) mentioned that the use of AI tools can enhance student interaction and engagement in online learning environments, facilitating personalized learning experiences and supporting self-regulated learning behaviours.
The incorporation of AI to promote engagement in online learning environments is currently the most suitable choice as it is currently widely accessible and recent as compared to VR, AR, and IoT solutions (Stecyck and Miciuła, 2024). This is because AI may enhance student engagement by tailoring content and learning paths to individual needs, providing immediate feedback, and enabling interactive simulations of complex systems. In addition, AI can also provide data-driven insights and predictive models to help educators understand student behaviour and learning patterns and tailor interventions to boost engagement (Ouyang et al., 2023).

This section reviewed the works of VR, AR, IoT, and AI. These technologies, except the latter, have shortcomings of requiring external devices and AR applications. Moreover, they are expensive to implement and adopt and might not be universally available to all students. However, the latter, AI, on the other hand is preferred for this domain of enhancing student engagement due its profound benefits such as tailoring content, personalizing learning, providing real-time feedback. As such, this paper utilizes GenAI, which plays a significant role in promoting student engagement in online classes. Moreover, it is cheaper to implement, and can be easily made accessible to the students and lecturers.

3. Methodology

Student engagement in an online course involves actively participating in class activities, interacting with instructors and classmates, completing assignments, and using e-learning platforms (Xie and Correia, 2023). Studies have shown that enhancing student interest can result in greater involvement and overall contentment, influencing academic success (Suriagiri et al., 2022). Therefore, this section presents mechanisms and tools that can promote student engagement.

We illustrate a flow diagram in Figure 1 that can guide to understand the internal processes of the proposed solution.

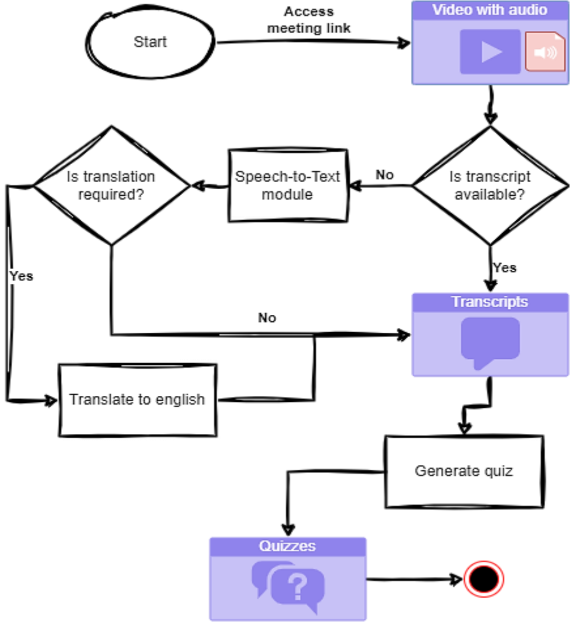


Figure 1. A flow of the proposed solution.

Below is a breakdown of these processes in categories of data preparation and collection, data cleaning, quiz generation, and quiz evaluation, for training the GenAI model prior to delivering the proposed solution.

3.1. Data preparation and collection

In this category, Microsoft Teams transcripts are collected from OneDrive using GraphAPI. GraphAPI is a tool by Microsoft that provides access to data stored across Microsoft 365 services such as OneDrive, OneNote, SharePoint, Teams, and the likes (Mishra, 2022). Please note that this approach assumes that Teams was configured to automatically save transcripts in a OneDrive folder named *Recordings*. In a Python project, we use Microsoft Authentication Library (***msal)*** to send requests to the GraphAPI. Utilizing the GraphAPI, we follow steps in Table 2.

Table 2. Steps illustrating acquisition of transcripts using Python and ***msal***.

Steps	**Description**
Authorization and Permissions	We set up GraphAPI access with proper authentication by registering an application in Azure Active Directory (AAD) and obtaining an access token that grants permission to access the user's OneDrive files.
Identify the transcript folder	Having ensured that Teams saves transcripts automatically in a OneDrive folder, we locate that folder and we feed the location to the GraphAPI.
Access files	We then use GraphAPI to access the transcripts files and download them to a Google Drive for further processing.

In cases where transcription was not enabled, a recording of the session is obtained using GraphAPI on OneDrive and fed to a Speech-to-Text Python module called MoviePy, which is designed for video editing (Kamalova, 2024). It can be employed to accurately perform basic operations, video compositing, video processing, or to generate transcription, and advanced effects (Priya et al., 2022). We then save the extracted transcripts to the location of where Teams transcripts would be saved on OneDrive for them to be easily downloaded for further processing.

3.2. Data processing and cleaning

With the transcripts downloaded as Microsoft Word documents (docx), we utilize *pandas* Python library to read the transcripts to extract information such as timestamps, conversation content, and lesson content which will be used in quiz generation. Pandas is a Python library specifically designed for manipulating and analysing data sets. The software possesses capabilities for analysing, purifying, investigating, and manipulating data (Stepanek, 2020).
Using the famous Natural Language Toolkit (NLTK), we define a stop word list, which will assist us in remove pause words in the transcript data. NTLK is a collection of libraries and programmes for symbolic and statistical natural language processing (NLP) in English. It provides semantic reasoning, parsing, tokenization, stemming, tagging, and classification capabilities (Wang and Hu, 2021).

We further apply the following cleaning steps:

- Standardize punctuation and formatting:

We use string manipulation techniques to remove extra spaces, convert new lines to a consistent format, and ensure consistent punctuation usage.

- Standardize speaker tags:

We identify patterns in speaker tags and use string manipulation to convert them to a uniform format.

3.3. Translation

The target data should be in English format as it is a lingua franca. We use a robust approach of selective translation. With this, we only translate non-English sentences using a translation API, GoogleTranslator, which is a library for detecting a language of a given sentence. Sentences that are detected to not be of ***en*** (non-English) are then translated to English. Transcripts are now ready to be utilized in the creation of quizzes.

3.4. Quiz Generation

In this sub-section, we use the cleaned and translated transcript data to generate close-ended questions (multiple-choice and true/false questions) as this is supposed to only allow the students to recall what was recently lectured.

3.4.1 Summarization

For concise question generation, we need to summarize the transcript data to obtain key points. This summary will become the input for question generation. We leverage Gemini API for this, which has a Text Summarization feature and is already trained with a large corpus. The Gemini API from Vertex AI provides access to the most recent generative models from Google, known as the Gemini family of models (Ahmed and Islam, 2024). We access this API in Python using the genai library, which offers unique features such as summarization, classification, text generation, code generation and explanation, and chat.

3.4.2. Question Prompt formulation

We use creative prompts that will guide the Gemini API towards generating quiz questions based on summary obtained in the previous step. An example of a prompt is shown below.

> **"Based on the following summary, generate 5 multiple-choice questions with 4 answer choices each and 5 true/false questions, focusing on the key points and main ideas discussed:**
>
> **[SUMMARY_TEXT]"**

3.4.3. Processing Generated Response

The Gemini API should respond with a generated narrative markdown response containing potential quiz questions. Therefore, we parse the response to extract individual questions, answer choices, and potentially correct answers using text manipulation techniques. These are then saved in an Excel spreadsheet for future reference.

3.5. Integration and Configuration

In putting it all together, we use Streamlit as a web framework for building a web application. Streamlit is an open-source Python framework that enables creation of interactive data applications (Dani, 2022; Khorasani et al., 2022). The objective is to create a web application that contains the model described and provides students with a well-organized user interface that displays the generated questions. Quite a number of technologies were utilized to achieve this, and these a listed in Table 3.

To configure Streamlit, we install it as a Python module using the command ***!pip install streamlit*** on Google Colab. This will provide an environment where we can code html pages while referencing the python code representing our model. In essence, we have created a machine learning model that acquires transcripts using Graph API, and then use the Gemini API that is able to analyze the transcript and generate quizzes. An advantage of using streamlit is that it comes with a Streamlit Server for deployment of web applications. This only requires that all files be saved in a GitHub repo and go to the Streamlit website to connect to the repo created.

Table 3. Technologies utilized in this paper.

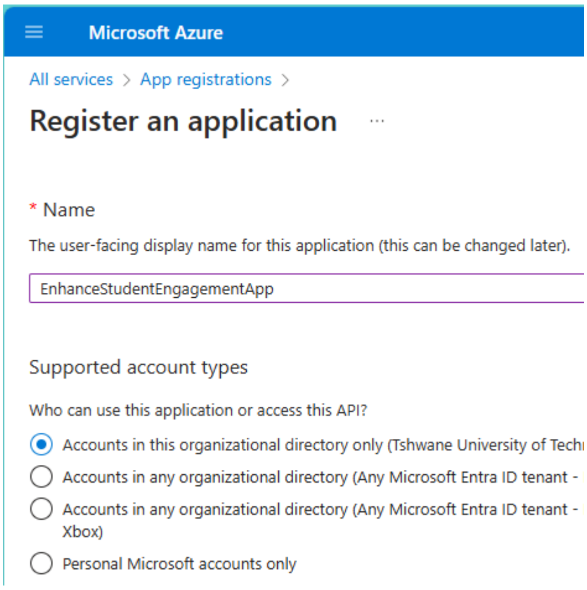
Technologies	
Programming language	Python's extensive libraries and frameworks allowed for the development of this AI project (Teoh and Rong, 2022).
Application programming interfaces	An API is a software intermediary that facilitates communication between two applications. APIs provide a convenient means to retrieve and distribute data (Lamothe et al., 2021). • GraphAPI • Gemini API
ASR service	MoviePy Python module.

NLP libraries	NLTK, Pandas, StopWords, GoogleTranslator
Web framework	Streamlit
Storage	Excel spreadsheet

4. Demonstration and Results

In this section, we demonstrate the entire workflow of the proposed solution through figures and illustrations. In doing so, we discuss and present results as per demonstration of each workflow stage.

During the data collection and preparation stage, we obtained a Teams session link of a meeting that has ended and saved to a OneDrive folder. To access the recording and the transcript, we utilized GraphAPI. However, we needed to get authorization and permission prior to accessing the files. Figure 2, 3, and 4, show the results of following step 1 presented in Table 2.



Microsoft Azure

All services > App registrations >

Register an application ...

* Name

The user-facing display name for this application (this can be changed later).

EnhanceStudentEngagementApp

Supported account types

Who can use this application or access this API?

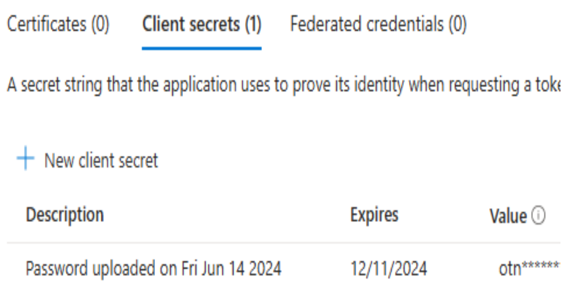
● Accounts in this organizational directory only (Tshwane University of Techı

○ Accounts in any organizational directory (Any Microsoft Entra ID tenant -

○ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Xbox)

○ Personal Microsoft accounts only

Figure 2. Registering an application in AAD.



Certificates (0) Client secrets (1) Federated credentials (0)

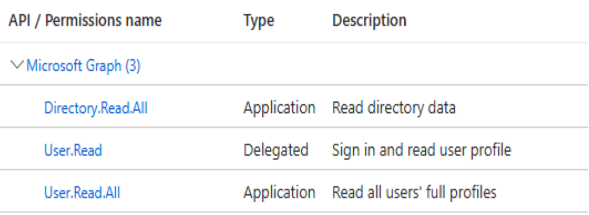
A secret string that the application uses to prove its identity when requesting a toke

+ New client secret

Description	Expires	Value ⓘ
Password uploaded on Fri Jun 14 2024	12/11/2024	otn******

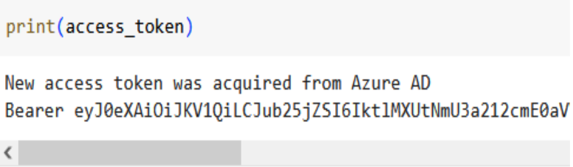
Figure 3. Obtaining secret key for application.

We then needed to obtain the access token. To do this, we made requests to the MS Graph API to acquire the token using the msal library for Python. Figure 5 illustrates that we were able to obtain an access token from Azure AD.



API / Permissions name	Type	Description
∨ Microsoft Graph (3)		
Directory.Read.All	Application	Read directory data
User.Read	Delegated	Sign in and read user profile
User.Read.All	Application	Read all users' full profiles

Figure 4. Granting permissions to read and write.

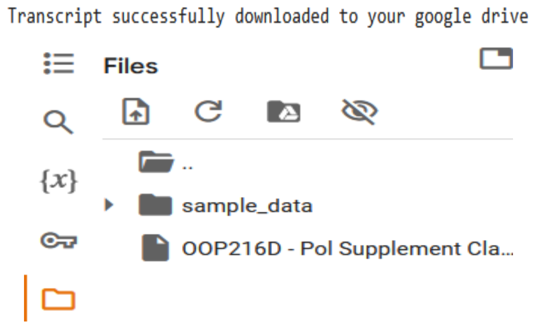


```
print(access_token)
```

```
New access token was acquired from Azure AD
Bearer eyJ0eXAiOiJKV1QiLCJub25jZSI6IktlMXUtNmU3a212cmE0aV
```

Figure 5. Obtaining directory of the recorded files.

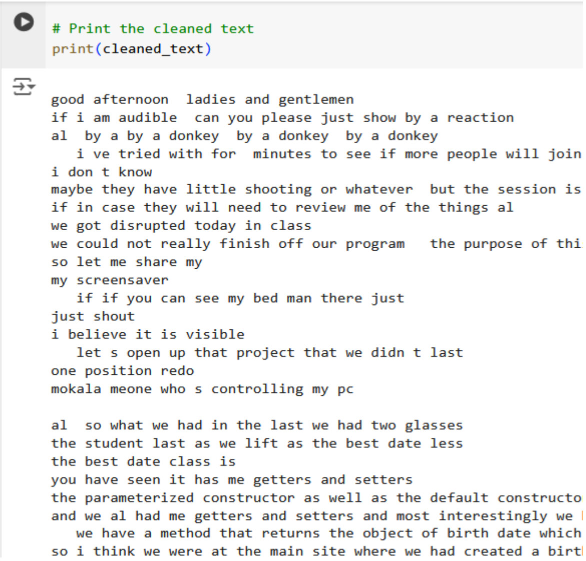
We then created a request, supplying a graph url that specifies the target directory of a recorded Teams video and transcript, and used JavaScript Object Notation (JSON) to format the output. JSON is A conventional textual format used to represent organised data (Pezoa, 2016). The next step, we use the file ids to download the recorded video and transcripts. Seeing that we have the transcripts file already, we did not require the video. Figure 6 shows successful download of the transcript file onto Google Drive, where preprocessing and cleaning is to be conducted.



```
Transcript successfully downloaded to your google drive
```

Figure 6. Downloaded transcript using GraphAPI.

We pre-processed and cleaned transcript data using pandas and NLP libraries to prepare it for the question generation model. Figure 7 shows processed data. Table 4 shows the elements removed to ensure data preparation.



```
# Print the cleaned text
print(cleaned_text)
```

```
good afternoon  ladies and gentlemen
if i am audible  can you please just show by a reaction
al  by a by a donkey  by a donkey  by a donkey
   i ve tried with for  minutes to see if more people will join
i don t know
maybe they have little shooting or whatever  but the session is
if in case they will need to review me of the things al
we got disrupted today in class
we could not really finish off our program   the purpose of thi
so let me share my
my screensaver
   if if you can see my bed man there just
just shout
i believe it is visible
   let s open up that project that we didn t last
one position redo
mokala meone who s controlling my pc

al  so what we had in the last we had two glasses
the student last as we lift as the best date less
the best date class is
you have seen it has me getters and setters
the parameterized constructor as well as the default constructo
and we al had me getters and setters and most interestingly we
   we have a method that returns the object of birth date which
so i think we were at the main site where we had created a birt
```

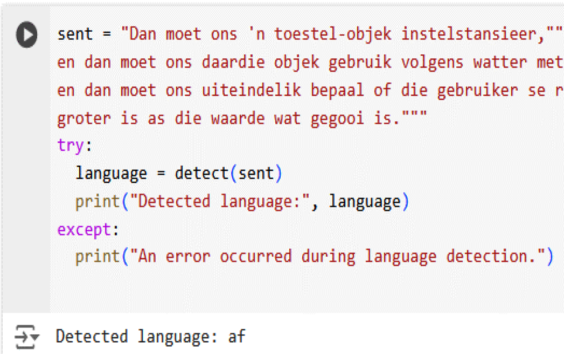
Figure 7. Processed data sample

Table 4. Elements removed from transcript data.

Element	Examples
Pause words	“uh”, “like”, “you know”, “so”, “I mean”, “actually”
Disfluencies	“umm”, “huh” This matches repeated pause words.
Formatting	Timestamps
Speaker tags	“Relebogile Makhulu Langa”, “K Lekaote”, “K Molalatladi”, “M Moteane”

Although the used transcript is in the English language, we have in place the language detection and translation modules. To demonstrate this, we manually extract a line from the transcript and manually change it to Afrikaans for testing. In Figure 8, we manually supply an Afrikaans line to the langdetect module to determine if this work should it be different languages are in the transcript. According to Abainia (2018), currently supports up to 55 languages out of the box and it is lightweight to implement.

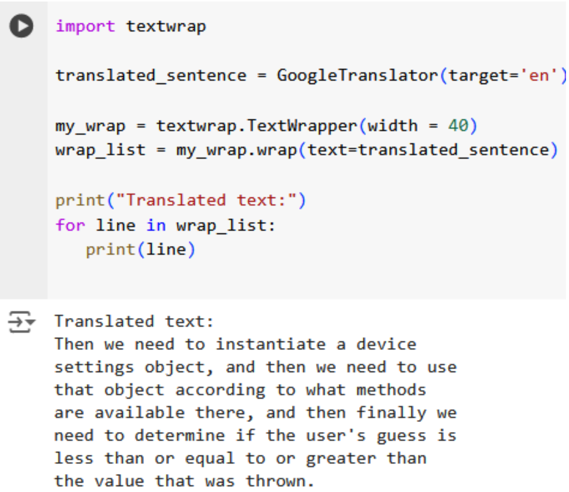
Language detection is functioning correctly, as evidenced by the output being identified as **"af,"** which is the abbreviation for Afrikaans. Subsequently, we employ the identified language for the purpose of translation. This aids the translation model in determining the specific languages for translation, both the source language and the target language. By utilising GoogleTranslator, we input the identical identified Afrikaans sentence and assess its translation capability. Figure 9 demonstrates the translation process.



```
sent = "Dan moet ons 'n toestel-objek instelstansieer,""
en dan moet ons daardie objek gebruik volgens watter met
en dan moet ons uiteindelik bepaal of die gebruiker se r
groter is as die waarde wat gegooi is."""
try:
  language = detect(sent)
  print("Detected language:", language)
except:
  print("An error occurred during language detection.")
```

```
Detected language: af
```

Figure 8. Language detection ability.



```
import textwrap

translated_sentence = GoogleTranslator(target='en')

my_wrap = textwrap.TextWrapper(width = 40)
wrap_list = my_wrap.wrap(text=translated_sentence)

print("Translated text:")
for line in wrap_list:
    print(line)
```

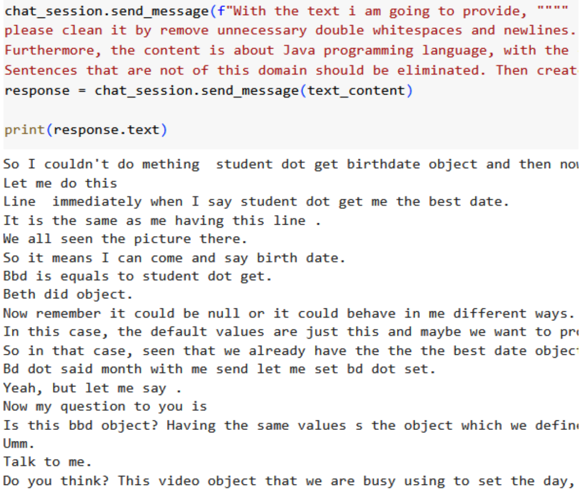
```
Translated text:
Then we need to instantiate a device
settings object, and then we need to use
that object according to what methods
are available there, and then finally we
need to determine if the user's guess is
less than or equal to or greater than
the value that was thrown.
```

Figure 9. Translation of Afrikaans text to Eng.

In the next task, we then introduced the Gemini API for question generation where we first applied text summarization followed by creation of template. For further cleaning, we noted that there were still unnecessary double whitespaces and newlines still present. Using Google’s genai Python module, we cleaned the remaining whitespaces and newlines and also created a summarized markdown output using a prompt shown in Figure 10.

We utilized the template used in Section 3.4 to inform the genai model how to generate questions that the students will need to answer post the online class. In utilizing the template, we found that the model was able to generate the questions as expected. Figure 11 shows a prompt to the model and the associated response. The authors then developed a web application powered by Streamlit framework, where the questions are presented to the students in form dialog. For ensuring user privacy, the student emails obtained collected the attendance register are not illustrated. However, all student emails found within the attendance register excel file are sent an email using the smtplib module for Python that contains a link to the web application for answering questions.

The Smtplib module provides a client session object for sending email to any online machine with an SMTP or ESMTP listener daemon (Goerzen, 2004). These closed-ended questions require no knowledge application, so the web application link is only valid for 5 minutes. This also prevents students from sharing answers.

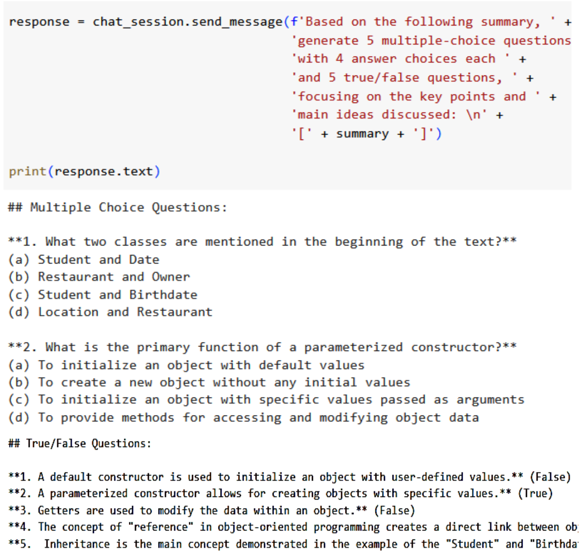


```
chat_session.send_message(f"With the text i am going to provide, """"
please clean it by remove unnecessary double whitespaces and newlines.
Furthermore, the content is about Java programming language, with the
Sentences that are not of this domain should be eliminated. Then creat
response = chat_session.send_message(text_content)

print(response.text)
```

```
So I couldn't do mething  student dot get birthdate object and then no
Let me do this
Line  immediately when I say student dot get me the best date.
It is the same as me having this line .
We all seen the picture there.
So it means I can come and say birth date.
Bbd is equals to student dot get.
Beth did object.
Now remember it could be null or it could behave in me different ways.
In this case, the default values are just this and maybe we want to pr
So in that case, seen that we already have the the the best date objec
Bd dot said month with me send let me set bd dot set.
Yeah, but let me say .
Now my question to you is
Is this bbd object? Having the same values s the object which we defin
Umm.
Talk to me.
Do you think? This video object that we are busy using to set the day,
```

Figure 10. Sample of summarized markdown.



```
response = chat_session.send_message(f'Based on the following summary, ' +
                                     'generate 5 multiple-choice questions
                                     'with 4 answer choices each ' +
                                     'and 5 true/false questions, ' +
                                     'focusing on the key points and ' +
                                     'main ideas discussed: \n' +
                                     '[' + summary + ']')

print(response.text)
```

```
## Multiple Choice Questions:

**1. What two classes are mentioned in the beginning of the text?**
(a) Student and Date
(b) Restaurant and Owner
(c) Student and Birthdate
(d) Location and Restaurant

**2. What is the primary function of a parameterized constructor?**
(a) To initialize an object with default values
(b) To create a new object without any initial values
(c) To initialize an object with specific values passed as arguments
(d) To provide methods for accessing and modifying object data

## True/False Questions:

**1. A default constructor is used to initialize an object with user-defined values.** (False)
**2. A parameterized constructor allows for creating objects with specific values.** (True)
**3. Getters are used to modify the data within an object.** (False)
**4. The concept of "reference" in object-oriented programming creates a direct link between ob
**5.  Inheritance is the main concept demonstrated in the example of the "Student" and "Birthda
```

Figure 11. Generated questions from summarized transcript.

The web application designed and developed is shown in Figure 13 and Figure 14.

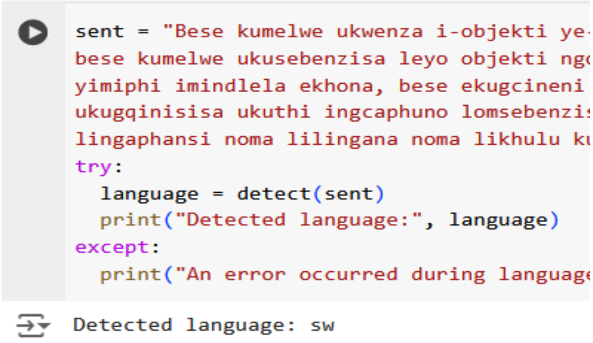
5. Evaluation and discussion

In this section, evaluation of methods, tools, technologies, and strategies presented in Section 4 is provided. The results are further discussed in comparison to those in the literature.

As presented, GraphAPI is currently the most suitable API for accessing data on Microsoft 365 platform (Lappin, 2022). Till recently, most of the available tools and strategies involved the use of transcripts datasets found on the Internet. This paper has shown detailed processes to follow when utilizing the GraphAPI. However, it should be taken into account that the utilization of GraphAPI to access OneDrive/Teams requires authorization. This implies that the proposed solution would be used by lecturers in ensuring students are assessed per online class.

In assessing the collected transcripts (both Teams generated and those derived from ASR service), we found that the information gets chopped when certain words are not pronounced properly. For instance, we observed that in the transcript, "**text**" was transcribed instead of "**test**", see Figure 11. This was probably due to the speed or volume used on the pronunciation of the word, or poor audio quality and background noise as also experienced in studies of (Kostaki and Karayianni, 2021; Augustina, 2022; Nifriza, 2022). Fitria (2022) identified these factors and suggested that presenters should always train to be concise, have stable internet connectivity, and use the same pace throughout the online class.

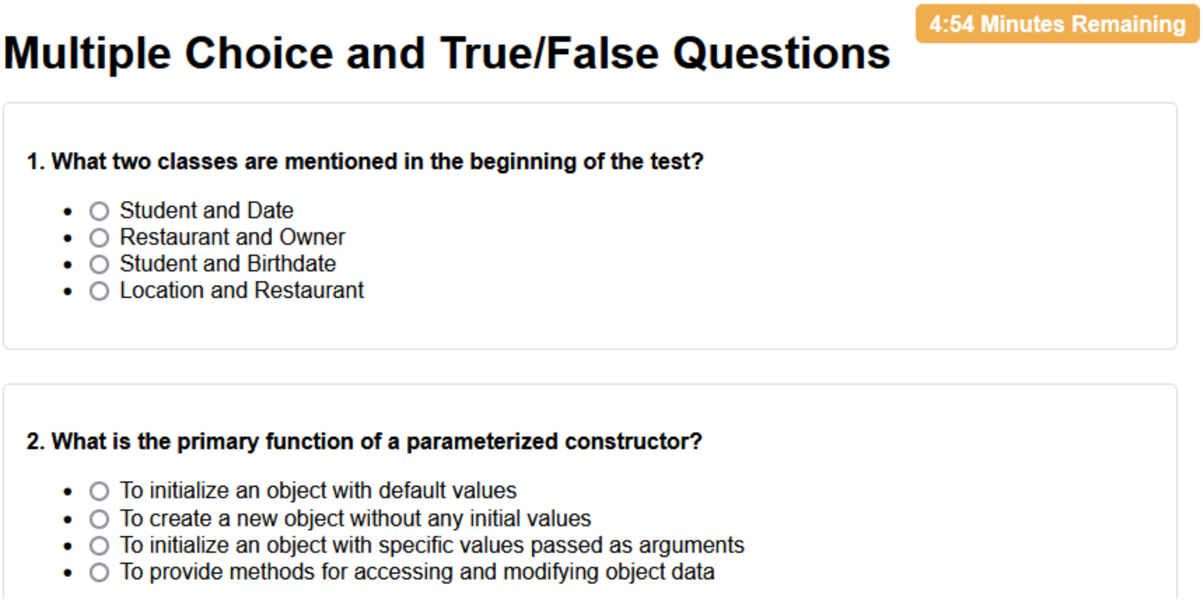
In terms of translation, the choice to use langdetect was on simplicity reasons, however, we observed it was not suitable for African-based studies, due to limited support for African languages. For example, in Figure 12, when attempting to detect the sentence in isiZulu, one of South Africa's official languages, it identified it as "**sw**", which is Swahili, based on ISO 639 language codes.



```
sent = "Bese kumelwe ukwenza i-objekti ye
bese kumelwe ukusebenzisa leyo objekti ng
yimiphi imindlela ekhona, bese ekugcineni
ukugqinisisa ukuthi ingcaphuno lomsebenzi
lingaphansi noma lilingana noma likhulu k
try:
  language = detect(sent)
  print("Detected language:", language)
except:
  print("An error occurred during language
```

```
Detected language: sw
```

Figure 12. Wrong language detection langdetect.



4:54 Minutes Remaining

Multiple Choice and True/False Questions

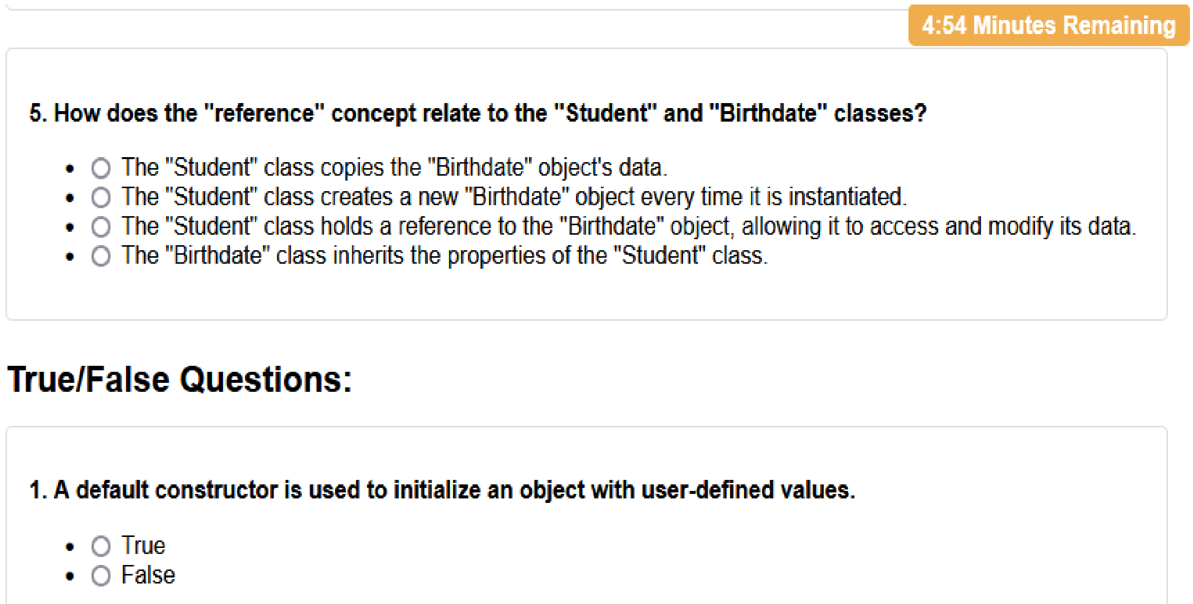
1. What two classes are mentioned in the beginning of the test?

- ○ Student and Date
- ○ Restaurant and Owner
- ○ Student and Birthdate
- ○ Location and Restaurant

2. What is the primary function of a parameterized constructor?

- ○ To initialize an object with default values
- ○ To create a new object without any initial values
- ○ To initialize an object with specific values passed as arguments
- ○ To provide methods for accessing and modifying object data

Figure 13. Web application showing questions.



4:54 Minutes Remaining

5. How does the "reference" concept relate to the "Student" and "Birthdate" classes?

- ○ The "Student" class copies the "Birthdate" object's data.
- ○ The "Student" class creates a new "Birthdate" object every time it is instantiated.
- ○ The "Student" class holds a reference to the "Birthdate" object, allowing it to access and modify its data.
- ○ The "Birthdate" class inherits the properties of the "Student" class.

True/False Questions:

1. A default constructor is used to initialize an object with user-defined values.

- ○ True
- ○ False

Figure 14. Web application showing questions.

The GoogleTranslator module was consistent with the translations and was suitable for this task. Utilizing Gemini API (genai) was beneficial to this task as it allowed the authors to use the pre-trained gemini-flash-1.0 model with ease. The model was able to summarize the transcripts data and generate relevant questions. However, for true/false questions, further string manipulation techniques are required as the possible answers were found at the end of each question. Although these limitations exist, they are all dependent on the transcript's quality. As such, the cleaner the transcript, the better the translation, summarization, and question generation.

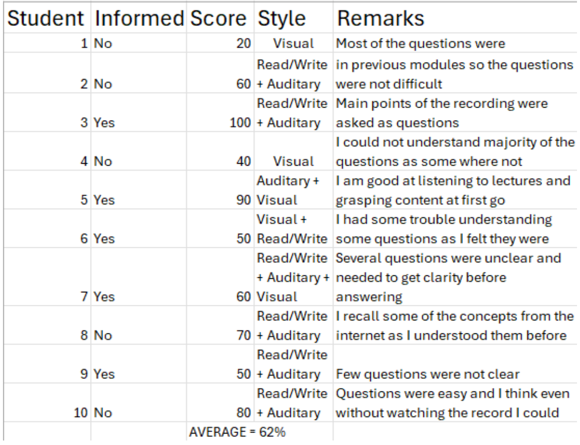
In evaluating the proposed solution, the experimental setup was as follows, due to a time constraint. We obtained a recording of a meeting and shortened it to 30 minutes. We selected ten random students to watch the recording, of which five were informed that there will be a quiz at the end of the recording, and the other five were not. This is to assess whether students knowing they have to answer a quiz and not knowing contributes to the level of attention they have to pay. We further ensured that the transcript does not contain any data that is beyond the 30^{th} minute to ensure relevancy of the questions to be generated.

Per Figure 15, we expected the group of students who knew about the quiz at the end of the recording to score better than the other group. This shows that student involvement depends on knowing the tasks they must complete during or after sessions to increase their attendance rate. This is in line with the findings of Inder (2021), where *Value to Students* is identified as a theme for influencing student engagement.

In additional investigation, we asked these students to examine the proposed solution, especially with score-related reasons. Without knowing they had to take a quiz, students 1 and 4 said the questions were ambiguous. Considering their learning styles, we found that these students are visual learners, and the lecture/recording used had fewer visual examples. This is supported by Willis (2024), as the author mentions the significance of inclusive lectures to support personalized learning.

Students 2, 8, 10 were also part of the uninformed group yet they were able to score well in their quiz. According to their remarks, they had background knowledge of the presented content. This fits with the theme *Student Experience* as identified by Inder (2024). Students 3, and 5, had positive remarks. We notice that the questions were based on their preferred learning style.

The remaining students (6, 7, 9) gave remarks of ambiguity of questions which led to poor scores obtained. The presence and availability of chatbots during quizzes, although impractical, can be useful to students to obtain clarity of the question. Thomas (2024) concurs that AI can enhance engagement through chatbots and virtual assistants. This paper disallows the use of chatbots during quizzes as it might give away solutions to the presented questions.



Student	Informed	Score	Style	Remarks
1	No	20	Visual	Most of the questions were
2	No	60	Read/Write + Auditary	in previous modules so the questions were not difficult
3	Yes	100	Read/Write + Auditary	Main points of the recording were asked as questions
4	No	40	Visual	I could not understand majority of the questions as some where not
5	Yes	90	Auditary + Visual	I am good at listening to lectures and grasping content at first go
6	Yes	50	Visual + Read/Write	I had some trouble understanding some questions as I felt they were
7	Yes	60	Read/Write + Auditary + Visual	Several questions were unclear and needed to get clarity before answering
8	No	70	Read/Write + Auditary	I recall some of the concepts from the internet as I understood them before
9	Yes	50	Read/Write + Auditary	Few questions were not clear
10	No	80	Read/Write + Auditary	Questions were easy and I think even without watching the record I could
		AVERAGE = 62%		

Figure 15. Student feedback.

6. Conclusion

The paper accomplished its objective to propose a method for evaluating the level of student engagement in an online course to ascertain whether they are actively engaged and present. The GraphAPI was instrumental in the acquisition of the recordings/transcripts and can be considered a dependable source for the collection of Microsoft Office data. Lecturers can evaluate students' participation levels by employing transcripts generated from the discontinued online course. Langdetect's limited language detections do not significantly impact the results, as the English language facilitates teaching and learning.

The web application generates questions on online class content, and the scores are saved in an Excel spreadsheet for future reference. However, student's learning preferences can be taken into consideration when generating questions. Additionally, students can have an option to alter the question difficulty levels to have personalized questions. This will in turn lead to personalized learning experiences. The revolutionary impact of GenAI will persist in higher education institutions as it has the potential to be highly beneficial for both students and lecturers. This has complemented the current AI research in order to increase student engagement in online classes. Our future objective is to incorporate this solution into Microsoft Teams. Additionally, we intend to enhance the accuracy of our translations by employing a variety of translation modules. Finally, we aim to expand the use of GenAI to other educational contexts such as automated grading and feedback.

7. References

Abainia, K. (2018). Toward accurate social media language identification: combining language features with a graphical approach. 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS). https://doi.org/10.1109/pais.2018.8598531

Adedoyin, O. B. and Soykan, E. (2020). Covid-19 pandemic and online learning: the challenges and opportunities. Interactive Learning Environments, 31(2), 863-875. https://doi.org/10.1080/10494820.2020.1813180

Agustina, D. (2022). Teacher's reflection on the decreasing participation within online classes. Beyond Linguistika, 5(1). https://doi.org/10.36448/bl.v5i1.2516

Ahmed, I. and Islam, R. (2024). Gemini-the most powerful llm: myth or truth. https://doi.org/10.36227/techrxiv.171177477.70151414/v1

Almusaed, A., Almssad, A., Yitmen, I., & Homod, R. Z. (2023). Enhancing student engagement: harnessing "aied"'s power in hybrid education—a review analysis. Education Sciences, 13(7), 632. https://doi.org/10.3390/educsci13070632

Azizan, M. T. (2023). Promoting cognitive engagement using technology enhanced book-end method in online active learning strategies. Asean Journal of Engineering Education, 7(2), 8-16. https://doi.org/10.11113/ajee2023.7n2.129

Bashir, A., Bashir, S., Rana, K., Lambert, P. A., & Vernallis, A. B. (2021). Post-covid-19 adaptations; the shifts towards online learning, hybrid course delivery and the implications for biosciences courses in the higher education setting. Frontiers in Education, 6. https://doi.org/10.3389/feduc.2021.711619

Coman, C., Țîru, L. G., Meseșan-Schmitz, L., Stanciu, C., & Bularca, M. C. (2020). Online teaching and learning in higher education during the coronavirus pandemic: students' perspective. Sustainability, 12(24), 10367. https://doi.org/10.3390/su122410367

Dani, H. (2022). Review on frameworks used for deployment of machine learning model. International Journal for Research in Applied Science and Engineering Technology, 10(2), 211-215. https://doi.org/10.22214/ijraset.2022.40222

Ferrer, J., Ringer, A., Saville, K., Parris, M. A., & Kashi, K. (2020). Students' motivation and engagement in higher education: the importance of attitude to online learning. Higher Education, 83(2), 317-338. https://doi.org/10.1007/s10734-020-00657-5

Fitria, T. N. (2022). Utilizing text-to-speech technology: natural reader in teaching pronunciation. JETLEE : Journal of English Language Teaching, Linguistics, and Literature, 2(2), 70-78. https://doi.org/10.47766/jetlee.v2i2.312

Gaim, F., Yang, W., & Park, J. C. (2022, June). GeezSwitch: language identification in typologically related low-resourced East African languages. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 6578-6584).

Goerzen, J. (2004). Simple message transport protocol. Foundations of Python Network Programming, 197-210. https://doi.org/10.1007/978-1-4302-0752-8_10

Hafeez, M., Naureen, S., & Sultan, S. (2022). Quality indicators and models for online learning quality assurance in higher education. Electronic Journal of E-Learning, 20(4), pp374-385. https://doi.org/10.34190/ejel.20.4.2553

Inder, S. (2021). Factors influencing student engagement for online courses: a confirmatory factor analysis. Contemporary Educational Technology, 14(1), ep336. https://doi.org/10.30935/cedtech/11373

Kamalova, N. I. (2024). HOW TO CREATE PROFESSIONAL VIDEO CONTENT IN PYTHON. World Scientific Research Journal, 26(1), 108–112. Retrieved from http://www.wsrjournal.com/index.php/wsrj/article/view/3179

Khorasani, M., Abdou, M., & Fernández, J. H. (2022). Web application development with streamlit. https://doi.org/10.1007/978-1-4842-8111-6

Kostaki, D. and Karayianni, I. (2021). Houston, we have a pandemic: technical difficulties, distractions, and online student engagement. https://doi.org/10.31219/osf.io/6mrhc

Lamothe, M., Guéhéneuc, Y., & Shang, W. (2021). A systematic review of api evolution literature. ACM Computing Surveys, 54(8), 1-36. https://doi.org/10.1145/3470133

Lappin, J. (2022). Project cortex and the future of document management in office 365. IQ: The RIMPA Quarterly Magazine, 38(2), 35-41.

Lee, K., Fanguy, M., Bligh, B., & Lu, X. S. (2021). Adoption of online teaching during the COVID-19 Pandemic: a systematic analysis of changes in university teaching activity. Educational Review, 74(3), 460–483.

https://doi.org/10.1080/00131911.2021.1978401

Lei, M., Medwell, J., & Wray, D. (2021). How does experience of online collaborative learning affect the views of trainee teachers in a Chinese context?. INTED2021 Proceedings. https://doi.org/10.21125/inted.2021.1125

Mahani, S. (2023). Applying culturally relevant pedagogy to online learning. Cultural and Pedagogical Inquiry, 15(1), 23-34. https://doi.org/10.18733/cpi29708

Meşe, E., & Sevilen, Ç. (2021). Factors influencing EFL students' motivation in online learning: A qualitative case study. Journal of Educational Technology and Online Learning, 4(1), 11-22.

Mishra, A. (2022). Sending emails using graph api. Microsoft Azure for Java Developers, 197-220. https://doi.org/10.1007/978-1-4842-8251-9_10

Pezoa, F., Reutter, J. L., Suárez, F., Ugarte, M., & Vrgoč, D. (2016). Foundations of json schema. Proceedings of the 25th International Conference on World Wide Web. https://doi.org/10.1145/2872427.2883029

Priya, T. J., Priya, K. P., Jenneyl, L. R., & Uma, K. V. (2022). Automatic question generation from video. Computational Intelligence in Pattern Recognition, 366-372. https://doi.org/10.1007/978-981-19-3089-8_35

Ratni, N. P. (2024). Digital empowerment in English language teaching (perspective of jean piaget-learning theory). Proceedings of International Conference on Education, 2(1), 359-369. https://doi.org/10.32672/pice.v2i1.1381

Sadler, T. D., Mensah, F. M., & Tam, J. (2024). Artificial intelligence and the journal of research in science teaching. Journal of Research in Science Teaching, 61(4), 739-743. https://doi.org/10.1002/tea.21933

Stepanek, H. (2020). Thinking in pandas. https://doi.org/10.1007/978-1-4842-5839-2

Stokel-Walker, C. and Noorden, R. V. (2023). What chatgpt and generative ai mean for science. Nature, 614(7947), 214-216. https://doi.org/10.1038/d41586-023-00340-6

Teoh, T. T., & Rong, Z. (2022). Artificial intelligence with python (pp. 3-328). New York: Springer.

Tiwow, G. M., Rawis, J. A., Lengkong, J. S. J., & Rotty, V. N. J. (2023). Strengthening the acceleration of digital transformation in developing a digital economy curriculum. International Journal of Information Technology and Education, 2(3), 108-117. https://doi.org/10.62711/ijite.v2i3.130

Tualaulelei, E., Burke, K., Fanshawe, M., & Cameron, C. (2021). Mapping pedagogical touchpoints: exploring online student engagement and course design. Active Learning in Higher Education, 23(3), 189-203. https://doi.org/10.1177/1469787421990847

Wang, M. and Hu, F. (2021). The application of nltk library for python natural language processing in corpus research. Theory and Practice in Language Studies, 11(9), 1041-1049. https://doi.org/10.17507/tpls.1109.09

Xie, J. and Correia, A. (2023). The effects of instructor participation in asynchronous online discussions on student performance: a systematic review. British Journal of Educational Technology, 55(1), 71-89. https://doi.org/10.1111/bjet.13350

Zimu, Y. (2024). Examining the Relationship between Teacher Self-Efficacy and Student Engagement in Technology-enhanced Learning Environments. International Journal of New Developments in Education, 6(2). https://doi.org/10.25236/ijnde.2024.060219