

# Unravel: AI-Driven Educational Platform

Dalia Mohamed Sobhy, Khalid Khalil, Ahmed Nabil, and Aly Barakat

**Abstract**—2020 has been a challenging year for the educational field due to the COVID-19 virus. Changing to online learning was not easy for both the students and the educators. In an attempt to address the problems of online learning to help educators around the world, this paper proposes unravel application to facilitate the job of educators and give them a smooth experience while providing educational materials to their students. Unravel is a web-based solution that enables educators from all backgrounds to upload their lectures to an online cloud video hosting service. Using cutting-edge Speech to Text Technology and NLP tools, video files are transcribed, and a timeline is provided by the application which allows the user to manipulate the file using the generated text. It facilitates the video editing process for users with no video editing experience via an intuitive graphical user interface. The platform will also enable educators to monitor the students and get analytics from the system about the views and will enable an anonymous feedback system so educators can get reviews from their students. The objective of the proposed approach is to build a comprehensive platform that fills the gap between educators and students while unraveling the complex educational problems in online learning.

**Index Terms**—AI, NLP, web services, online learning, web application.

## I. INTRODUCTION

The COVID-19 pandemic has greatly affected our lives, the stay-in-home rules recently introduced to combat infections have led to many aspects of our lives shifting to a no-contact or remote model. From work, shopping, education, and many other fields, everything has been impacted by this change. Students were mostly affected by the switch to the remote education model [1]. In particular, COVID-19 has severely impacted education in many ways, the restrictions enforced by governments and world health organizations have led to thousands of school closures around the world and millions of students forced to study from home. According to the UNESCO [2], COVID-19 has forced 1.4 billion students to stay at home. Neither students nor instructors were ready for such a change. Even if life is returning to normal, various countries have presumed lockdown due to the new COVID variant, *Omicron* [3]. Further, the COVID-19 outbreak has accelerated the use of online learning, which encouraged several

schools/universities to adopt online learning strategies in post-pandemic time as well (e.g. online degrees). This new paradigm will enhance distance learning, which will cause a revolution in the education era. There are already many systems and platforms made for remote learning, but they do not usually address the different challenges that remote education imposes. Further, instructors usually have limited experience in dealing with computers and technology in general; it is hard for them to deliver professional-grade materials for their students [4].

Based on previous surveys [5], some tutors and students face problems in adopting e-learning. The tutors' problems are summarized as follows: 1) Difficulties adapting to new technologies; 2) Wasting time preparing the materials; some of them are even re-recording the lectures (e.g. to reduce the lectures' time, remove unnecessary/incorrect information, and many filler words like (um, uh, or blanks), etc); 3) Overhead in learning how to deal with hectic video editing. Additionally, the students' problems are summarized as follows: 1) They were overwhelmed with materials scattered between different platforms, some students missed some materials as they did not notice the presence of other platforms (e.g., having materials on both Moodle and Classroom); 2) Loss of focus due to boring long unprofessional videos; 3) Some students may experience bad internet connections (i.e. denied access to on-line sessions or even entered the sessions but with interrupted connection). In this paper, an AI-driven educational platform "Unravel"<sup>1</sup> is proposed to ease the educational process. In particular, our objective is to *create a single online educational platform that offers both teachers and students a tool that simulates the classroom* by providing the following novel features:

**Single channel educational platform:** Teachers can create their courses and add all the related materials (videos, documents, and data) easily.

**Seamless AI-driven video editing tool:** Teachers can upload their videos, have them transcribed, and edit their videos just by swiping on the text they want to remove/change at a glance. In other words, the core feature of Unravel is video editing that utilizes speech transcription to offer instructors an editing experience similar to editing documents in editors like Microsoft Word. By moving words around and removing words with the cursor, the instructor can edit the content of his materials without spending hours on professional video editing software.

**Video to Text:** The video is transcribed to text, which could be a potential way to improve the student's learning journey.

The manuscript is structured as follows: Section II presents the related work, Section III provides detailed description of

Manuscript received March, 2022; revised May, 2022.

Dalia Mohamed Sobhy is with Computer Engineering Department, Arab Academy of Science and Technology and Maritime Transport, Alexandria, Egypt (e-mail: dalia.sobhi@aast.edu).

Khalid Khalil is with Facebook, London, UK (e-mail: khalid.k@aol.com).

Ahmed Nabil is with Incorta Company, San Mateo, California, USA (e-mail: ahmed.nasr@incorta.com).

Aly Barakat is with Minly Company, Cairo, Egypt (e-mail: abarakat@minlyapp.com).

<sup>1</sup> <https://github.com/UnravelAI>

the proposed architecture. Section IV illustrates the implementation and performance evaluation of the proposed model. Finally, Section V concludes the work and provides some potential directions for future work.

## II. RELATED WORK

Covid-19 has led to thousands of school closures. This has raised the awareness of the need for remote education as an immediate response [6]. Despite that remote education imposes challenges for both students and instructors [7]. Communication and content delivery are one of the most affected aspects of going remote [8], [9]. In this context, several virtual platforms have been proposed in the COVID-19 era addressing the communication aspect. For instance, Zoom as a video conference meetings tool has highly penetrated the market in early 2020 [10]. In particular, educational institutions (e.g. universities) have used Zoom for hosting online lectures/tutorials. For instance, the American College of cardiology has integrated Zoom and Microsoft Teams to facilitate continued fellowship educational activities [11]. In addition, Faval et al. [12] have proposed BigBlueButton application, which provides high-quality audio, video, and screen sharing application using the browser's built-in support for web real-time communication (WebRTC) libraries. Other communication virtual tools are Skype in education, Google Meet, Webex, and other platforms [13]. These virtual platforms had a positive impact on online learning.

However, in some situations, some students may experience bad internet connections. In the latter case, recorded lectures are necessary. Therefore, the instructors can then upload their recorded lectures on their deployed classroom platform (e.g. moodle [14], canvas, google classroom, etc). The other challenge is content delivery in terms of recorded lectures, materials management, assessment tools, etc. Various platforms support content delivery. Google suite for education is a suite of software and productivity tools (e.g. Google Classroom, Google Docs, Google sheets, etc) that simulates the virtual classroom on the cloud. Other web-based learning management systems, such as moodle [15] and canvas [16], are used by several institutions worldwide.

Focusing on recorded lectures, instructors do not have time to learn hectic unnecessary video editing skills, so the videos are long and full of filler words like (um, uh, or blanks). Further, in some contexts, instructors may need to remove some parts of recorded lectures due to incorrect content mentioned unintentionally. To solve the prior problems, video editing tools are necessary, but most of the available tools are generally hard and have a steep learning curve. Instructors can then use another tool for transcribing videos, which is subscription-based (i.e. may be expensive in some contexts like long videos). After that, they can upload their videos without instructions on one of the platforms like Google's Classroom, YouTube, Moodle, Google drive. Every platform mentioned is lacking many basic features. For example, you can not upload a document on YouTube, instead, you upload the document on Moodle and a video on YouTube and attach the link on Moodle, and all that without

editing nor performing any AI easing feature. To the best of our knowledge, there is no virtual educational platform that integrates smart video editing tools, video-text functionalities, and materials management in one single channel in the research literature.

## III. UNRAVEL: AI-DRIVEN EDUCATIONAL PLATFORM

In this section, the video processing and editing feature in Unravel, the system requirements, and the main components of the proposed architecture are discussed.

### A. Video Processing and Editing

The core feature of Unravel is video editing which utilizes speech transcription to offer instructors an editing experience similar to editing documents in editors like Microsoft Word. First, the instructor uploads the video saved in particular storage, which is then ready for editing whenever needed. By moving words around and removing words by the cursor, the instructor can edit the content of his/her materials without spending hours on professional video editing software. Fig. 3-A depicts the process of video management. The first step is transcoding the video to streamable content with low quality to give the instructor/video-editor the ability to view his/her video and edit it in a previewing mode (*Post-uploading flow* presented in Fig. 3-A). The following step is passing the video through a video editing function that runs FFMPEG commands to remove the desired words. Whenever the instructor finishes the video editing, the streaming video, extracted words, and summarised text are ready to be published (*Post- editing flow* in Fig. 3-A).

### B. System Requirements

Firstly, the functional and non-functional requirements are gathered from several stakeholders (such as students and tutors).

*Functional Requirements:* These functional requirements represent the services offered by Unravel application, as follows:

- User Management System
  - Users should be able to register, log in, and retrieve their password when lost.
  - System should send a confirmation email upon the user's registration to verify his email.
  - System should authenticate users upon login and maintain the user session to keep him logged in.
- Video Transformation
  - Users should be able to upload videos and audio files. The system should support popular video and audio file formats.
  - System should process the uploaded videos and automatically transcribe the speech to readable text.
  - Users should be able to manipulate the video or audio file by altering the generated text.
- Data extraction
  - Users can automatically remove filler words such as (uh, um, etc) from their videos or audio files.
  - System should use the generated text to apply topic modeling and extract useful data regarding topics mentioned in the speech.

- Users should be able to extract the speech highlights.
- Users should be able to upload videos and audio files. The system should support popular video and audio file formats.
- System should process the uploaded videos and automatically transcribe the speech to readable text.
- Users should be able to manipulate the video or audio file by altering the generated text.
- **Materials Sharing**
  - Users can generate a link to share the video and processed data to other viewers.
  - Users can choose to restrict the video to particular viewers by setting a password.
  - Viewers should be able to download video(s) if the user allows it.
  - Users should monitor the statistics for their videos through their dashboard.

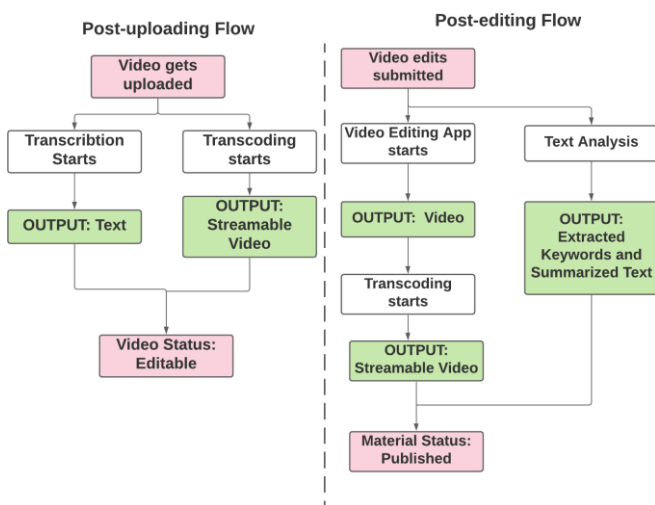


Fig. 1. Video processing and editing.

**Non-Functional Requirements:** These non-functional requirements represent the design constraints, as follows:

- **Usability:** Offering a user-friendly UI and swift processing.
- **Privacy and Security:** Keeping user data safe from intruders by implementing security and privacy protocols.
- **Budget:** Maintaining a reasonable costs bracket and offering multiple pricing packages that suit every need.
- **Maintainability:** Developing a well-structured modular code for future use.
- **Scalability** Offering the ability to automatically scale concerning user demand (i.e. by adopting cloud services).
- **Availability:** Application is guaranteed to stay online and always available to the users with almost no downtime. This depends on the cloud service provider.

### C. Unravel Workflow

The pipeline architecture on the server side is divided into two main parts: *cloud services* and *Unravel Backend server*. Fig. 3-C shows the abstract system architecture. The client interacts with unravel server using the frontend; the server handles the requests and the logic while utilizing several web services. In particular, Unravel's frontend takes video input in a multi-part form and sends it to the backend, which acts as a RESTful API as an HTTP request. The backend sends

the target file to storage (Source) which triggers two functions (audio separation and video streaming). The audio separation function is responsible for extracting the audio and passing it to the Speech to text (STT) engine. The video streaming function then processes the video by transcoding it to a stream- able format (HLS). The output video is pushed to a content delivery network (CDN). The streaming link is generated on the backend and sent back to the frontend. Both video streaming and audio extraction functions send a notification to the backend through HTTP requests to signal their completion. Further details related to software architecture are mentioned below.

- **Frontend:** The main user-facing interface is *webpages*. The frontend is built using ReactJS; a JavaScript frontend library based on the notion of composing components to render a webpage.
- **Backend:** The Unravel backend server is the orchestrating module that handles all the requests from the frontend, manages the database, and triggers various actions in the cloud services. It enforces versatile security measures when dealing with user requests to ensure that data is safe and secure from intruders using authentication techniques. A NodeJS JavaScript runtime environment builds the backend server, which provides fast asynchronous request handling. Typescript is used, which is syntactically stricter than JavaScript. Typescript supports static typing, and provides a more stable runtime code error-free from unhandled dynamic type casting. ExpressJS framework is responsible for handling the requests and creating the server.

The server is designed and built as a REST API following a set of rules defined in [17]. To summarize, the backend server consists of several endpoints that interact between the client (frontend), database, and the several web services utilized.

- **Cloud Services:** To implement the system architecture shown in Fig. 3-C, we have used some cloud services.
  - **Audio transcribing service:** Audio transcribing is the process of transforming audio (from videos) to text synchronously to the video timeline. Speech-To- Text (STT) is one of Unravel's core features used with other proposed features, so it was a challenge choosing the right STT tool to use. The best STT libraries and frameworks that allow the application to directly implement the STT functions locally require too much CPU performance and lack accuracy as they are still under development, such as Mozilla's DeepSpeech API and CMUSphinx. The second solution was to use an industrial STT provider, which provides a considerable leap in accuracy and performance. Based on the accuracy of audio extracted from videos not captured professionally, the best operating APIs are Google Enhanced (video) API, Microsoft Cognitive speech, and Amazon STT API. In addition to the accuracy and performance, using an external API provides better scalability; it allows us to design a functional model in the backend that can support plug-and-play any STT provider whenever needed.

According to a recent benchmark for STTs [18], [19], it showed that Amazon Transcribe and Azure STT showed similar performance. Both provide a solid transcription

accuracy with an average Word Error Rate (WER) of less than 9%, which is quite good compared to other cloud platforms. Alongside the accuracy, Transcribe offers a reasonable price rate of approximately 2.4 cents/min (better cost than Azure STT). Therefore, Amazon Transcribe was used as it provides the most balanced cost-benefit trade-off as compared with Azure STT. Amazon Transcribe provides a variety of features to ease the Speech to Text process, such as automatic speaker diarization, punctuation, and formatting that make it compete with human transcription. It can also ensure customer privacy by automatically detecting Personal Identifiable Information (PII) and removing it from text upon request. In this context, Amazon Transcribe is used in the proposed application to transcribe the videos in editing mode with word time indication to allow for swipe to edit video and perform other operations on the generated text.

- **Cloud Storage Service:** In unravel, AWS S3 bucket stores the raw and processed videos for easy integration with Amazon Transcribe, which offers massive cloud object storage for different content (e.g. images, videos, and other file types). AWS S3 bucket also stores the transcription of video speech. It supports our event-driven architecture by running different types of cloud functions based on different file types that get pushed in an S3 bucket (storage unit).
- **Audio separation and Video streaming services:** These two services generally require high computational power. To integrate these services with Amazon Transcribe, Amazon Lambda functions were used to listen on S3 buckets (storage) and update the video status (*Processing, Editable, Edited and Published*). One of the main beneficiaries of Lambda functions is the computational power, which aided us in performing all the editing and cropping of videos by offloading this high-performance computation from user's PCs and the API server.
- **Data Extraction and Classification Service:** Another vital service in unravel is the need for extracting useful information from unstructured data. For instance, videos could be automatically classified according to relevant topics, which could ease the learning process for the student to search for educational videos by relevance. Further, in some contexts, lecturers may mention some sensitive information (e.g. phone numbers, bank numbers, locations, *etc*) by mistake. In other contexts, some videos could contain hate speech (i.e. overall negativity). We have found that Amazon Comprehend can complement the proposed application by classifying videos with respect to relevance, and detecting public identifiable information so that teachers can crop them from the video for security purposes. Besides, it extracts sentiment bias from the full text, so if hate speech is detected, the application could block posting the video. Finally, if the system encountered an overall positivity, the proposed application can thank the teacher for posting such a positive video and reward him/her with free uploads and transcription.

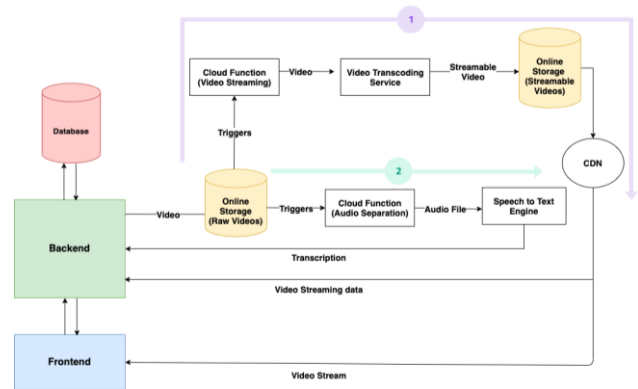


Fig. 2. Abstract system architecture.

#### IV. IMPLEMENTATION AND PERFORMANCE EVALUATION

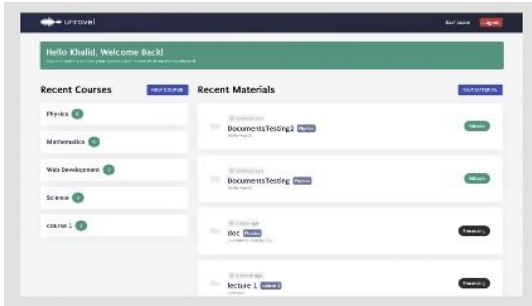
The proposed application consists of various web pages developed using components mentioned in Section III, summarized as follows<sup>2</sup>:

- **Homepage:** The entry point of unravel is composed of different React components that render various parts reusable in different screens, such as buttons, navigation bar, and the logo.
- **Registration Page:** It is responsible for account creation. The page's functionality consists of client-side validation linked with server-side validation to ensure malicious requests do not pose any threat. This page communicates with the backend API to handle the account creation.
- **Login Page:** It is responsible for authenticating the user by sending an HTTP request to grant valid users a JWT token to be used for subsequent requests following the OAuth authentication protocol.
- **Instructors Dashboard:** The main screen for instructors (Fig. 3a) allows them to create/access courses, access/add materials, and view their status. It interacts with the backend in several ways: fetching courses associated with the instructor's account, fetching materials of instructors on press of any of the courses tab, and creating additional courses and materials.
- **Materials (Editing Mode):** The core feature of Unravel is video editing which utilizes speech transcription to offer instructors an editing experience similar to editing documents in editors like Microsoft Word. First, the instructor uploads the video saved in the storage unit (Fig. 3-C), and it is then ready for editing whenever needed. By moving words around and removing words with the cursor, the instructor can edit the content of his materials without spending hours on professional video editing software. The FFMPEG video editing function primarily was benchmarked against 4 different-sized videos (Table I). As depicted in the 4<sup>th</sup> row in Table I, an average-sized 60 mins 1080p lecture video (2648MB = 2.5GB) with a maximum memory of (5300MB = 5GB, half of what AWS Lambda's memory is supporting (10GB)) took 1 min (66000ms) processing time by Lambda. In this context, the application is scalable because it can handle average videos with satisfying performance in the

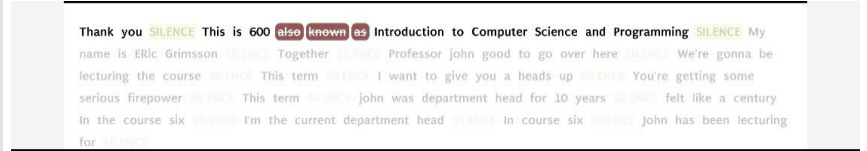
<sup>2</sup> More details related to Unravel-API documentation are available in <https://github.com/UnravelAI>.

consumed time. The interface shown in Fig. 3b depicts the core functionality of Unravel, planned to be as intuitive as possible. It does not require more than a press on the undesired word for removal from the video.

- **Materials' student view (Published Mode):** The main point of contact between students is materials, where a student can view his material and download the supporting documents.



(a) Instructors Dashboard



(b) An example of editing video text. The user just chooses the words to be removed by pressing (highlighted in red).

Fig. 3. Some features of unravel application.

Further, a small-scale case study is conducted to test the approach's applicability. Several tutors have uploaded multiple videos, each ranging from 45-60 minutes (with a total maximum size of 10G). Their processing time was a maximum of 1 minute, which was still manageable in terms of the user experience. Further, several students have shown interest in using Unravel because of the user-friendly interface and having all educational materials on a single educational platform.

## V. CONCLUSION AND FUTURE DIRECTIONS

This paper aimed to solve the main problems of online educational platforms by proposing a single platform that would ease and clarify the process of online learning. The approach aimed to handle the issue of having long lecture videos and hectic video editing tools by providing a seamless video editing tool. The latter allows teachers to view their videos in a document-like representation. The teachers can also edit the videos by deleting/swiping over the text seamlessly. Moreover, the paper addressed the problem of scattered materials for students by providing a platform that is video/lecture oriented and combined with all lecture materials (e.g. documents, notes, etc). In summary, the focus was on providing a single-channel education platform. In particular, unravel contributes to a seamless AI-driven video editing tool using video-text and machine learning techniques. In the future, we aim to add more advanced AI-driven text analysis to facilitate the learning process: 1) *Automatic summarization*: shortening a set of data computationally to create a subset that represents the most important or relevant information within the original content; 2) *Topic detection*: The process of detecting the vital keywords mentioned during the lectures and highlighting them to the students. Topics can be a scientific term that the tutor keeps repeating or stating that is important in another context or a number, a percentage, or a formula; and 3) *Seamless document search*: provides a search engine that scans the documents and the transcribed videos using elastic

TABLE I: EVALUATION OF VIDEO EDITING COMPONENT ON VARIOUS SIZES OF VIDEOS. NOTE THAT THE BILLED DURATION DEPICTS THE DURATION THAT THE LAMBDA FUNCTION TOOK TO PROCESS THE VIDEO

Video file Size (MB)	Max memory used by Lambda Function (MB)	Billed duration (ms)
33.8	187	2600
194.1	521	6200
529.6	1180	13200
2648	5300	66000

search. We also aim to test the proposed approach using web services from other cloud providers, such as Microsoft, Google, and IBM.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

Conceptualization, D.M.S, K.K, A.N, and A.B; data curation, K.K; formal analysis, D.M.S, K.K, A.N, and A.B; investigation, A.N and A.B; software, K.K, A.N, and A.B; validation D.M.S; visualization, A.B; writing and reviewing, D.M.S, K.K, A.N, and A.B. All authors had read and approved the published version of the manuscript.

## ACKNOWLEDGMENT

This project has won the first place in smart eduhackathon 2021 competition and received a grant of 30,000LE, which was sponsored by Africa and Arab Collegiate Programming Championship.

## REFERENCES

- [1] S. Burgess and H. H. Sievertsen, "Schools, skills, and learning: The impact of covid-19 on education," *VoxEu. org*, vol. 1, no. 2, 2020.
- [2] (2021). 1.37 billion students now home as covid-19 school closures expand. [Online]. Available: <https://en.unesco.org/news/137-billion-students-now-home-covid-19-school-closures-expand-ministers-scale-mul>
- [3] (2021). Covid: Dutch go into christmas lockdown over omicron wave. [Online]. Available: <https://www.bbc.com/news/world-europe-59713503>
- [4] N. Bergdahl and J. Nouri, "Covid-19 and crisis-prompted distance education in sweden," *Technology, Knowledge and Learning*, vol. 26, no. 3, pp. 443–459, 2021.
- [5] E. Aboagye, J. A. Yawson, and K. N. Appiah, "Covid-19 and e-learning: The challenges of students in tertiary institutions," *Social Education Research*, pp. 1–8, 2021.
- [6] X. Zhu and J. Liu, "Education in and after covid-19: Immediate responses and long-term visions," *Postdigital Science and Education*, vol. 2, no. 3, pp. 695–699, 2020.

- [7] O. B. Adedoyin and E. Soykan, "Covid-19 pandemic and online learning: The challenges and opportunities," *Interactive Learning Environments*, pp. 1–13, 2020.
- [8] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, "E-learning: Challenges and research opportunities using machine learning & data analytics," *IEEE Access*, vol. 6, 2018.
- [9] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat, "Exploring the critical challenges and factors influencing the e-learning system usage during covid-19 pandemic," *Education and Information Technologies*, vol. 25, pp. 5261–5280, 2020.
- [10] E. R. Carlson, "Covid-19 and educational engagement," *Journal of Oral and Maxillofacial Surgery*, vol. 78, no. 7, pp. 1049–1051, 2020.
- [11] Z. I. Almarzooq, M. Lopes, and A. Kochar, "Virtual learning during the covid-19 pandemic: A disruptive technology in graduate medical education," pp. 2635–2638, 2020.
- [12] T. Favale, F. Soro, M. Trevisan, I. Drago, and M. Mellia, "Campus traffic and e-learning during covid-19 pandemic," *Computer Networks*, vol. 176, p. 107290, 2020.
- [13] M. Tropea and F. Rango, "Covid-19 in Italy: Current state, impact and ict-based solutions," *IET Smart Cities*, vol. 2, no. 2, pp. 74–81, 2020.
- [14] W. Rice and H. William, *Moodle*, Packt Publishing Birmingham, 2006.
- [15] A. Sheshasaayee and M. N. Bee, "Evaluating user experience in moodle learning management systems," in *Proc. 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017, pp. 735–738.
- [16] A. N. Endozo, S. Oluyinka, and R. G. Daenos, "Teachers' experiences towards usage of learning management system: Canvas," in *Proc. the 2019 11th International Conference on Education Technology and Computers*, 2019, pp. 91–95.
- [17] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*, O'Reilly Media, Inc., 2011.
- [18] B. Xu, C. Tao, Z. Feng, Y. Raqui, and S. Ranwez, "A benchmarking on cloud based speech-to-text services for french speech and background noise effect," arXiv preprint arXiv:2105.03409, 2021.
- [19] (2022). Speech-to-text benchmark. [Online]. Available: <https://github.com/Picovoice/>

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



**Dalia M. Sobhy** is working as a lecturer in the Computer Engineering Department at the Arab Academy of Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt. She has been awarded a Ph.D. degree in the School of Computer, University of Birmingham, in 2019. She received the MSc degree in Computer Engineering Department at the Arab Academy of Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt, and a BSc degree from the same university. She was also awarded the gold medal as the top-ranked candidate (1st) in the graduating class. She held a teaching post in AASTMT. Dalia's main

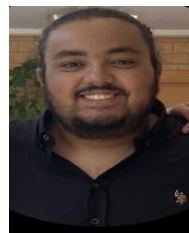
research interests are software architecture, economics-driven software engineering, machine learning for software engineering, search-based software engineering, and cloud and services software engineering. Her focus lies predominantly in the domain of software architectures for dynamic environments, such as IoT. Her work has been published in internationally renowned conferences such as the European Conference of Software Architectures (Springer) and top conferences such as the Journal of Systems and Software (Elsevier) and ACM Transactions on Software Engineering and Methodology (TOSEM). Among other roles, Dalia is a reviewer in IEEE Software Journal and JSS (Elsevier Journal of Software and Systems) manuscripts; a sub-reviewer in IEEE International Conference on Services Computing; Women in Engineering (WIE) chair; and IEEE board member for AASTMT branch.



**Khalid Khalil** is working as a software engineer based in London at Meta in a Frontend-focused role. He is mainly interested in building solid UX and tackling product problems. He received his BSc degree in the Computer Engineering Department at the Arab Academy of Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt. Khalid's main research interests are software engineering and UX. He had several interns: Frontend Engineer in Facebook, React Native Developer in CivilSoft, Software Engineer Intern in eSpace, and UI Designer Intern in Pimula Agency.



**Ahmed Nabil** is working as a software engineer based in Egypt at Incorta. He is mainly interested in building solid UX and tackling product problems. He received his BSc degree in the Computer Engineering Department at the Arab Academy of Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt. Ahmed's main research interests are software engineering and robotics engineering. He had several interns: robotics engineering at the University of Central Lancashire, UK, and Software Developer in Valeo.



**Aly Barakat** is working as a software engineer based in Egypt at Minly. He received his BSc degree in the Computer Engineering Department at the Arab Academy of Science and Technology and Maritime Transport (AASTMT), Alexandria, Egypt. Aly's main research interests are software engineering, mobile development, and robotics engineering. He had several interns: Cross-Platform Mobile Developer in CivilSoft, Software Engineer in CivilSoft, and Programming and Robotics Instructor in Young Leaders Academy.