

# Scheduling Online Oral Assessments Using an Iterative Algorithm: A Profound Software for Educational Continuity

Said Ally<sup>1\*</sup> 

## AFFILIATIONS

ICT & Math Department, The Open University of Tanzania, Dar es Salaam, Tanzania.

## CORRESPONDENCE

Email: [said.ally@out.ac.tz](mailto:said.ally@out.ac.tz)\*

## EDITORIAL DATES

Received: 21 February 2024

Revised: 28 April 2024

Accepted: 30 April 2024

Published: 14 May 2024

## Copyright:

© The Author(s) 2024.

Published by [ERRODF Forum](#).

This is an open access article distributed under Creative Commons Attribution (CC BY 4.0) licence.



DOI: [10.38140/ijer-2024.vol6.15](https://doi.org/10.38140/ijer-2024.vol6.15)

**Abstract:** In response to the socioeconomic effects of the global COVID-19 pandemic, including the disruption of educational continuity, an online oral assessment solution is proposed. While the next pandemic is unpredictable, the need for effective student assessment during lockdown is essential. This paper presents an iterative algorithm for scheduling online oral examinations as a profound software innovation capable of securely assessing students remotely while adhering to COVID-19 safety requirements. The efficacy of the software was tested by university tutors and undergraduate students for 60 days on local and web hosts. The results show that the algorithm offers optimal performance despite extraneous factors such as bandwidth capacity, operating system type, and client-side web browsers. The software manages the systematic random pairing of examiners, sessions, courses, and candidates, as well as the automated generation of question papers with a marking rubric. This invention would largely provide assessment entities with significant pedagogical, financial, and technological benefits while ensuring educational continuity regardless of the pandemic's status.

**Keywords:** OREX, online oral assessment, software algorithm, online examination, COVID-19.

## 1. Introduction

The emergency and widespread COVID-19 pandemic have significantly stimulated software innovation in response to the affected socioeconomic sectors. In the field of education, ensuring continuity in student assessment emerged as a major challenge despite the availability of various online teaching and learning services. Overall, the COVID-19 pandemic has presented significant obstacles to educational delivery (Schleicher, 2020), particularly during periods of lockdown.

Although the majority of schools and universities were impacted (Nicola et al., 2020), many were still able to continue teaching and learning through different forms of online education, including e-learning platforms and audiovisual collaboration tools for video conferencing services (Linden & Gonzalez, 2021). While these online learning facilities proved beneficial for both synchronous and asynchronous teaching modes during the COVID-19 pandemic (Daniel, 2020), the main challenge was how to administer examinations, considering that both tutors and students were geographically dispersed and were held under lockdown. During this period, assessment methods that relied on face-to-face sessions and physical classes were completely prohibited, as they failed to meet COVID-19 safety requirements (Adedoyin & Soykan, 2020). Instead, a remote-based online assessment emerged as the sole alternative solution to ensure educational continuity. This form of assessment was the recommended approach (García-Peñalvo et al., 2021) due to its ability to connect participants from anywhere. For example, the University of Twente (UoT, 2020) and Iowa State University (ISU, 2020) utilised video conferences and telephones for student assessment during the COVID-19 pandemic. Other educational institutions employed zoom-invigilated exams to facilitate online remote assessment (Linden & Gonzalez, 2021). However, not all online assessment methods are suitable or compatible with the COVID-19 pandemic (Weiner & Hurtz, 2017). Online written

### How to cite this article:

Ally, S. (2024). Scheduling online oral assessments using an iterative algorithm: A profound software for educational continuity. *Interdisciplinary Journal of Education Research*, 6, 1-11. <https://doi.org/10.38140/ijer-2024.vol6.15>

examinations, for instance, require students to be physically present in a closed computer lab, highlighting the importance of developing a remote-based invigilation solution (Sefcik et al., 2022).

In response to the academic challenges posed by COVID-19, the Open University of Tanzania developed OREX, an advanced software solution for the administration of online oral examinations (Ally & Oreku, 2022). OREX is the only viable option for the institution, as it ensures compliance with COVID-19 safety requirements. Through OREX, conditions such as social distancing, lockdown measures, and travel restrictions, all imposed by the COVID-19 pandemic, can be effectively managed, as students can be assessed from any location.

Numerous academic domains, including psychology, science, postgraduate programs and medical institutions, the maritime industry, and special education, have extensively utilised oral assessments (OREX, 2020). This paper presents the algorithms employed in the OREX system, along with clearly defined software functionalities, conditions, and a set of constraints necessary for assessing software performance. The proposed solution contributes valuable software knowledge to the existing knowledge base (Wohlin & Aurum, 2015).

## **2. Materials and Methods**

The study addresses a real-world issue (Easterbrook et al., 2008) and is designed as action-based research to tackle the educational challenges presented by the global COVID-19 epidemic. The OREX project was initiated on May 11, 2020, at the Open University of Tanzania (OREX, 2020) in order to replace the installation of video conference facilities as mandated by the university's strategic plans (OUT, 2018). To ensure efficient and secure software, the structural design of the OREX architecture is based on well-defined business processes related to examination administration, as well as software engineering attributes (Ross et al., 1975). In order to enhance clarity, the evaluation of traditional physical oral assessment methods (Friedman, 2003) was a key aspect of the OREX architecture. OREX was also developed as a flexible prototype in order to adapt to rapid technological changes and the non-deterministic challenges of the COVID-19 era. Given that interaction and instant feedback are crucial success factors in software development (Fichman & Kemerer, 1993), a prototyping model (Sabale & Dani, 2012) was chosen.

### **2.1 Development tools and testing plan**

OREX development is based on open-source tools such as Linux OS (Ubuntu 20.04 SE, 2020), MySQL database (8.0), PHP programming language (PHP, 2020), Apache web server (Apache, 2020), and the MVC framework for rapid software development. All tools were hosted in a testing box with the following specifications:

- Processor Type: Intel® Core™ i7-8565 CPU@1.80GHz, 1.99GHz,
- Installed Memory (RAM): 16 GB with 15.7 GB usable, System Type,
- System Type: 64-bit operating system, x64-based processor,
- Operating System: Windows 10 Home 2019

For efficient OREX testing, the following set of activities were completed:

- Development of registration and examination modules.
- User training on the use of OREX software.
- A question bank of 879 courses with a total of 12 questions for type 1 and 36 questions for type 2.
- Server configuration with all necessary underlying software programs and drivers.
- OREX guiding tools: a quick short guide for students, OREX templates for Type 1 and Type 2 oral questions, a short guide for the chief examiner (CE), second examiner (SE), and exam observer (EO), a short guide for audiovisual settings, a marking rubric, and a template for the observer's report.

All participants possessed basic web skills and a smart electronic device with audiovisual capabilities, reliable power, and Internet access.

## 2.2 Algorithm testing

As a web-based and real-time application, OREX was tested intensively for its security and efficiency (Ally, 2014). The software was tested for two months, from June 18th, 2020, to August 14th, 2020, using two testbeds configured as local and webhost machines. The web test was configured at <http://orex.out.ac.tz>, where the server memory was upgraded from 8 to 32 GB of RAM in order to handle increasing concurrent server requests and workloads. Similarly, the Internet speed was increased from 150 to 300 Mbps during the same period (Ally & Oreku, 2022). Table 1 presents important testing parameters for the webhost machine.

*Table 1. Webhost testing parameters*

SN	Data Aspect	Frequency
1	No. of courses	879
2	No. exam sessions	9380
3	No. of students	1994
4	No. of examiners	295
5	No. of question papers	8900
6	No. of days	31
7	Registration period	12 days
8	Examination phase	19 days
9	No. of sessions per day	10 sessions
10	Participation rate	95%

Source: OREX Report (2020).

## 2.3 Performance metrics

The performance metrics of OREX are determined by the randomisation and automation levels in the registration and examination phases. The number of sessions for a specific course or examiner, however, depends on the number of registered students. The performance test is based on various factors, such as the repetition level of pairing aspects, the number of loop iterations after the condition is met, CPU execution time (ns), and the functional programming loop. The pairing aspects in the registration phase include candidate/course (cc), candidate/date (cd), and candidate/session (cs). In the examination phase, the pairing aspects are examiner/candidate (ec), examiner/examiner (ee), examiner/observer (eo), examiner/subject (es), examiner/date (ed), and examiner/session (es).

## 2.4 Control of extraneous factors

For clarity, an optimal performance level of the OREX algorithm was achieved by controlling extraneous factors by configuring a localhost testing machine. The testbed was configured with an Intel® Core™ i7-8565 CPU at 1.80 GHz, 16 GB of RAM, a 64-bit OS system type, and an x64-based processor. The following were the controlled extraneous factors:

- EF1: Internet speed of the client machine,
- EF2: Diversification of Internet browsers among examiners and candidates,
- EF3: OS type of the client machine,
- EF4: CPU speed of the client machine,
- EF5: RAM size of the client machine,
- EF6: Latency factor from client to server.
- EF7: Connect time between user and server, including SSL handshake.
- EF8: Number of examiners per subject.
- EF9: Course capacity, i.e., the number of students in a particular subject.

The first five extraneous factors (EF1-EF5) were completely controlled by the localhost machine. The last two factors (EF8 and EF9) were controlled by limiting the number of examiners and candidates by setting the following criteria:

- Each course must have at least five examiners.
- The class size should range between 100 and 120 students.
- CPU Execution Time (CET) was used instead of Server Response Time (SRT) because the Internet speed was considered an extraneous factor.
- To ensure the computational complexity of the proposed algorithm and the accuracy of the data collection are of high precision, each experiment was repeated three times.

### 3. OREX Structural Design

#### 3.1 Requirement specification

The OREX system enables oral assessments through audiovisual online collaboration tools. The solution saves time and financial resources, as it allows for remote connection with the web audience. The system also facilitates instant feedback, as invigilation and marking are completed during the session. Figure 1 illustrates a typical OREX session with four participants connected via a web link.

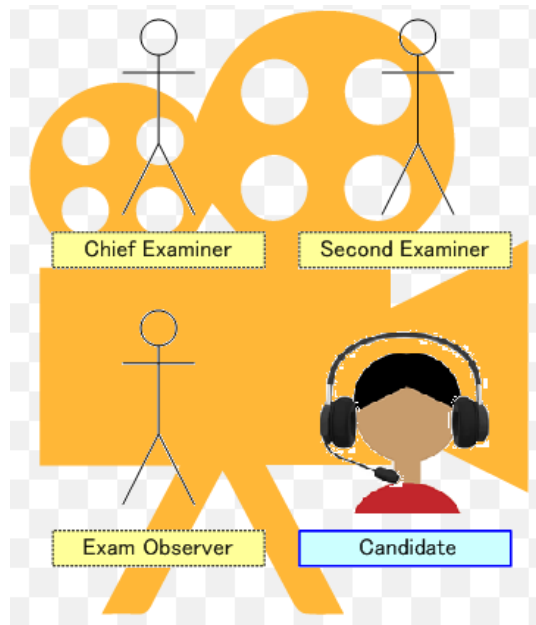


Figure 1: Virtual Exam Hall (i~Room)

The OREX web audience includes the Chief Examiner (CE), Second Examiner (SE), Exam Observer (EO), and candidate. Candidates are uniquely identified by their registration numbers, while CE, SE, and EO are identified by staff numbers. Users are registered based on their assigned roles and privileges and are given access to the virtual link through software.

OREX works by assessing only one candidate in a single session. While CE and SE are subject-specialists, EO can be any academic staff member randomly selected from the pool. The role of each participant is defined in Figure 2.

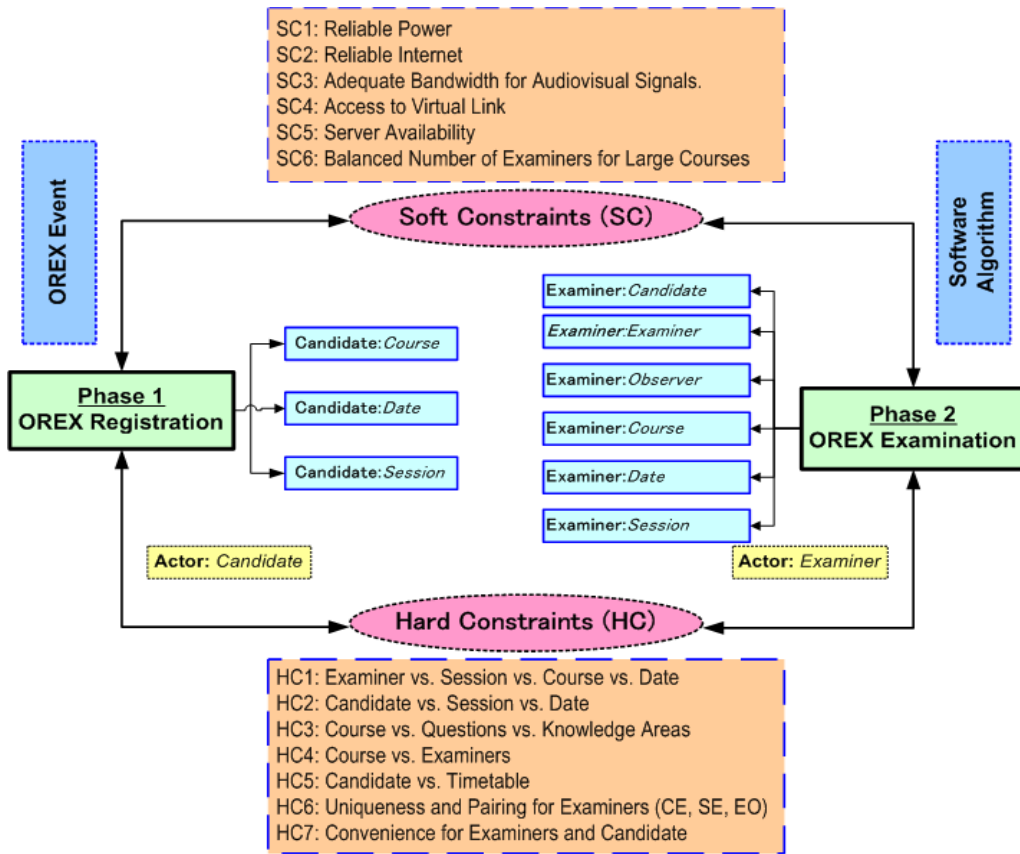


Figure 2: OREX Structural Design

The random pairing of CE/SE/EO is of high anonymity level and is determined by an OREX algorithm based on user workload and convenience. Each day, the OREX live sessions lasting for a maximum of 60 minutes cannot exceed 10 per examiner. The assessment covers all learning outcomes by dividing the course into two parts, each with six comprehensive oral questions. Each question is assigned a unique code in the following format: *CourseCode\_Part\_QuestionSerialNumber*.

The CE ask question 1 while the SE ask question 2. Both examiners are markers of oral responses based on the defined rubric. The question papers and marking schemes are randomly generated and are activated at the time of print (ToP) based on the OREX Visa Card (OVC). The OVC is a timetable printout with the candidate's academic and demographic information. The system computes and saves the final score recorded by CE and SE in an online mark sheet for permanent storage.

As depicted in Figure 2, OREX must satisfy both hard constraints and soft constraints to be feasible. The following set of constraints guided the development of the software algorithms:

### 3.2 Hard constraints

For OREX to attain maximum efficacy, seven hard constraints (HC1 to HC7) are prerequisites.

- a) HC1: Examiner vs. Session vs. Course vs. Date
  - No examiner can be assigned more than one *session* at the same time.
  - No examiner can assess more than one *course* at the same time.
  - For convenience, no examiner is assigned two or more sessions on the same day if the event last long.
- b) HC2: Candidate vs. Session vs. Date

- A candidate cannot sit for two or more sessions at the same time.
- For convenience, no candidate is assigned two or more sessions on the same day if the event last long.
- c) HC3: Course vs. Question vs. Knowledge Area
  - Each course has at least 12 questions with model answers.
  - Each part has six questions.
- d) HC4: Course vs. Examiners
  - Each course has at least two examiners registered in the system.
  - Examiners are subject specialists who can ask and mark.
- e) HC5: Candidate vs. Timetable
  - A candidate has a unique timetable.
- f) HC6: Uniqueness and Pairing of Examiners (CE/SE/EO)
  - The pairing of CE, SE, and EO is randomly defined.
  - Examiners and Candidates are uniquely identified.
- g) HC7: Convenience for Examiners and Candidates
  - Convenience of examiner is determined by the software.
  - Convenience of candidate is self-determined.

### **3.3 Soft constraints**

A total of six soft constraints (SC1 to SC6) were identified as prerequisites for OREX software to work efficiently.

- a) SC1: Reliable Power
  - The web audience must have a reliable power supply.
  - The OREX server must operate with power backup systems.
- b) SC2: Reliable Internet
  - The web audience must have reliable Internet access.
  - The OREX server must operate with reliable Internet access.
- c) SC3: Adequate Bandwidth for Audiovisual Signals
  - The web audience must have adequate Internet bandwidth capable of transmitting audiovisual signals.
  - The OREX server must have adequate Internet bandwidth to accommodate audiovisual computing workloads.
- d) SC4: Access to Virtual Link
  - The web audience must have timely access to a virtual link.
  - The virtual link is accessible from OREX software at the time of print as per the OREX Visa Card.
- e) SC5: Server Accessibility
  - The server is up and running throughout the OREX session and is accessible 24/7 a week.
  - The server is free from hardware defects.
  - OREX software is free from software bugs.
- f) SC6: Balanced Number of Examiners for Large Courses
  - Large courses (those with 500 students and above) must have at least five examiners.
  - The software should be able to detect the examiner's workload from other courses.

### **3.4 OREX Software Algorithm**

The OREX algorithm constantly maintains the automation and randomisation processes for all key software functionalities, including session allocation and question generation. The algorithm controls

the random generation of question papers, mark sheets, and computation of the final score. The algorithms are helpful in automating the examination timetable. The randomisation process considers a set of constraints, logics, and conditions for all pairing aspects as follows:

- Registration phase: candidate/course (*cc*), candidate/date (*cd*), and candidate/session (*cs*).
- Examination phase: examiner/candidate (*ec*), examiner/examiner (*ee*), examiner/observer (*eo*), examiner/subject (*es*), examiner/date (*ed*), examiner/session (*es*).

The following are the set of conditions constrained in the algorithm:

- No examiner is assigned more than one session at the same time.
- No examiner is assigned more than one course at the same time.
- A student cannot sit for two or more examinations in the same session.
- No examiner can assess two or more sessions on the same day if there is a substantial time span of OREX days.
- The first OREX question comes from the first part of the course (knowledge areas 1-3), while the second question comes from the second part of the course (knowledge areas 4-6).
- Each course has at least two examiners who can assess and mark the questions.
- Each student has a unique examination schedule.
- The software uniquely identifies each CE, SE, and EO using the staff personal file (PF) number.
- The software should uniquely identify each candidate using a student registration number.
- The convenience of the examiner is determined by software.
- The convenience of the candidate is decided by the student himself.

The solution consists of five algorithms focused on candidate registration, session allocation, examiner selection, collision free for candidates as well as collision free for examiners and observers.

---

**Algorithm 1: OREX Event Registration Algorithm**

---

Condition 1: The algorithm is constrained within specific OREX registration period.

---

Condition 2: The OREX event is defined in the University Almanac.

---

Condition 3: The OREX event can run as on-demand and it is not in the University Almanac

---

Condition 4: Number of Days can vary:  $d_1, d_2, d_3, \dots, d_n$

---

**Algorithm 2: OREX Session Algorithm**

---

Condition 1: Number of sessions per day is constrained within student/course registration.

---

Condition 2: Maximum number of OREX sessions per day is 10.

---

Condition 3: Session duration is 60 minutes.

---

Condition 4: Number of Sessions can vary:  $S_1, S_2, S_3, \dots, S_m$

---

**Algorithm 3: OREX Examiners Algorithm**

---

Condition 1: Session activation depends on the availability of Examiners.

---

Condition 2: Maximum number of Examiners per session is 2.

---

Condition 3: Chief Examiner (CE) and Second examiner (SE).

---

Condition 4: CE and SE must be chosen from pool of subject experts.

---

Condition 5: Exam Observer (EO) is sourced from the pool of all academic staff

---

Condition 6:  $Examiner\{CE, SE\}$

---

---

**Algorithm 4:** Collision Free for Candidate

---

Condition 1: OREX session allocated to a candidate based on available CE/SE/EO.

---

Condition 2: The system checks for a student collision free session

---

Condition 3: For student's convenience, the nested loops between OREX days and OREX sessions are checked.

- (a) The first session  $S_1$  is initially checked for all days before the second session  $S_2$  is checked.
  - (b) The second session  $S_2$  is checked for all days before the third session  $S_3$  is checked, and so on.
  - (c) Equation 2 is outer loop.
  - (d) Operation Mode:  $S1d1, S1d2, S1d3, \dots S1dn, \dots S2d1, S2d2, S2d3, \dots S2dn, \dots S3d1, S3d2, S3d3, \dots Smdn, \dots$
- 

Condition 4: The algorithm iterates all OREX days for the first session, reiterate again for all days for the second session until when a collision free session is found.

---

Outer Loop: *for* (*int*  $S_i = 1;$   $S_i \leq m;$   $S_i++$ )

---

Inner Loop: *for* (*int*  $d_i = 1;$   $d_i \leq n;$   $d_i++$ )

---

**Algorithm 5:** Collision Free for Examiner and Exam Observer

---

Condition 1: Once the collision free session  $S_i d_i$  is found for a candidate, the system checks for the two examiners  $\{CE, SE\}$ . The system checks for examiner collision free session

---

Condition 2: Examiners (CE/SE) are selected from a pool of course experts. *Examiner* $\{CE, SE\}$

---

Condition 3: If the course is allocated more than two examiners, the system checks for combination or pairing of examiners  $\{CE, SE\}$  *for* ( $d_i S_i$ ).

---

Condition 4: Student collision free is checked: *if* ( $S_i d_i$ ) = = 1

---

Condition 5: Examiners pairing is checked: *if* ( $CE, SE$ ) = = 1

---

Condition 6: Examiners collision free is checked: *if* ( $d_i S_i$ ) = = 1

---

Condition 7: Exam Observer is selected from the pool of all tutors: *Exam Observer* $\{EO\}$

---

Condition 8: Collision free for Exam Observer is checked: *if* ( $S_i d_i$ ) = = 1, *Return* $\{CE, SE, EO\}$

---

### 3.5 Software report production and retrieval

The OREX software produces reports for use at departmental and university levels for exam registration, exam sessions, exam results, and examiner lists. A list of registered students displays information such as the student's full name, registration number, course code, exam date, session time, and confirmation status. The OREX Visa Card (OVC) is simply an examination hall ticket which consists of admission and academic data. Admission data includes student name, student photo, registration number, physical address, exam centre, program name, index number, sponsor, birthdate, and printed date, while academic data includes course code, course name, exam date, and session time. The list of examiners includes staff file number, staff name, course code, role, status, action, web link, and phone number. The system can also generate a list of courses and the number of questions in the database. The question paper comes with a course code, question code, question text, and model answer. The online mark sheet with an inbuilt marking rubric composes score boxes for communication skills, creativity, knowledge and understanding, problem-solving, understanding of learning outcomes, confidence, fluency, and accuracy. The course and student result sheet includes the student registration number, course code, examiner, question, and score awarded. The observer report captures information about students, sessions, infrastructure, and academic profiles. Finally, four basic system profiles include the following:



- Student profile (student name, examination center, registration number, program, birthdate, sponsor, and Form 4 index number).
- Session profile (date of exam, time-in and time-out, course code, and the session link), attendance status of each participant (CE, SE, EO, and candidate), verification of the candidate, if CE/SE has worked independently and with fair treatment, participant's dress code).
- Academic profile (question paper generated, mark sheet, examination irregularity matters, number of follow-up questions asked by examiners)
- Infrastructure profile (Internet accessibility, availability of the OREX server, Internet speed, number of session breaks and disconnects, and the adequacy of physical arrangements in the exam virtual room, including noise and interference).

#### **4 Discussion**

The development of OREX software provided an opportunity for examiners to receive instant feedback from students. Based on the question paper blueprint, the questions prepared were comprehensive, integrated, and randomly generated from the software. Follow-up questions were also used to prompt more elaborate answers from the students. This assessment system has helped students build confidence and evaluate their communication skills, creativity, application, problem-solving abilities, knowledge, and understanding of the learning outcomes. This was made possible due to the high flexibility and greater objectivity of oral assessment compared to written examinations.

A comparative analysis to evaluate the performance level of the algorithm was theoretically conducted using a mathematical technique called big-O notation. The assessment of the computational complexity and efficiency of the software algorithm using run-time analysis shows that the loops behave equally on the server local host and the server web host if extraneous factors (EF1-EF9) are handled properly. For instance, the performance difference between the web host and localhost was insignificant when the internet speed increased from 150 Mbps to 300 Mbps, regardless of the web browser and operating system used in a client computer with minimum computing specifications, such as a 2 GHz CPU and 4 GB RAM. This concludes that the OREX algorithm offers optimal software performance if extraneous factors are taken care of. Since the estimated run time recorded to measure the time spent when user requests are submitted between the client and server depends on the input size, a slight performance difference was observed for big courses with at least 120 concurrent logins. Run time is directly proportional to the input size for both web host and localhost servers when a linear search algorithm is used. In this case, a localhost machine that runs a linear search program exhibits a linear growth rate. On the other hand, a binary search showed high performance in a web host machine when the input size became large. This means that a web host machine running the binary search program exhibits a logarithmic growth rate because the run time increases by a constant amount. Even though a localhost is a faster machine due to no interference from extraneous factors, a web host machine will inevitably surpass a localhost in runtime because it runs an algorithm with a much slower growth rate.

#### **5 Conclusion and Recommendation**

The OREX algorithm is a state-of-the-art software solution that helps maintain educational continuity. Candidates can be assessed from anywhere through an audiovisual online tool. The algorithm controls automation and random selection of examiners, observers, and question paper generation while managing workloads, convenience, and security. Overall, the development of the OREX algorithm surpasses the challenges posed by COVID-19 to the education industry. Higher

learning institutions can adopt this software for efficient management of oral assessments for students participating from anywhere in a cost-effective manner.

## 6. Declarations

**Funding:** This research did not receive any external funding

**Acknowledgements:** I would like to acknowledge the Open University of Tanzania for their support during the entire development of the OREX system. I would also like to thank the programmers in the Directorate of Teaching, Learning, and Examination Services for their technical support.

**Conflicts of Interest:** The author declares no conflict of interest.

**Data availability:** The article did not include data collection, as the author chose to keep the underlying information confidential. However, the system is accessible online through this link: "https://orex.out.ac.tz/index.php/login".

## References

- Adedoyin, O. B., & Soykan, E. (2023). 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>
- Ally, S. (2014). Security vulnerabilities of the web-based open-source information systems: adoption process and source codes screening. *Huria: Journal of the Open University of Tanzania*, 17, 1-13.
- Ally, S., & Oreku, G. (2022). Information system innovative framework for online oral examinations. *International Journal of Education and Development using Information and Communication Technology*, 18(1), 236-252.
- Apache. (2020). Apache HTTP web server project, website. <https://apache.org/>
- Daniel, S.J. Education and the COVID-19 pandemic. *Prospects* 49, 91-96 (2020). <https://doi.org/10.1007/s11125-020-09464-3>
- Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J., Sjøberg, D.I.K. (eds) *Guide to Advanced Empirical Software Engineering*. Springer, London. [https://doi.org/10.1007/978-1-84800-044-5\\_11](https://doi.org/10.1007/978-1-84800-044-5_11)
- Fichman, R. G., & Kemerer, C. F. (1993). Adoption of software engineering process innovations: The case of object orientation. *Sloan Management Review*, 34, 7-7.
- Friedman, K. (2003). Theory construction in design research: Criteria: Approaches, and methods. *Design Studies*, 24(6), 507-522. [https://doi.org/10.1016/S0142-694X\(03\)00039-5](https://doi.org/10.1016/S0142-694X(03)00039-5)
- García-Peñalvo F.J., Corell A., Abella-García V., Grande-de-Prado M. (2021) Recommendations for Mandatory Online Assessment in Higher Education During the COVID-19 Pandemic. In: Burgos D., Tlili A., Tabacco A. (eds) *Radical Solutions for Education in a Crisis Context. Lecture Notes in Educational Technology*. Springer, Singapore. [https://doi.org/10.1007/978-981-15-7869-4\\_6](https://doi.org/10.1007/978-981-15-7869-4_6)
- Iowa State University. (2020). Statement on preliminary and final oral examinations during the COVID-19 response period. <https://www.grad-college.iastate.edu/news/article/updated-oral-exam-statement>
- Linden, K., & Gonzalez, P. (2021). Zoom invigilated exams: A protocol for rapid adoption to remote examinations. *British Journal of Educational Technology*, 52(4), 1323-1337. <https://doi.org/10.1111/bjet.13109>
- MySQL (2020). MySQL database management system (DBMS). <https://www.mysql.com/>
- Nicola, M., Alsafi, Z., Sohrabi, C., Kerwan, A., Al-Jabir, A., Iosifidis, C., ... & Agha, R. (2020). The socioeconomic implications of the coronavirus pandemic (COVID-19): A review. *International journal of surgery*, 78, 185-193. <https://doi.org/10.1016/j.ijsu.2020.04.018>
- OREX (2020). Approved proposal for oral examinations. The Open University of Tanzania Senate Minutes, 11th May 2020.

- OUT (2018). Rolling strategic objectives of 2018/2019 – 2022/2023, strategic objective no. 1, Target 1.1. The Open University of Tanzania.
- PHP (2020). Internet Programming Language (PHP), Website: <https://www.php.net/>
- Ross, D. T., Goodenough, J. B., & Irvine, C. A. (1975). Software engineering: Process, principles, and goals. *Computer*, 8(5), 17-27. <https://doi.org/10.1109/C-M.1975.218952>
- Sabale, R. G., & Dani, A. R. (2012). Comparative study of prototype model for software engineering with system development life cycle. *IOSR Journal of Engineering*, 2(7), 21-24.
- Schleicher, A. (2020). The impact of COVID-19 on education insights from education at a glance 2020. <https://www.oecd.org/education/the-impact-of-covid-19-on-education-insights-education-at-a-glance-2020.pdf>
- Sefcik, L., Veeran-Colton, T., Baird, M., Price, C., & Steyn, S. (2022). An examination of student user experience (UX) and perceptions of remote invigilation during online assessment. *Australasian Journal of Educational Technology*, 38(2), 49-69. <https://doi.org/10.14742/ajet.6871>
- Ubuntu (2020). Ubuntu server edition. <https://ubuntu.com/download/server>
- UoT, (2020). UT framework for Remote Assessment during the COVID-19 crisis. University of Twente, Netherlands
- Weiner, J. A., & Hertz, G. M. (2017). A comparative study of online remote proctored versus onsite proctored high-stakes exams. *Journal of Applied Testing Technology*, 18(1), 13-20.
- Wohlin, C., & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, 20(6), 1427-1455. <https://doi.org/10.1007/s10664-014-9319-7>

**Disclaimer:** The views, perspectives, information, and data contained within all publications are exclusively those of the respective author(s) and contributor(s) and do not represent or reflect the positions of ERRCD Forum and/or its editor(s). ERRCD Forum and its editor(s) expressly disclaim responsibility for any damages to persons or property arising from any ideas, methods, instructions, or products referenced in the content.