KWARA STATE UNIVERSITY, MALETE, NIGERIA

SCHOOL OF POSTGRADUATE STUDIES (SPGS)

An Enhanced Web-Based Examination System using Automated Proctoring with Background Activity Detection

Toyyib Olaitan OLANREWAJU

22/47MCS/017

B.Sc. (COMPUTER SCIENCE)

November 2025

KWARA STATE UNIVERSITY, MALETE, NIGERIA

SCHOOL OF POSTGRADUATE STUDIES (SPGS)

AN ENHANCED WEB-BASED EXAMINATION SYSTEM WITH AUTOMATED PROCTORING WITH BACKGROUND ACTIVITY DETECTION

A (M.Sc.) DISSERTATION SUBMITTED AND PRESENTED

BY

TOYYIB, OLAITAN OLANREWAJU

22/47MCS/017

In Partial Fulfilment of the requirements for the award of Master of Science in Computer Science

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY,

KWARA STATE UNIVERSITY, MALETE

NIGERIA

November 2025

DECLARATION

I hereby declare that this dissertation titled "An Enhanced Web-Based Examination System using Automated Proctoring with Background Activity Detection" is record of my research. It has neither been presented nor accepted in any previous application for a higher degree.

....	....
Toyyib Olaitan Olanrewaju	Signature / Date

APPROVAL

This is to certify that this dissertation by Toyyib Olaitan OLANREWAJU has been read and approved as meeting the requirements of the Department of Computer Science for the award of the degree of Masters (M.Sc.) in Computer Science.

....	....
Dr. R. M. Isiaka	Signature / Date
Supervisor	
....	....
Dr. (Mrs.) R. S. Babatunde	Signature / Date
Co Supervisor	
....	....
Dr. (Mrs.) F. J. Ajao	Signature / Date
Head of the Department	
....	....
Internal Examiner	Signature / Date
....	....
External Examiner	Signature / Date
Professor Olawale Mashood Aliyu	....
Dean, School of Postgraduate Studies (SPGS)	Signature / Date

DEDICATION

In the name of Allah, the Most Gracious, the Most Merciful, I dedicate this work to the Almighty, whose infinite Mercy and Guidance has been my strength throughout this academic journey. Alhamdulillah, through His Grace and Favor, I have been able to overcome the challenges and reached this point.

I dedicate this research to my esteemed supervisor, Dr. Rafiu Mope Isiaka, who has also served as a director at my workplace, my lecturer, and an academic mentor. His guidance, wisdom, and support have been instrumental in the successful completion of this work. I am deeply grateful for his unwavering encouragement and invaluable contributions throughout this academic journey.

I also dedicate this dissertation to my beloved family and dear friends. I express my heartfelt gratitude to my parents, Olanrewaju Adebimpe and Sijuola Gambari, whose continuous prayers, encouragement, and reminders of patience and perseverance have been invaluable. My brother, Muhammed Jamiu, and my sisters, Balikis and Nimatallahi, have always been by my side, offering unwavering support. May Allah reward them abundantly.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Dr. Rafiu Mope Isiaka, and my co-supervisor Dr. (Mrs.) Ronke Seyi Babatunde for their generosity in sharing expertise and for dedicating their valuable time throughout this research journey. Your unwavering guidance, support, and encouragement have been instrumental in shaping this work. I also extend my sincere appreciation to our Head of Department, Dr. (Mrs.) Falilat Jumoke Ajao, for her countless hours of reflection, reading, and her immense patience, which were vital throughout this process.

Special thanks are due to my other distinguished lecturers Prof. Kazeem Alagbe Gbolagade, Dr. Akinbowale Nathaniel Babatunde, Dr. Sulaiman Olaniyi Abdulsalam, Dr. (Mrs.) Shakirat Ronke Yusuff, Dr. Akeem Femi Kadri, Mr. Shuaib Babatunde Muhammed, Mrs. Oluwasayo Ekundayo, Mr. Habeeb Sulaiman for their wholehearted support, insightful guidance, and willingness to serve. Their contributions have been invaluable.

I would also like to acknowledge and thank the Department of Computer Science for granting me the opportunity to undertake this research and for providing the necessary resources and support. A special mention goes to the staff members of Computer Science Departments for their continued encouragement.

Finally, I extend my appreciation to the early-career lecturers, mentor-lecturers, and administrators in the university community who assisted me with this project. Their enthusiasm, feedback, and dedication made the completion of this research a rewarding and enjoyable experience.

TABLE OF CONTENTS

Pages

LIST OF FIGURES

Pages

LIST OF TABLES

Pages

LIST OF ACRONYMS/ABBREVIATIONS

Pages

Abstract

The proliferation of online education necessitates well-conditioned mechanisms to ensure academic integrity during remote examinations. This dissertation presents the development of an enhanced web-based examination system featuring integrated, multi-modal automated proctoring capabilities designed to detect cheating. The system provides a platform for educators to create and manage objective, subjective, and practical assessments, while students undertake exams within a monitored environment. Key security features include user registration and login incorporating facial recognition via the DeepFace library for identity verification. During examinations, the system employs automated proctoring techniques including continuous analysis of webcam feeds to detect mobile phone usage, the presence of unauthorized individuals, and anomalous head/eye movements. Concurrently, browser focus tracking monitors navigation away from the test window, and background audio analysis using Voice Activity Detection (VAD) and Root Mean Square (RMS) energy flags suspicious noise levels or voice communications. Potential proctoring violations identified through these visual, auditory, and browser-based checks are systematically logged for real-time review. All the artificial intelligence and automated proctoring models were implemented within a Flask-based development environment, seamlessly integrated with a web browser interface to facilitate deployment and user interaction. The developed prototype demonstrates the successful integration of diverse proctoring modalities within a functional online testing framework, offering a comprehensive approach to reducing academic dishonesty in remote assessments and contributing to the maintenance of credible academic standards in digital learning environments. The system achieved accuracy of 72% in functional testing results and 90%+ on other detection mechanisms. Further development focuses on enhancing AI model sophistication, optimizing system performance, and ensuring user fairness.

Keywords: *Voice Activity Detection, Background Activity Detection, Server-Sent Events, Root Mean Square, Audio Cheating, You Only Look Once.*

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Traditional methods of administering examinations required students to be physically present in classrooms under the supervision of human invigilators to deter and prevent cheating (Shilpa *et al.,* 2023). However, with the rapid advancement of technology and the increasing adoption of virtual learning platforms, online examinations have become more common. This shift has introduced new challenges to maintaining academic integrity, prompting the development of Automated Proctoring Systems (AiAPS). One crucial component of such systems is the ability to analyze audio environments in real-time, where Voice Activity Detection (VAD) has found relevance due to its application in fields such as home care (Van Hengel and Anemüller, 2019), surveillance (Kotus *et al.,* 2019), environmental monitoring (Stowell *et al.,* 2020), and urban traffic control (MMeucci *et al.* 2020).

Despite these technological advancements, existing automated proctoring systems often fall short in detecting and interpreting background audio events that may indicate malpractice. While extensive research has addressed issues such as multiple face detection, screen violation, and eye movement monitoring (Tweissi *et al.,* 2022), there remains a significant gap in detecting subtle background activities like whispers, overlapping speech, or coded audio signals. These auditory cues, if undetected, pose a threat to examination integrity (Rahardjo *et al.,* 2024). To bridge this gap, there is a need for systems that can assess environmental audio conditions mine whether an examination environment is conducive to fair assessment (V *et al.,* 2023; Amer-Yahia, 2022; Niveditha et al, 2023).

To address this problem, the research work uses Voice Activity Detection (VAD) to monitor environmental sounds during online examinations. It classifies the setting as either conducive or disruptive based on detected noise. If suspicious audio is identified, the system flags the environment as unsuitable; otherwise, it allows the exam to proceed. This simple yet effective approach helps to uphold integrity in remote assessments without relying on complex or invasive tools (Waleed *et al.,* 2023).

The motivation for this research lies in contributing to the growing body of work that aims to preserve academic integrity in the digital age. As educational institutions continue to integrate technology into learning and assessment, there is a critical need for systems that uphold fairness and trustworthiness. By focusing on auditory monitoring through Voice Activity Detection, this study aspires to set a precedent for audio-aware proctoring solutions and to support the broader goal of making learning adaptive, secure, and accessible. The envisioned system not only mitigates the risk of cheating but also supports a serene examination environment, which is vital for genuine performance evaluation.

Above all this study intends to enhance the reliability of automated proctoring systems by embedding an audio-based verification mechanism capable of detecting environmental disruptions. This idea not only helps to curb examination malpractice but also aligns with the broader educational objective of promoting personalized, adaptive learning environments. By addressing a previously overlooked aspect of proctoring with background activity detection, this research contributes to a future where remote learning and assessment are both secure and equitable.

1.2 Statement of the Problem

The rapid expansion of online learning and assessment has revolutionized education, offering unprecedented flexibility and accessibility (Dawson, 2024). However, this transition brings significant challenges in upholding academic integrity during remote examinations (Holden *et al.,* 2021; Tweissi *et al.,* 2022). Online exams are increasingly vulnerable to impersonation, unauthorized assistance, and illicit collaboration, all of which undermine the validity and fairness of assessment outcomes (Vegendla & Sindre, 2019; Dawson *et al.,* 2024). Existing solutions often present drawbacks such as manual remote proctoring, while providing human oversight, is resource-intensive, costly to scale, and can still be intrusive or circumvented (Hussein *et al.,* 2020). Conversely, basic automated checks may not be sophisticated enough to detect diverse or subtle cheating tactics effectively (Sokurenko, 2025).

Therefore, a critical problem exists in ensuring the credibility of online examinations. There is a clear need for more advanced, reliable, and scalable automated proctoring solutions that can comprehensively monitor the testing environment through multiple modalities (visual, auditory, system interaction) in real-time to effectively detect academic dishonesty, thereby ensuring the credibility and fairness of online examination outcomes.

1.3 Aim and Objectives

Develop an enhanced web-based examination system that integrates automated proctoring and background activity detection techniques. The objectives are to:

i. adapt voice activity detection (VAD) with noise monitoring, for real-time background activity detection;
ii. implement and integrate background activity detection with automated proctoring functionalities system; and

iii. evaluate performance of the integrated proctoring system using web server performance, latency & throughput, resource utilization, delay and logging metrics.

1.4 Scope of the Study

This dissertation is centered on the development of a functional web-based examination platform using the Flask framework. It specifically includes the integration of automated proctoring modules: facial recognition (DeepFace) for identity checks, visual analysis (detecting phones, extra persons, movement via logged data), browser focus tracking, and background audio analysis using Voice Activity Detection (VAD) and noise level monitoring. The work further encompasses the creation of database logging for proctoring events and an evaluation focused on the system's performance metrics like responsiveness and event logging latency, rather than large-scale user trials or comparative analysis with commercial systems. The system will classify the exam environment as permitted or non-permitted using sound classification models and provide real-time alerts or post-exam reports to flag violations.

1.5 Significance of the Study

This research is significant as it tackles the critical challenge of maintaining academic integrity in the growing field of online assessments by developing and demonstrating a practical solution. It contributes by showcasing the integration of multi-modal automated proctoring techniques (visual, auditory, browser-based) within a single platform, offering a more comprehensive approach to deterring and detecting academic misconduct than single-method systems. The project provides valuable insights into the feasibility and performance aspects of such integrated systems, delivering a prototype that can inform institutions and serve as a foundation for future developments in secure remote examination technology.

1.6 Dissertation Outline

This study is structured into five chapters, each addressing a key aspect of the research. Chapter One introduces the work by presenting the background, aim, objectives, scope, significance, and expected outcomes. Chapter Two reviews existing literature on voice detections mechanisms and proctoring systems, examining past efforts, current developments, and gaps in knowledge, while also providing conceptual insights, related works, and a synthesis of the reviewed materials. Chapter Three outlines the research methodology, detailing the theoretical framework, systems employed, and the integration of artificial intelligence to achieve the study's objectives. Chapter Four discusses the results, presenting system outcomes with illustrative examples and visual evidence, followed by a critical analysis and interpretation of the findings. Finally, Chapter Five provides a summary of the research, draws conclusions based on the results, and offers recommendations for future improvements and further development.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview of Proctor System

The concept of a proctor application system involves leveraging computerized systems to improve decision-making processes, conducting online exams, and efficiently monitoring student evaluations. These systems, such as the VIKOR method-guided application for determining the best type of monitoring examinations (Dinda *et al.,* 2018), online exam surveillance applications with proctoring features (Baso *et al.,* 2023), and an Online Examination Portal with face recognition capabilities for live proctoring (Hemant *et al.,* 2022), are designed to streamline exam procedures, prevent misconduct, and save time in assessment processes.

Additionally, systems like proctor support student evaluation and monitoring in courses, providing management efficiency, quality control, and improved staffing in educational settings (Baso *et al.,* 2023). Furthermore, advancements in multi-modal biometric verification, such as the proposed CBUA-OE system, offer enhanced security and accuracy for live online authentication processes, addressing challenges associated with remote proctoring during the pandemic (Purohit *et al.,* 2022).

2.2 In-Person Proctoring

The traditional method of in-person proctoring has long been used to maintain examination integrity. However, advancements in technology have given rise to remote proctoring solutions as viable alternatives (Elshafey *et al.,* 2021). Studies have compared in-person proctoring with remote webcam services, uncovering potential differences in learning outcomes and exam performance (Harmon *et al.,* 2017; Dendir & Maxwell, 2020).

The shift to remote proctoring, accelerated by the COVID-19 pandemic, revealed flaws in fairness, data privacy, and usability (Elshafey *et al.,* 2021). To address the limitations of in-person proctoring, automated AI-based systems with features like face detection, multiple-person detection, and head pose estimation have been developed (Potluri *et al.,* 2023). Studies have also shown that proctor presence can reduce careless responses and improve data quality in online evaluations, with in-person proctoring demonstrating certain advantages over un-proctored methods (Francavilla *et al.,* 2019).

2.3 Live Online Proctoring

Live online proctoring is a crucial aspect of modern education systems, especially with the shift towards online and remote learning. Various research papers highlight the development and implementation of automated AI-based proctoring systems, such as the 'Attentive system,' which utilizes face detection, multiple person detection, face spoofing, and head pose estimation to monitor examinees in real-time Tejaswi *et al.* (2023). These systems aim to ensure academic integrity by detecting and flagging any suspicious activities during online exams, providing instructors with the tools to maintain fairness and reliability in the evaluation process Satre *et al.* (2023).

Moreover, systematic literature reviews have identified different types of proctoring methods, including fully live online, recorded & reviewed, and fully automated, showcasing the evolution of anti-cheat techniques and tools over the past five years Fatima *et al.* (2022) By leveraging technologies like AI for face detection and spoofing detection, live online proctoring systems play a vital role in minimizing malpractices and upholding the credibility of online examinations (Sri Raaghav *et al.,* 2022).

2.4 Recorded Proctoring

Recorded proctoring is a method of online assessment where students record themselves during exams for later review (Chun *et al.* 2022; Fatima *et al.* 2022). This approach aims to enhance security and prevent cheating during high-stake assessments in e-learning environments Fatima *et al.* (2022). Various types of proctoring methods exist, including fully live online, recorded & reviewed, and fully automated proctoring systems (Fatima *et al.*, 2022).

Studies have shown that implementing recorded proctoring mechanisms can effectively detect suspicious events without impacting exam outcomes, with background noise being a common issue (Schoenmaker's *et al.*, 2022). Additionally, the development of intelligent online proctoring systems utilizing artificial intelligence technologies has been proposed to monitor online exams and ensure exam integrity in educational settings (Jia; He 2022). These systems analyze various behaviors such as facial expressions, eye movements, and speech to maintain exam security and fairness in online learning environments (Jia & He, 2022).

2.5 Automated Proctoring (AI Proctoring)

Automated proctoring, also known as Artificial Intelligent proctoring, is a cutting-edge solution that leverages artificial intelligence and computer vision technologies to monitor online exams effectively, ensuring academic integrity and preventing cheating (Shilpa *et al.,* 2023; Yamuna *et al.,* 2023; Sharma *et al.,* 2023) the shift towards online education accelerated by the COVID-19 pandemic, the need for reliable online exam proctoring systems has become paramount (Kulshrestha *et al.,* 2023). These systems offer features like real-time monitoring, customizable alerts for suspicious behavior, and the ability to detect cheating attempts through various means such as eye gaze tracking, head pose estimation, and lip movement analysis (Shilpa *et al.,* 2023; Yamuna *et al.,* 2023; Sharma *et al.,* 2023).

By eliminating the need for physical testing facilities and providing a secure and user-friendly interface for both instructors and examination takers, AI proctoring systems are revolutionizing the examination process, ensuring fairness, and upholding academic standards in remote learning environments. Figure 2.1 and Figure 2.2 show various activities surrounded by Automated Proctoring System.

2.5.1 Student Operations on Automated Proctoring System

Login for the examination, the student starts by logging into the proctoring system using their credentials, ensuring they access the correct examination portal.

System compatibility check before the exam begins, the system checks the student's device for compatibility, verifying necessary software, hardware, and internet connection are all functioning properly.

Student's face input collection the system captures the student's facial input using the device's camera to authenticate their identity and ensure that the registered student is the one taking the exam. Begin the examination once the setup is complete and the student is authenticated; they can start the examination. The system presents exam questions and records their responses.

While writing the examination on the server-side check for cheating and violations as the student writes the exam, the server side of the proctoring system actively monitors any signs of cheating or violations, such as unusual movements or unauthorized materials, using various detection algorithms and techniques. Show examination results based on trust score and total score after completing the exam, the system evaluates the student's performance, factoring in both their total score on the exam and their trust score, which reflects their adherence to exam rules and absence of suspicious behavior. The result is then displayed to the student.

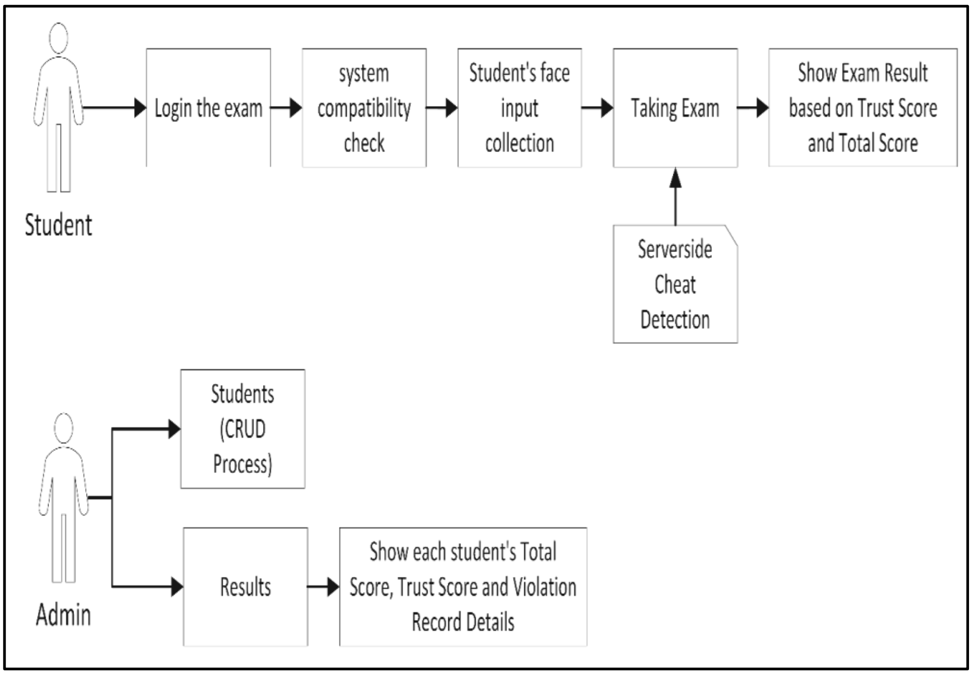


Figure 2.1: Process Flow Diagram Illustrating the Interaction Between the Two Actors in an Automated Proctoring System *(*Tweissi *et al., 2022).*

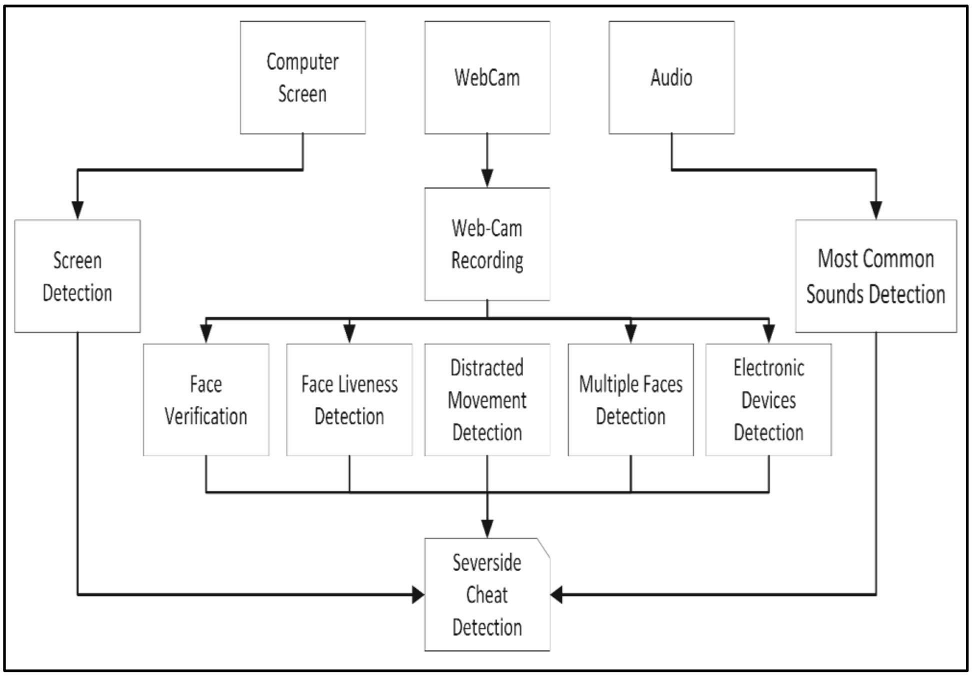


Figure 2.2: Server-Side Activity Monitoring in the Automated Proctoring System *(*Tweissi *et al.,* 2022*).*

2.5.2 Administration Operations on Automated Proctoring System

Check students' CRUD operations admins have the capability to Create, Read, Update, and Delete (CRUD) student records within the Proctored system. This allows administration users to manage student information efficiently.

Check results administration can access and review examination results. This includes displaying each student's total score, trust score, and detailed violation records, enabling thorough evaluation and monitoring of student performance and adherence to examination rules.

2.5.3 Server Cheat Detection

Computer Screen to Screen Detection: During the examination, the proctoring system continuously monitors the student's computer screen to ensure that only authorized exam content is displayed. It detects attempts to switch screens or access unauthorized applications and websites, flagging any suspicious activity for further review.

Webcam to Web-Cam Recording: The system uses the student's webcam to record the exam session. This includes face verification to confirm the student's identity and face liveness detection to ensure that a real person is present. Additionally, it monitors distracted movements, the presence of multiple faces, and the use of electronic devices, detecting and reporting any anomalies that suggest cheating or violations.

Audio to Most Common Sound Detection: The proctoring system employs audio detection to listen for common sounds that could indicate cheating, such as whispered conversations, rustling papers, or electronic device alerts. This helps ensure the integrity of the examination environment by identifying and flagging any unusual or unauthorized sounds.

2.6 Hybrid Proctoring

Hybrid proctoring involves the integration of various technologies and methods to ensure secure and efficient monitoring of individuals during assessments or evaluations. This approach combines different modalities such as face recognition, action recognition, and biometric verification to enhance the accuracy and reliability of the proctoring process (Purohit *et al.,* 2021; MMukhanbet *et al.,* 2021). Automated proctoring solutions have become increasingly customizable, catering to online, hybrid, and face-to-face instruction scenarios, while also addressing concerns about exam integrity and cost allocation (Solis *et al.,* 2023).

However, the use of hybrid procedures extends beyond proctoring, as seen in the collaboration between cardiothoracic surgeons and interventional cardiologists for complex therapies, highlighting the importance of teamwork and innovative thinking in achieving successful outcomes in hybrid programs (Holzer *et al.,* 2016). By leveraging a combination of technologies and interdisciplinary cooperation, hybrid proctoring offers a comprehensive solution for ensuring the security and integrity of assessments in various educational and professional settings.

2.7 Lockdown Browsers

During the COVID-19 lockdowns, browsing behavior significantly changed, with increased internet activity observed among both men and women (Miller *et al.,* 2023; Giardili *et al.,* 2023). Gender disparities emerged in browsing patterns, with women showing reduced online job searches, potentially impacting their employment opportunities (Miller *et al.,* 2023). In Kenya, total daily browser usage increased post-lockdown, with notable gender differences in content consumption, such as women spending more time on YouTube and Netflix compared to men (Giardili *et al.,* 2023).

Additionally, innovative methods like browser locking techniques were developed to enhance security during increased online activities, ensuring stricter settings for active content within browsers (Li Wenjing, 2016; David *et al.,* 2012). These findings underscore the importance of understanding digital behavior shifts during lockdowns and implementing measures to safeguard online activities and data security.

2.8 Mobile Proctoring

Mobile proctoring is a modern solution that addresses the challenges posed by traditional exam settings, especially during the COVID-19 pandemic. The concept involves using mobile applications to facilitate contactless submission of paper-based exams, enabling students to view questions, write solutions on paper, and then scan and upload their answers for grading (Yaghi *et al.,* 2022).

Additionally, proctoring systems like "Pratirakshak" have been developed specifically for online exams, incorporating features such as mobile phone detection, person verification, and monitoring of test-takers' eye and head movements to prevent cheating and ensure fairness in evaluations (Ranka *et al.,* 2023). Furthermore, the feasibility of virtual proctoring through audio-video communication has been demonstrated in complex medical procedures, showcasing the potential for remote expert guidance and knowledge sharing in various fields, including interventional cardiology (Ungureanu *et al.,* 2023).

2.9 Theory of Proctor Systems

Proctor systems play a crucial role in various domains, ranging from datacenter management (Kannan *et al.,* 2018) to online proctoring for exams Raj *et al.*, 2015) and even in the field of pediatric surgery for training purposes (Murakami *et al.,* 2023). These systems are designed to detect performance intrusions, identify root causes of performance variations, monitor behavior during online exams, and provide onsite coaching and supervision for surgical procedures.

Proctor systems utilize techniques such as statistical analysis, machine learning models, image processing, audio processing, and behavior logging to achieve their objectives. By leveraging real-time analytics, lightweight approaches, and scalable frameworks, Proctor systems enhance performance outcomes, improve quality-of-service, diagnose anomalies, prevent cheating, and facilitate the dissemination of advanced skills in various fields. The integration of these systems is essential for ensuring efficiency, accuracy, and competency in complex environments where performance monitoring and supervision are critical.

2.10 Computer-aided Instruction

The Proctor system is an interactive tool for managing and monitoring students in Personalized System of Instruction (PSI) courses. It includes two modules: Tester, which handles student interactions such as quiz administration, scoring, and record-keeping, and Editor, used by instructors to modify student records and course materials. Students' progress at their own pace, taking unit tests and receiving feedback, while instructors can adjust course content and materials (Charles *et al.,* 2022).

PSI emphasizes mastery of written content over lectures, with students progressing through lessons at their own pace and using behavioural objectives to guide their learning. While PSI offers personalized learning, it requires significant preparation of supplementary materials like study guides and alternative tests. However, resources like PSI-specific textbooks and national clearinghouses help mitigate these challenges (Johnson & Ruskin, 2021; Robin, 2020; Sherman & Ruskin, 2021).

The Proctor system streamlines PSI course management by automating processes. Tester allows students to interact with the course and track progress, while Editor gives instructors control over course materials and records, preventing conflicts between the two modules.

This system simplifies managing PSI-based courses and adjusting course content as needed (Charles *et al.,* 2022).

2.11 Privacy Protection and Human Disturbance Quantification

The Acoustic Voice Activity Detection (ecoVAD) model, a convolutional neural network designed for human speech detection in eco-acoustic data, was trained using augmented datasets that combined human speech with background ecosystem noises. The model demonstrated strong performance in detecting speech, achieving high accuracy for male, female, and child voices within a 10-meter range (mean confidence > 0.8). As the distance increased to 20 meters, detection accuracy dropped (mean confidence around 0.7), though the speech was nearly inaudible and unintelligible at this range. The model maintained low confidence (mean confidence 0.30) for non-speech audio, ensuring it could distinguish between speech and background sounds.

ecoVAD was tested in two ecosystems in Børsa, Norway, where it detected speech up to 20 meters, showing its effectiveness in large-scale biodiversity monitoring while addressing privacy concerns. In comparisons with other voice detection models, ecoVAD outperformed existing systems with higher F1 scores (0.917 vs. 0.890 for pyannote and 0.876 for WebRTC VAD). Additionally, long-term passive recordings from Bymarka, Norway, revealed ecoVAD's capability to identify speech patterns linked to peak traffic times, offering valuable insights into human activity and demonstrating its potential for fine temporal resolution in ecological studies. This ability to anonymize data enhances privacy while enabling large-scale studies of human-wildlife interactions.

2.12 Neural Vocoders

Vocoders are key in Text-To-Speech (TTS) and Voice Chat (VC) systems, converting Mel-spectrograms into audio waveforms. However, the process of converting audio to Mel-spectrograms causes information loss, making waveform recovery challenging. Recent advances in deep neural network-based vocoders have improved training efficiency and synthesis quality. These vocoders fall into three main types: autoregressive models, diffusion models, and GAN-based models.

2.13 AI synthetic

The proposed synthesized voice detection framework goes beyond binary classification by incorporating a vocoder identification module with a multi-class classification loss. This module guides the feature extraction network to prioritize detecting vocoder artifacts, which illustrates the activities of voice in the Voice Activity Detection (VAD) framework. The detection of synthetic voices, which can be misused, has become increasingly important, and various methods have been explored to tackle this challenge (Wu *et al.,* 2015; Patil & Kamble, 2018).

Early approaches like bi-spectral analysis identified inconsistencies in the local phases of synthetic voices, which differ from the random local phases of real human voices due to sound transmission and reflection (AlBadawy *et al.,* 2019). Other methods, such as DeepSonar, analyze network responses to audio signals to detect synthetic audio (Patil & Kamble, 2018). The ASVspoof Challenge 2021 has also advanced synthetic voice detection techniques, with key baseline algorithms like CQCC-GMM, LFCC-GMM, LFCC-LCNN, and RawNet2 showing reliable performance in identifying synthetic voices (Todisco *et al.,* 2019).

2.14 A Voice-Based Real-Time Emotion Detection Technique

The proposed method combines Bag of Audio Words (BoAW) for feature extraction with a Recurrent Neural Network (RNN) to classify emotions in audio conversations. BoAW, inspired by the Bag of Words (BoW) technique in Natural Language Processing (NLP), generates fixed-size audio embeddings that capture essential features of the audio.

These embeddings, along with low-level descriptors (LLDs) extracted using openSMILE, allow the model to handle varying utterance lengths. The RNN models both short- and long-term context, and with the support of a codebook of frequent feature vectors, a Bidirectional RNN with attention is used to predict emotions from the conversation.

The model operates in three phases: feature extraction, BoAW embedding generation, and emotion extraction. In the first phase, 130 low-level features are extracted every 10 milliseconds of an utterance. The BoAW component then creates a term frequency matrix from these features, generating rich embeddings that help improve accuracy by bridging the gap between text and audio representations. In the final phase, emotion extraction is performed using the RNN, which utilizes the generated embeddings to predict six basic emotion categories.

2.15 Eliminating Unwanted Signals in Sound by Using Digital Signal Processing System

The proposed solution uses digital filters to remove noise from signals. The process begins by analyzing the original signal to determine the number and type of filters needed based on the noise characteristics, such as low-pass filters (LPF) for low-frequency noise and high-pass filters (HPF) for high-frequency noise. The focus is on digital filters like Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters due to their affordability and easy implementation.

FIR filters are stable and described by the equation:

$$y(n)=bo\ x(n)+b1\ x(n-1)\ +b2\ x(n-2)\ ....\ +bM\ x(n-M) \quad (1)\ Eqn.$$

$$-a1y\ (n-1)-a2y\ (n-2)-\cdots-aNy\ (n-N) \quad (2)\ Eqn.$$

$$y(n)=\Sigma bk\ x(n-k)\ -\Sigma al\ y(n-l)\ Nl=1Mk=0 \quad (3)\ Eqn.$$

Where x: is the input signal, so y: the output signal.

The constant bk, k=0,1,2,3,4,5….M, al, l=1,2,3,4,5……N are called the coefficients. The filter will be design based on three main factors: identifying filter type, the values of cut off frequencies, and the number of poles and zeros.

The algorithm designs the filter based on the filter type, cut-off frequencies, and the number of poles and zeros. The system uses Digital Signal Processing (DSP) to adaptively filter out unwanted noise from audio signals. Depending on the characteristics of the noise, one or multiple filters are applied, such as LPF, HPF, bandpass (BPF), stopband (SPF), or notch filters, to restore the original signal.

2.16 Overview of Voice Activity Detection (VAD)

Voice Activity Detection (VAD) is a critical component in speech processing applications, tasked with distinguishing between speech and non-speech segments in audio signals. This process is essential for enhancing the efficiency and accuracy of speech recognition systems by reducing the computational load and improving robustness against background noise. VAD techniques have evolved over the past fifty years, with various algorithms developed to address the challenges posed by diverse acoustic environments. The following sections provide an overview of the key aspects of VAD, including methodologies, features, and recent advancements.

Traditional VAD methods often rely on energy and spectral entropy calculations to differentiate between speech and non-speech frames. These methods, however, can be sensitive to environmental noise and require careful threshold setting (Jizhong, 2019). Recent advancements include the use of Convolutional Recurrent Neural Networks (CRNNs), which leverage deep learning to classify audio features such as melspectrograms, achieving high precision in detecting speech, music, and pauses (Aguiar-Pontes & Intriago-Pazmiño, 2023).

Unsupervised methods, like the updated K-Means clustering algorithm, offer advantages by not requiring pre-training, making them adaptable to new audio streams without prior knowledge (Sharma *et al.,* 2021).

Various features are used in VAD, including power, harmonicity, and modulation, each targeting different speech properties. The choice of features can significantly impact the performance and complexity of VAD systems (Graf *et al.,* 2015). Temporal context is crucial, as it affects latency and the ability to accurately detect speech in real-time applications (Graf *et al.,* 2015).

Developing VAD systems that perform well in unseen environments remains a significant challenge. The need for robust algorithms that can handle diverse acoustic conditions is a driving force in current research (Patil & Patil, 2024). The integration of advanced machine learning techniques presents opportunities for improving VAD accuracy and adaptability across various applications (Patil & Patil, 2024; Aguiar-Pontes & Intriago-Pazmiño, 2023). While VAD has seen substantial progress, challenges such as handling diverse noise conditions and achieving real-time processing remain. The exploration of new features and machine learning models continues to be a promising area for enhancing VAD performance and applicability in real-world scenarios. Figure 2.3, Figure 2.4, and Figure 2.5 show the illustration of voice activity detection (VAD).

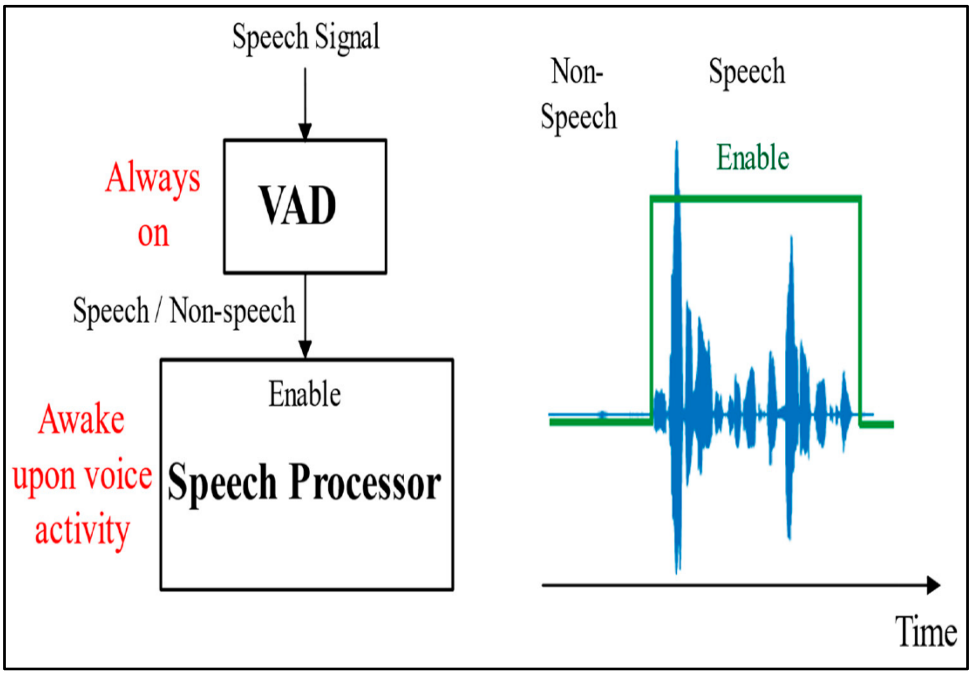


Figure 2.3: Always-on Voice Activity Detection as a Wakeup Mechanism

(Faghani *et al., 2023)*.

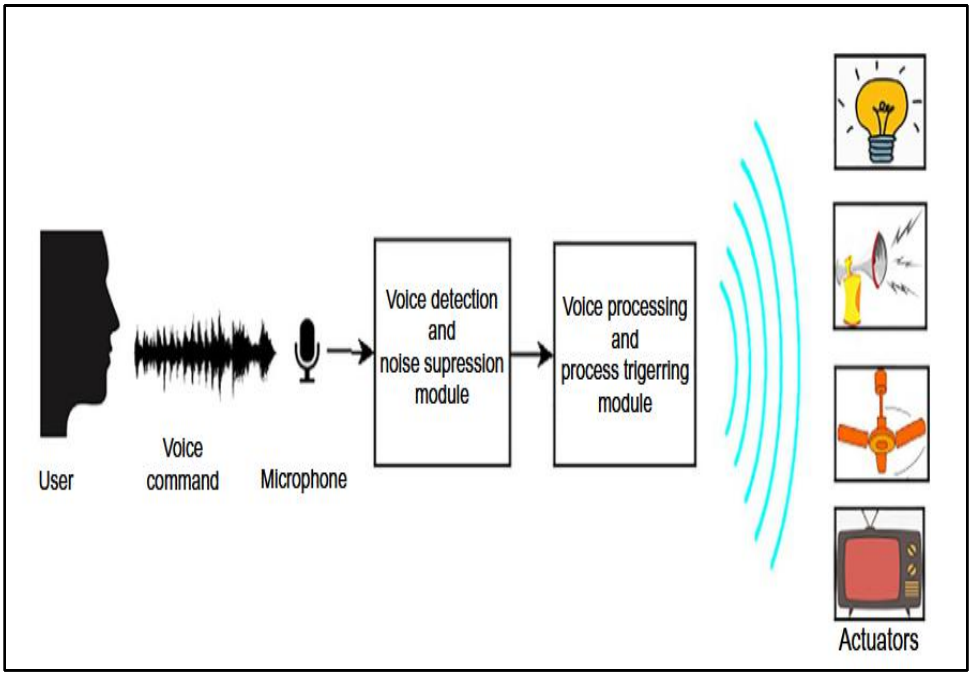


Figure 2.4: Conceptual Design of the VAD-based Automation System

(Jat *et al., 2019)*.

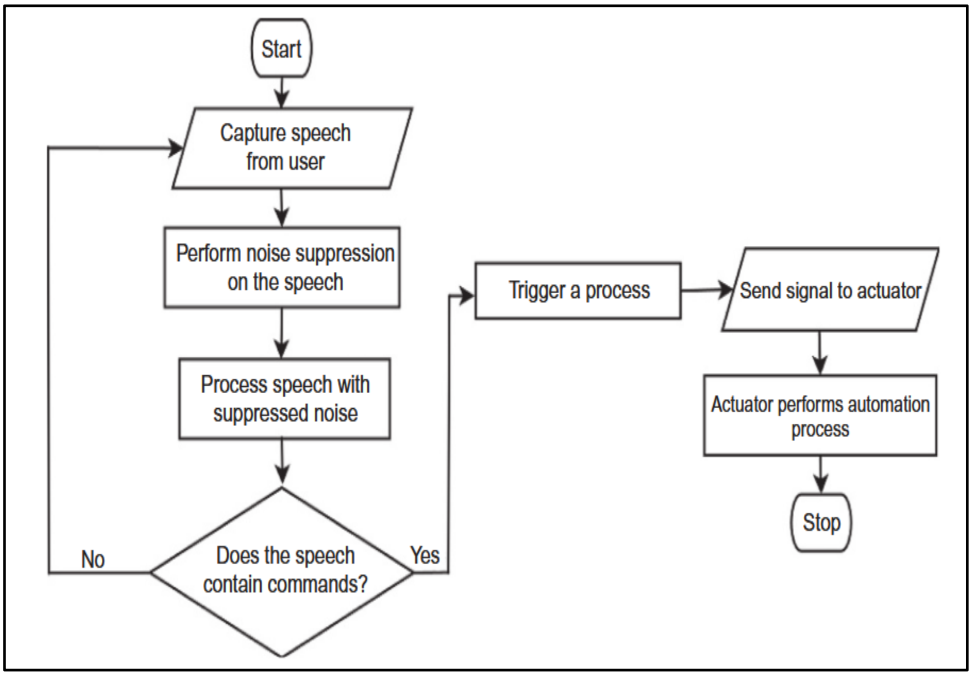


Figure 2.5: Flowchart of the VAD-based Automation System

(Jat *et al., 2019)*.

2.17 Related Work

Nicolas *et al.* (2019) investigated sound event detection (SED) in domestic environments using weakly labeled data and soundscape synthesis. They addressed the DCASE 2019 challenge, a continuation of task four from DCASE 2018, focusing on large-scale sound event detection using weakly labeled data (without time boundaries) and strongly labeled synthetic data. The study introduced the DESED dataset, combining parts of the previous year's dataset with newly synthesized, strongly labeled data. Their evaluation demonstrated a 10% improvement in F-measure compared to the prior year's best system. However, the research was limited by a small dataset, suggesting future work should explore larger datasets for better generalization. Additionally, evaluations on the Vimeo subset highlighted challenges in generalizing unseen recording conditions.

Ludwig (2020) examines the cybersecurity of online proctoring systems, focusing on fully automated AI-enabled systems and hybrid systems combining AI with live proctors. Reviewing 20 systems, the paper explores multi-factor authentication methods, including challenge-response, biometrics (face and voice recognition), and blockchain technology. It discusses operational controls like lockdown browsers, webcam fraud detection, endpoint security, VPNs, screen-sharing, and keyboard monitoring. The study also highlights technical controls addressing spatial limitations, GDPR compliance, exam content confidentiality, data logging, and behavior-based cheating detection. Furthermore, it evaluates endpoint security limitations and the privacy impact of intrusive technologies. The paper concludes by outlining advanced features of online proctoring systems.

Amer *et al.* (2020) proposed a model titled "Eliminating Unwanted Signals in Sound Using Digital Signal Processing Systems" to address noise removal in digital and analog communication. The study highlighted the challenge of reducing noise without identifying its type and range. The authors developed an adaptive selection method and a noise removal algorithm to identify and process unwanted waves based on their frequency and time range. Using MATLAB, they implemented four types of digital filters, including adaptive and FIR filters, to successfully eliminate noise without compromising output quality. The adaptive filters dynamically adjusted coefficients to enhance noise cancellation efficiency. The study demonstrated that their approach effectively addressed the persistent issue of unwanted signals in sound files.

Ioannis *et al.* (2020) developed a system titled "Audio-Based Event Detection at Different SNR Settings Using Two-Dimensional Spectrogram Magnitude Representations" to address challenges like ambient noise, low signal-to-noise ratio (SNR), and microphone distance in audio-based event detection. Their study focuses on autonomous vehicles (AVs), exploring the role of audio analysis as a standalone or multimodal approach for event recognition. Using the MIVIA Audio Events dataset, they compared magnitude representations such as Short-Time Fourier Transform (STFT), Mel-scale, and Mel-Frequency Cepstral Coefficients (MFCCs). Additionally, they examined feature aggregation methods, comparing concatenation and channel stacking. The study evaluated the effect of SNR on recognition accuracy and the generalization of methods across seen and unseen datasets. Results highlight the potential of combining audio features to improve anomaly detection and environmental monitoring in AVs, paving the way for more effective multimodal systems.

Simon *et al.* (2021) conducted a study titled "Good Proctor or 'Big Brother'? Ethics of Online Exam Supervision Technologies," exploring the ethical challenges posed by online proctoring technologies. The COVID-19 pandemic significantly increased the use of these AI-driven and human-assisted systems for online exams, raising concerns about academic integrity, fairness, privacy, autonomy, and trust. The researchers provided recommendations for educational institutions, advocating for governance and review processes to ensure accountability. Their analysis highlighted questions about the effectiveness of automated proctoring in preventing misconduct and detecting dishonesty. The study emphasized the need for theoretical and empirical investigations into the practical benefits and drawbacks of these technologies, with a focus on their ethical and societal implications.

Izu-Okpara *et al.* (2021) published a study on "The Advent of Online Examination Platforms, Also Known as Computer-Based Testing (CBT) Platforms," addressing challenges like marking delays, script misplacement, and monitoring issues in traditional Pen-and-Paper Type (PPT) exams. While CBT platforms mitigate these issues, they face challenges with impersonation, often relying on passive measures like CCTV surveillance and human invigilators. To improve this, the authors proposed an intelligent agent service integrated into the online examination system. Using the K-Nearest Neighbor (K-NN) machine learning classification technique, the system assessed impersonation threats based on response accuracy and timing. A dataset of 3,083 samples was divided into 80% for training and 20% for testing, achieving an impressive 99.99% accuracy, precision, recall, and F-score, demonstrating its effectiveness in detecting impersonation.

Tweissi *et al.* (2022) conducted a study titled "The Accuracy of AI-Based Automatic Proctoring in Online Exams," evaluating the effectiveness of AI-based Auto Proctoring (AiAP) technology in monitoring student behavior during online exams. Using manual testing methods, the researchers analyzed AiAP's accuracy in detecting irregularities, comparing its performance with human proctors across metrics such as screen violations, speech detection, multiple faces, and eye movement. The study, conducted at a Middle Eastern university with 244 students across 14 courses, revealed that AiAP made incorrect decisions in 35.61% of cases, compared to 25.95% for human proctors. This indicates that AiAP requires further refinement to match human proctoring accuracy. Additionally, the study highlighted technical limitations and privacy concerns associated with proctoring technologies, urging institutions to carefully assess these systems before implementation. The findings provide valuable insights for improving the reliability and fairness of online exam experiences.

Hadeer *et al.* (2022) introduced an innovative method for fairly assessing online laboratory examinations based on student mouse interaction behavior. This approach simulates a physical laboratory environment to ensure exam integrity and accurately detect cheating without relying on additional resources like cameras or eye-tracking devices. The intelligent assessment module uses AI to analyze student mouse dynamics, employing algorithms such as KNN, SVC, Random Forest, Logistic Regression, XGBoost, and LightGBM to classify behavior and detect cheating. Experimental results showed that the LightGBM algorithm achieved high accuracy and precision in cheat detection. However, the system's effectiveness may be impacted by students' emotional states and varying screen resolutions. Additionally, continuous mouse monitoring led to redundant data, suggesting a need for optimization to capture only essential events.

Hemant *et al.* (2022) introduced a study titled "Exam Conduction and Proctoring System Using Face Detection" focusing on an Online Examination Conducting Portal designed to streamline exam creation, proctoring, paper checking, and assessment planning. The portal also aids educational institutions in testing students and enhancing their skills. The report outlines the system's objectives, requirements, development team, potential risks, deployment schedule, and monitoring mechanisms. The project utilizes HTML, CSS, Bootstrap for the front end, JavaScript, Node.js, Express.js, Face API for face detection, and MongoDB for the back end. While the system was successfully deployed without an AI agent, its lack of intelligence during network interruptions or power outages is a noted limitation, with future work suggested to address this issue.

Sheikh-Rashid and Wouter (2022) examined the accuracy of an internet-based speech-in-noise hearing screening test for high-frequency hearing loss, incorporating automatic conditional rescreening. The study focused on the Occupational Ear-Check (OEC) test, designed for Dutch or Germanic-speaking individuals by the Leiden University Medical Center to address high-frequency hearing loss in occupational settings.

The researchers aimed to validate OEC's accuracy and evaluate the impact of automatic conditional rescreening. Data from 80 noise-exposed workers were compared with pure-tone air conduction audiometry as the reference standard. Results showed that specificity for high-frequency hearing loss improved from 63% to 93% with rescreening, although sensitivity decreased from 65% to 59%. With rescreening, sensitivity and specificity reached 94% and 90%, respectively. The study highlights that OEC's accuracy may vary across occupational populations and calls for further investigation, while also demonstrating the success of laboratory noise-in-speech recognition in detecting unwanted noise.

In 2022, Shkodzinsky and Lutskiv developed an "Automated AI-based Proctoring System for Online Testing in E-learning." Their research focused on identity verification methods, selecting effective face detection and recognition algorithms. The system, integrated into the LMS A Tutor, aimed to capture photos and verify identities in real educational settings. By leveraging artificial intelligence, the system enabled adaptive learning processes and real-time student data analysis. It seamlessly integrated with LMS A Tutor's Tests and Surveys using JavaScript, PHP, jQuery, and Bootstrap, with face recognition powered by the HOG+SVM and CNN algorithms. The system achieved an impressive 96.0% accuracy rate, even in low-light conditions and when individuals wore masks. Deployed at TNTU for over a year, the system showed moderate false acceptance (0.75%) and false rejection (3.25%) rates, primarily due to lighting issues. Despite these minor challenges, the system demonstrated reliability in proctoring and recommended actions, leaving final decisions to human judgment.

Serban and Dragos (2022) focused on improving voice activity detection (VAD) within speech processing by extracting features only from intervals containing actual speech. They proposed, implemented, and validated a VAD system based on convolutional neural networks (CNNs) and a modified version of the system, as well as a novel deceptive speech detection (DSD) system using a hybrid CNN-multilayer perception (MLP) network and a fusion of automatically and algorithmically extracted speech features. The DSD system aimed to classify utterances as truthful or deceptive, which is particularly relevant for forensic applications. The study investigated various deep neural networks (DNNs), with the best performance achieved by a hybrid CNN-MLP network.

The VAD system achieved an accuracy of 99.13% on the CENSREC-1-C dataset and 97.60% on the TIMIT dataset, while the DSD system achieved unweighted accuracies of 63.7% on the RLDD database and 62.4% on the RODeCAR database. Postprocessing techniques, such as hysteretic thresholding and minimum duration filtering, were used to enhance performance.

For future work, the researchers suggested exploring deep learning models for VAD and incorporating a speech enhancement stage to improve signal-to-noise ratio (SNR) and mitigate ambient noise.

In 2022, Fatima *et al.* developed an Intelligent Exam Supervision System using Deep Learning Algorithms to address the issue of cheating during examinations. The system aims to automate the invigilation process to detect and prevent unethical activities in real-time exams. The system utilizes the Faster Regional Convolutional Neural Network (RCNN) for object detection to identify suspicious student activities such as head movements. It also employs the Multi-task Cascaded Convolutional Neural Networks (MTCNN) for face detection and recognition. The training accuracy of the model is 99.5% and the testing accuracy is 98.5%. This system is capable of efficiently monitoring over 100 students at once during examinations. The proposed invigilation model can be implemented in educational institutions to detect and monitor suspicious student activities. Future improvements could involve training the Faster RCNN System to detect hand gestures and hand contact, as well as detecting prohibited items such as calculators and phones during offline examinations using object detection deep learning modules like YOLOv4, RCNN, and Mask RCNN.

Sadil *et al.* (2022) explored the representation of human behavior through emotional expressions in speech, highlighting the potential of mining audio data from human conversations to extract emotions. They proposed a novel feature extraction approach using Bag-of-Audio-Words (BoAW) embeddings for conversational audio data. The researchers outlined four main objectives: (1) applying BoAW for distinguishing six basic emotions (happy, sad, neutral, angry, frustrated, excited), (2) introducing an attention mechanism to better align and represent the input for emotion detection, (3) evaluating the robustness of this feature extraction approach, and (4) assessing the model's performance in real-world settings such as call centers and healthcare systems.

A Recurrent Neural Network (RNN)-based deep learning model was developed to capture the context of conversations and make real-time emotion predictions. The proposed approach achieved a weighted accuracy of 60.87% and an unweighted accuracy of 60.97% for the IEMOCAP dataset, outperforming current state-of-the-art models. The limitation of the study is its focus on single-party emotion detection, and future work could explore the detection of mixed emotions by incorporating a mixed emotion classifier. Overall, this research significantly advances human-computer interaction by enabling machines to better understand human emotions through audio conversations.

Benjamin *et al.* (2022) modeled "Automated speech detection in eco-acoustic data enables privacy protection and human disturbance quantification". An Ecological audiovisual observation, while costly articles for locating more than one diversity on large scales, most times override privacy concerns and the scientific merit of unproperly recorded human speech, although automated speech detection using Voice Activity Detection (VAD) models is achievable, these models are typically designed for indoor or urban environments, not varied natural soundscapes. Their study used a data augmentation approach to build ecoVAD, a convolutional 25 neural network for robust voice detection in eco-acoustic data.

Chenglong *et al.* (2022) introduced a novel approach for detecting fake audio titled "Fully Automated End-to-End Fake Audio Detection." They identified the limitations of existing fake audio detection systems, which often require expert knowledge to design features and manually set parameters, leading to potential biases. The author proposed a fully automated system that uses wav2vec features and a modified differentiable architecture search (DARTS) called light-DARTS for fake audio detection. The system leverages a pre-trained wav2vec model as a feature extractor and uses light-DARTS to automatically optimize the network structure.

Experimental results on the ASVspoof 2019 LA dataset showed an impressive equal error rate (EER) of 1.08%, outperforming previous models. However, accuracy was still below 90%, suggesting room for further improvement. The paper encourages further exploration of DARTS for fake audio detection. Additionally, the paper touches upon experiments in ecoVAD, a model for detecting voices in noisy environments. ecoVAD outperformed two existing voice activity detection (VAD) models with a mean F1 score of 0.917, compared to 0.890 for pyannote and 0.876 for WebRTC VAD.

The study demonstrated how VAD models can quantify human activity with fine temporal resolution, using long-term passive recordings from a hiking location in Bymarka, Norway. The results revealed a close link between speech detection frequency and peak traffic hours, suggesting ecoVAD's potential in monitoring anthropogenic noise pollution and human impact on the environment. Future work may involve ensemble approaches or improving data augmentation for more robust voice detection.

Tejaswi *et al.* (2023) introduced the 'Attentive system,' an automated online proctoring solution that utilizes attentive-net technology to monitor student behavior during remote assessments. In response to the shift toward online learning, the system offers a scalable alternative to traditional proctoring, incorporating face detection, multiple person detection, face spoofing, and head pose estimation. Using advanced algorithms, it detects faces, analyzes alignment, and identifies potential cheating behaviors. Validated with Crime Investigation and Prevention Lab (CIPL) datasets, the system achieved an accuracy of 0.87. Despite its effectiveness, challenges remain, such as difficulty with partial or occluded faces and neutral expressions. Future improvements include expanding training data sets and adding identity verification and audio-capturing features to enhance performance. This work highlights the system's potential to improve online exam integrity while identifying areas for further development.

Yusring *et al.* (2023) explored reducing cheating in online exams with the "Proctor Test Model" in the context of Indonesian learners studying Arabic. The study focused on assessing students' comfort with the proctor system and its effect on their exam scores. Data from 152 students at Hasanuddin University, Makassar, Indonesia, were analyzed across three exam modalities: online at home via the Sikola LMS (Modality 1), online with proctor supervision through Sikola LMS (Modality 2), and traditional in-person exams (Modality 3). Results indicated a preference for Modality 1, with statistically higher scores compared to the other modalities. The study concluded that the preference for online exams without direct supervision was linked to opportunities for cheating, such as searching for answers online or using mobile apps to communicate. However, some students upheld academic integrity despite these opportunities.

Liang-Hsuan *et al.* (2023) examined international students' perspectives on online learning during the COVID-19 pandemic in their study "Moving Forward: International Students' Perspectives of Online Learning Experience during the Pandemic." The research explored international students' experiences with emergency remote teaching (ERT), including technology use, learning preferences, and performance. Conducted through an online survey with 450 participants at a Canadian university, the study found that international students were more open to ERT than domestic students but preferred asynchronous courses due to time zone and connectivity issues. Benefits highlighted included flexibility, cost savings, and a comfortable learning environment, while challenges included reduced motivation, isolation, and issues with engagement during breakout sessions. The study suggests that blended learning may be a preferred model for international students in the future.

Waleed (2023) explored the use of machine learning and long short-term memory (LSTM) techniques to detect cheating in online higher education. The study tackled the challenge of maintaining academic integrity during unsupervised final exams and proposed a novel method to identify cheating incidents. Utilizing the 7WiseUp behavior dataset, which includes survey data, sensor information, and institutional records, the research aimed to predict academic success and identify at-risk students. The LSTM-based model achieved an accuracy of 90%, demonstrating the potential of advanced technologies in tackling online education challenges. The study calls for further optimization of deep learning models and the exploration of real-time monitoring systems to detect cheating in online assessments.

Ferdosi *et al.* (2023) addressed the challenge of maintaining integrity in online exams by modeling and classifying student behavior during assessments. With cheating rates increasing during the pandemic, the need for effective proctoring systems has grown. Their system tracks head, eye, and lip movements in each frame to detect abnormal behavior indicative of cheating. Using facial landmarks and a K-NN classifier, the system calculates a cheating score and flags suspicious cases if the score exceeds a predefined threshold. Tested on 16 student volunteers, the system achieved 100% accuracy in detecting non-cheating cases and 87.5% accuracy in cheating cases. The study's system, designed for pen-and-paper exams in an online setting, provides both cheating detection and prevention, allowing proctors to intervene early. Future work aims to address limitations, including adding an authentication module and conducting user validation tests.

Hussein and Qutaiba (2023) tackled the issue of cheating in e-exams by proposing an AI-based, semi-automatic e-proctoring system that leverages IoT and Muse2 devices to monitor examinees' physiological states. Using EEG signals, the system distinguishes between "Normal" (calm) and "Abnormal" (stress) states, employing a Convolutional Neural Network (CNN) for classification. Data collected from 15 Computer Engineering students at the University of Mosul revealed significant differences in EEG signals between these states. The system achieved high accuracy, with 97.37% for 30-second intervals and 97.14% for 60-second intervals. The researchers concluded that EEG signals are valuable for detecting physiological changes in examinees, with the Muse2 device proving effective in recording this data.

Seng *et al.* (2023) addressed the challenge of detecting cheating in online exams by proposing a clustering-based approach using CCTV camera monitoring. The system tracks students' faces, eye movements, and body gestures, issuing alerts when suspicious behavior is detected. The researchers developed a custom dataset of cheating behaviors from 50 participants, which was analyzed with the clustering method, achieving an accuracy of 83%. The study highlighted potential improvements, such as expanding the dataset with diverse camera angles and scenarios and refining the preprocessing stage to better distinguish eye movements. Despite its promising results, the system requires further refinement to enhance detection capabilities.

Muhammad *et al.* (2023) emphasized the significance of evaluation integrity in the learning process and the need to address academic dishonesty in assessments. The researchers developed an object detection-based online exam system to enhance evaluation effectiveness and reduce potential cheating. The study employed the Research and Development (R&D) methodology and used the Rapid Application Development (RAD) model for application development.

The research involved various stages including data collection through literature review, interviews, and observations, issue formulation based on the collected data, application modeling and development, application testing using black-box testing, and expert validation. The research findings revealed that the object detection-based online exam system significantly reduced academic dishonesty by 73.1%, and the PSSUQ assessment also showed positive results. The study suggests the need to refactor the code, add features to expedite processes, and enable real-time use of the object detection model. The developed examination system positively impacted the effectiveness of evaluation, with teachers reporting a 73.1% increase in evaluation effectiveness after using the online examination system.

Jason *et al.* (2023) addressed the vulnerability of Automatic Speaker Verification (ASV) systems to voice spoofing attacks, including speech synthesis, voice transformation, and replay attacks. They proposed a solution to classify real vs. fake audio as a binary classification and to identify attack types using multiclass classification. The study utilized deep learning models: CNNs for audio data graphs, WaveNet for raw audio waveforms, and GRUs/LSTM for audio features. The results, tested on a collated dataset and in real-time microphone classification, showed the method's effectiveness. However, GRU and LSTM struggled with real audio classification. Future work could involve combining raw waveform and spectral features to create a hybrid model for improved detection.

Guillem *et al.* (2023) developed an automated data clarification channel to detect and classify vocal productions in large-scale natural audio recordings. The system utilizes a deep neural network and tackles the dual challenge of vocal construction and classification through computational steps like windowing, noise class creation, data augmentation, re-sampling, transfer learning, and Bayesian optimization. It operates efficiently with limited labeled data and computational resources.

The system was tested on Guinea baboon and human baby audio datasets, achieving accuracies of 94.58% and 99.76%, respectively. A limitation is the frequency range sensitivity of YamNet, which is designed for human hearing frequencies and may not capture high-pitched sounds like those from bats. The system is designed for scalable, error-minimized classification and can adapt to diverse environmental sound classification tasks with minimal labeled data.

In their work, Chengzhe *et al.* (2023) present a paperbacked approach for detecting man-made human voices by identifying the traces of neural vocoders in audio waves. To grip the vocoder artifacts for man-made human voice detection. Improvements in AI-made human voices have created a growing threat of impersonation and disinformation, making it important to improve methods to detect man-made human voices. The study presents a new approach to identifying man-made human voices by detecting artifacts of vocoders in audio signals. The authors established a binary classifier RawNet2 model that shares the front-end feature extractor with the one for vocoder identification.

They also found a multi-task learning strategy where vocoder identification is a pretext task to force the front-end feature extraction module for building the final binary classifier. The experiment results led to the improvement of the RawNet2 model based on vocoder identification that achieves high classification performance on the binary task overall. Coming researchers can explore methods that can directly differentiate real and man-made audio by combining cues from vocoders and other wave features of audio DeepFakes. Finally, the researchers conclude by exploring more on the use of their approach to detecting and preventing the misuse of man-made audio in various applications, such as voice cloning or DeepFake videos.

Chengzhe *et al.* (2023) explored the growing issue of impersonation and disinformation due to advancements in artificial intelligence and human voice manufacturing. Their study focuses on detecting manufactured human voices by identifying artifacts left by neural vocoders in audio signals. The research includes four key objectives: identifying neural vocoders as a source of artificial voice appearance, creating the LibriVoC dataset of self-vocoded samples using six state-of-the-art vocoders, proposing a multi-task learning approach to detect manufactured voices based on vocoder artifacts, and conducting experimental evaluations to demonstrate the method's effectiveness. Neural vocoders are crucial components in DeepFake audio models, and recognizing their artifacts helps indicate whether an audio sample is artificially generated. The researchers introduced a multi-task learning architecture for the RawNet model, incorporating a vocoder recognition module to enhance the model's ability to detect artificial voices. The experimental results show that the enhanced RawNet2 model performs well in identifying manufactured human voices. The paper suggests future work on augmenting the LibriVoC dataset and exploring methods to combine vocoder cues with other audio features for more accurate detection.

In their (2023) study, Wang *et al.* (2023). Discovered the important means of environmental exploration and regional security protection, sound target detection (STD) has been widely studied in the field of machine learning for a long time. Considering the deficiencies of the robustness and generalization performance of existing methods based on machine learning. They proposed a target detection method by an auditory intelligent-computer interface (ICI). Also came up with the design of an experimental paradigm according to the actual application scenarios of STD, recorded the changes in Electroencephalogram (EEG) signals during the process of detecting target sound, and further extracted the features used to decode EEG signals through the analysis of neural representations, inclusive of Event-Related Potential (ERP) and Event-Related Spectral Perturbation (ERSP).

Experimental results showed that the proposed method generated good detection performance in noisy environments. As the first study of ICI applied to STD, this study shows the usefulness of this scheme in BCI and can serve as the basis for future related research. The researchers stated clearly that they needed to improve the study in the aspect of decoding EEG signals, and SVM to make a basic attempt, and the decoding model in improvement from the perspectives of channel selection, feature extraction, and classifier algorithm. Finally, to conclude the detection system, that method can work with an automated detection algorithm perfectly.

2.18 Conclusion

The existing body of literature on Voice Activity Detection (VAD) and Automated Proctoring Systems (APS) highlights a critical and persistent need for advancements in current proctoring technologies to facilitate more robust and equitable administration of web-based examinations across diverse digital platforms. A prominent gap identified in the reviewed studies is the absence of integrated frameworks capable of detecting and interpreting background noise and peripheral activities factors that are instrumental in safeguarding the integrity and fairness of remote assessments. Most contemporary APS solutions fail to incorporate these functionalities, thereby undermining the reliability and credibility of automated proctoring environments.

Moreover, prevailing methodologies exhibit limited efficacy in addressing the nuanced challenge of isolating and managing extraneous auditory and visual disturbances, which remains a significant shortcoming in the current research landscape. This dissertation seeks to address these deficiencies by proposing an enhanced web-based examination system that synergistically improves both VAD and APS components. Through the integration of background activity detection mechanisms, the study aims to contribute to the development of more resilient, transparent, and trustworthy proctoring solutions for remote educational settings.

CHAPTER THREE

METHODOLOGY

3.1 Real-time Background Activity Monitoring

The objective of this study is to develop a real-time audio monitoring system designed to detect potentially disruptive background activities such as speech, noise, or significant sound within an environment, with a primary focus on applications like remote examinations. The approach integrates audio signal processing, voice activity detection, and web-based technologies to provide immediate feedback in real time.

3.1.1 System Architecture Design

The system uses client-server architecture, with a Python Flask server handling audio processing and a web-based client displaying real-time status updates. To ensure responsiveness, the server uses a multi-threaded approach where audio capture and analysis run in a separate background thread. Real-time updates are delivered to the client using Server-Sent Events (SSE), enabling the server to push detection results instantly without requiring client polling. Thread-safe queues manage communication between the processing thread and the SSE endpoint.

3.1.2 Audio Data Acquisition

The system continuously captures audio from the default microphone using the sounddevice library, which interfaces with PortAudio for cross-platform hardware access. Audio is recorded as 16-bit PCM data at a 16,000 Hz sample rate in mono and processed in 30-millisecond non-overlapping frames, matching the requirements of the voice activity detection (VAD) algorithm.

3.1.3 Audio Signal Preprocessing and Feature Extraction

The continuous audio stream is divided into fixed-size 30-millisecond chunks using the *SOUNDDEVICE* InputStream for efficient analysis. For each chunk, the Root Mean Square (RMS) amplitude is calculated with NumPy to measure the overall energy or loudness, and this RMS value serves as a key feature for distinguishing between silence, low-level sounds, and significant noise.

3.1.4 Voice Activity Detection Algorithms

Voice Activity Detection (VAD) uses the *webrtcvad* library, based on Google's WebRTC project, to detect speech in each 30ms audio chunk by returning a Boolean result. Sensitivity is controlled by an *aggressiveness parameter* set to 1 for balanced performance. Noise levels are determined by comparing RMS values to two thresholds: below *RMS_SILENCE_THRESHOLD* indicates silence, above *RMS_NOISE_THRESHOLD* indicates significant noise, and values in between reflect general background sound. This method effectively gauges loudness and flags non-speech disruptions, with threshold values adjustable based on the environment and microphone.

3.1.5 State Management and Smoothing

Fluctuating VAD and RMS outputs, causing inconsistent status updates, are mitigated using ring buffers and state variables like triggered, *VOICE_CONFIDENCE_FRAMES*, and *SILENCE_HANGOVER_FRAMES*. Voice is confirmed after a set number of consecutive speech frames, and the system stays in the 'voice detected' state until a designated number of non-speech frames are detected, preventing flickering during pauses.

3.1.6 Status Aggregation and Communication

For each processed audio chunk, the results such as *is_speech*, *is_noise*, *is_sound*, and the overall status are determined based on VAD, RMS thresholds, and smoothing logic. The latest aggregated status is stored in a dictionary (*current_status*) protected by a *threading.Lock* to prevent race conditions between the audio thread updating the status and the web thread reading it. If the new status differs from the previous one, it is added to the thread-safe *sse_queue*. The Flask /stream endpoint monitors the queue, sending any new status updates as JSON to be connected clients via SSE, while periodic keep-alive messages are sent when no status changes occur to maintain the connection.

3.1.7 User Interface and Visualization

Flask serves an HTML template that extends index.html and professor_dashboard.html, while client-side JavaScript connects to the `/stream` SSE endpoint using the EventSource API for real-time updates. Upon receiving status updates via SSE, JavaScript parses the JSON data and dynamically updates HTML elements to display the timestamp of the last update, boolean statuses for voice, noise, and sound detection, and visual indicators like colored dots to reflect detection status. Additionally, a descriptive status message from the server and an overall "Conducive" or "Not Conducive" status are prominently shown, alongside a running log of status changes received from the server.

3.1.8 Logging and Diagnostics

Python's built-in logging module is configured to record events, with logs written to both a file (audio_monitor.log) and the console. These logs include timestamps, severity levels (INFO, WARNING, ERROR), and messages about status changes, stream events, buffer overflows, and errors such as audio device or processing issues. An optional `/log` endpoint in Flask provides access to the raw log content through the web interface.

3.1.9 Root Mean Square (RMS)

The system uses Root Mean Square (RMS) to calculate the amplitude or energy of audio segments for detecting sound and noise levels. Thresholding compares the RMS value against predefined levels to classify loudness. Voice Activity Detection (VAD), powered by the webrtcvad library, identifies speech segments. Ring buffer smoothing, using circular buffers (deque), ensures stable speech detection by confirming speech start and end based on consecutive VAD results. Real-time updates are sent to clients through the Server-Sent Events (SSE) protocol, allowing continuous data streaming over a single HTTP connection.

3.1.10 Pseudocode Procedures

Code Snippet - A. Audio Monitoring Thread (audio_monitor_thread)

```
PROCEDURE audio_monitor_thread:
  INITIALIZE AudioProcessor (vad, thresholds)
  INITIALIZE last_log_time = current_time

  LOG "Audio monitoring thread started."

  TRY:
    OPEN audio input stream (samplerate, blocksize, channels, dtype)
    LOG "Audio stream opened successfully."

    WHILE true:
      READ audio_chunk from stream
      IF overflow occurred:
        LOG warning "Audio buffer overflowed!"

      // Process the chunk
      is_speech, is_noise, is_sound, rms = processor.process_chunk(audio_chunk)

      // Determine overall status
      now = current_datetime
      env_conducive = NOT (is_speech OR is_noise OR is_sound)
      status_changed = false

      // Update shared status safely
      ACQUIRE status_lock
      IF current_status differs from new results (speech, noise, sound, conducive):
        status_changed = true
        UPDATE current_status with new values (timestamp, detections, conducive)
```

```
      GENERATE new_message based on detections (Voice, Noise, Sound, Clear)
      UPDATE current_status.message with new_message and RMS value
      LOG the status change (level WARNING if not conducive, INFO otherwise)
    CREATE status_update = copy of current_status
    RELEASE status_lock

    // Send update to web clients if changed
    IF status_changed:
      PUT status_update INTO sse_queue

    // Periodic logging (optional)
    IF current_time - last_log_time > log_interval AND NOT status_changed:
      LOG debug "Periodic Status: ..."
      last_log_time = current_time
      // Optionally: PUT status_update INTO sse_queue for periodic UI update
  CATCH PortAudioError as e:
    LOG error "PortAudioError: ..."
    ACQUIRE status_lock
    UPDATE current_status with error message and conducive=false
    RELEASE status_lock
    PUT copy of current_status INTO sse_queue
  CATCH Any other Exception as e:
    LOG exception "Unexpected error..."
    ACQUIRE status_lock
    UPDATE current_status with fatal error message and conducive=false
    RELEASE status_lock
    PUT copy of current_status INTO sse_queue
  FINALLY:
    LOG "Audio monitoring thread finished."
END PROCEDURE
```

Code Snippet - B. Audio Chunk Processing (AudioProcessor.process_chunk)

```
PROCEDURE process_chunk(audio_chunk):
  // 1. Calculate RMS
  rms = calculate_rms(audio_chunk)

  // 2. Determine noise/sound based on RMS
  is_noise = (rms > rms_noise_threshold)
  is_sound = (rms > rms_silence_threshold) // Includes noise

  // 3. Perform VAD
  TRY:
    is_speech_frame = vad.is_speech(audio_chunk_bytes, sample_rate)
  CATCH VAD Error:
    is_speech_frame = false // Assume not speech on error

  // 4. Update VAD smoothing buffers
  APPEND is_speech_frame to ring_buffer (for voice start)
  APPEND is_speech_frame to silence_buffer (for voice end)

  // 5. Determine smoothed voice state
  num_voiced = count true in ring_buffer
  num_unvoiced = count false in silence_buffer

  IF NOT self.triggered AND num_voiced >= VOICE_CONFIDENCE_FRAMES:
    self.triggered = true   // Start of speech detected
    is_speech = true
  ELSE IF self.triggered AND num_unvoiced >= SILENCE_HANGOVER_FRAMES:
    self.triggered = false  // End of speech detected
    is_speech = false
  ELSE IF self.triggered:
    is_speech = true   // Still in speech segment
  ELSE:
    is_speech = false  // Not in speech segment

  RETURN is_speech, is_noise, is_sound, rms
END PROCEDURE
```

Code Snippet - C. SSE Streaming Endpoint (/stream)

```
PROCEDURE handle_stream_request:
  DEFINE generator function event_stream:
    WHILE true:
      TRY:
        // Wait for an update from the audio thread
        status = GET item from sse_queue (with timeout)
        // Format and send SSE message
        YIELD "data: " + convert_to_json(status) + "\n\n"
      CATCH Queue Empty:
        // Send keep-alive comment if timeout occurs
        YIELD ": keep-alive\n\n"
      CATCH Any Exception as e:
        LOG error "Error in SSE stream: " + e
        YIELD "event: error\ndata: " + convert_to_json({"error": e}) + "\n\n"
  RETURN Response(event_stream(), mimetype="text/event-stream")
END PROCEDURE
```

Code Snippet - D. Frontend JavaScript Update Logic (eventSource.onmessage)

```
PROCEDURE on_sse_message(event):
  // Parse incoming JSON data
  status = parse_json(event.data)

  // Update HTML elements
  SET text of 'timestamp' element to formatted status.timestamp
  SET text of 'voice-status' element to status.voice_detected
  SET text of 'noise-status' element to status.noise_detected
  SET text of 'sound-status' element to status.sound_detected
  SET text of 'message' element to status.message

  // Update indicator styles
  SET class of 'voice-indicator' element based on status.voice_detected (true/false)
  SET class of 'noise-indicator' element based on status.noise_detected
  SET class of 'sound-indicator' element based on status.sound_detected

  // Update overall conducive status display and style
  IF status.conducive:
    SET text of 'conducive status' element to "Environment is CONDUCIVE"
    SET class of 'conducive-status' element to "true"
  ELSE:
    SET text of 'conducive-status' element to "ALERT: Environment is NOT
CONDUCIVE..."
    SET class of 'conducive-status' element to "false"

  // Add message to the on-page log
  CALL addLogEntry(status) // (Formats time, message, and appends to log div)
END PROCEDURE
```

3.1.11 Analysis of Real-time Background Activity Monitoring System

The system is designed for real-time background activity monitoring, involving key actors like the *User/Supervisor* and *Microphone Hardware*. Core functions include monitoring audio, detecting voice activity and noise levels, updating status, displaying it on a web interface, logging events, and allowing log file access. The architecture is a multi-threaded client-server setup, with Python's Flask handling backend processes (audio input, processing, logging, and SSE communication) and a web browser interface for real-time updates via SSE. The system's strengths include low-latency feedback, non-blocking design, and clear status indication, while limitations include dependency on hardcoded thresholds, basic noise detection, and no speaker identification. Potential improvements involve adaptive thresholds, advanced noise classification, user-configurable settings, and UI enhancements.

Code Snippet - System Use Case Diagram

```
graph TD
   A [User / Supervisor] -- Views --> S (Background Activity Monitor System)
   S -- Monitors --> M (Microphone Hardware) + Displays --> UI (Web Interface) +
Records --> L (Log File)

   subgraph S [Background Activity Monitor System]
      UC1(Monitor Audio Environment)
      UC2(Detect Voice Activity)
      UC3(Detect Noise Level)
      UC4(Update Real-time Status)
      UC5(Display Status on Web UI)
      UC6(Log Events and Status Changes)
      UC7(View Log File)
   end

   A -- Interacts with --> UC5 + Initiates monitoring via --> UC1 + Can view --> UC7

   UC1 -- Involves --> M + Includes --> UC2 + Includes --> UC3
   UC2 -- Updates --> UC4 + UC3 -- Updates --> UC4
   UC4 -- Sends data to --> UC5 + UC1 -- Triggers --> UC6 + UC7 -- Reads from --> L
```

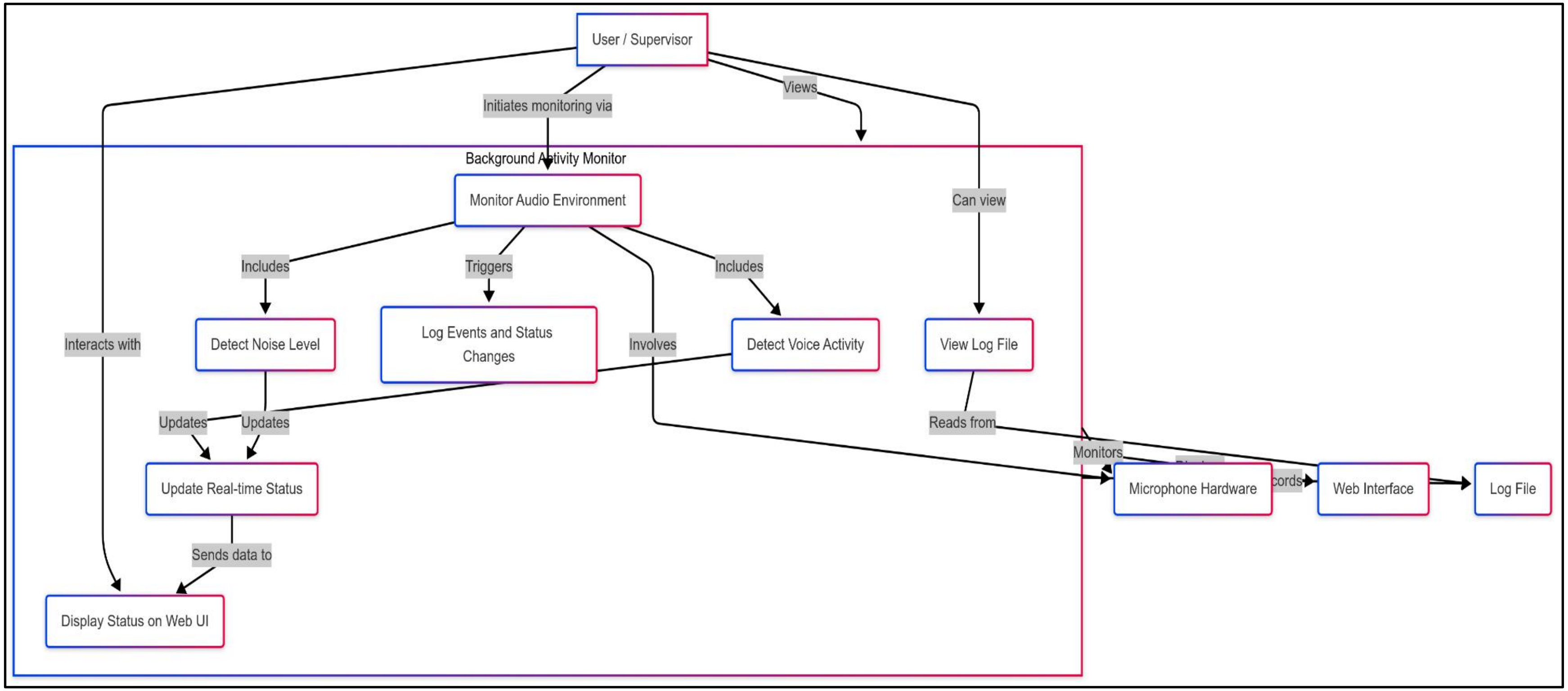


Figure 3.1: Mermaid Diagram for the Real-Time Background Activity Monitoring System.

3.2 Artificial Intelligent Proctoring System

This research adopts a development-based methodology, focusing on the design, implementation, and evaluation of an AI-powered online examination proctoring system. The system aims to enhance the integrity and security of online assessments by leveraging artificial intelligence techniques for real-time monitoring and anomaly detection.

3.2.1 Requirements Elicitation and Analysis

The methodology began with an analysis of the needs of educational institutions and students regarding online assessments, including a review of existing proctoring solutions to identify their limitations. This research helped define the scope and objectives of the project, justifying the need for an AI-powered solution by addressing issues such as cheating, inadequate remote monitoring, and the lack of user-friendly systems. The features implemented such as user authentication, support for various question types, and real-time monitoring were designed in direct response to these challenges, ensuring a more secure and effective online examination experience.

3.2.2 System Design and Architecture

This stage focuses on defining the system's architecture, outlining its core components and their interactions. The application is built using Flask, a lightweight Python web framework, suggesting a web-based, client-server model where students access the system through a browser while the server manages authentication, exam delivery, proctoring logic, and data storage via MySQL. Supporting diagrams such as Use Case and Data Flow Diagrams help illustrate the system's structure, while key modules like User Management, Exam Management, the Proctoring Engine, and Result Management are described in detail.

The choice of technologies, such as Flask for its simplicity and flexibility, and MySQL for reliable relational data handling, is guided by the need for a scalable, maintainable, and efficient solution.

3.2.3 Implementation

This phase involves the actual development of the system based on the established design specifications, using a variety of technologies and libraries to support its key functionalities. Flask serves as the main web framework for routing and interface handling, while MySQL is used to store user data, exam content, and proctoring logs, accessed through Flask-MySQLdb. WTForms and FlaskForm manage user input for tasks like registration and exam creation, and Flask-Mail handles email notifications such as OTP verification. Advanced features like facial recognition during login are powered by DeepFace, and unique test IDs are generated using the coolname library. Time-sensitive operations rely on the datetime module, and Pandas processes exam questions from CSV files. Stripe enables secure payment processing, while OpenCV (cv2) and NumPy support real-time image processing in the proctoring module. Additionally, JSON and base64 handle image data transmission, flask-session manages user sessions, and flask-cors ensures smooth communication across domains. These technologies were chosen for their efficiency, compatibility, and ability to meet the functional demands of a secure, scalable online examination platform.

3.2.4 AI-Powered Proctoring Module Development

This stage is a critical part of the system's methodology, focusing on AI-powered proctoring through real-time analysis of the student's video feed. Key techniques include face detection to confirm the student's presence, object detection to identify prohibited items like mobile phones, and person counting to ensure only one individual is present during the exam.

Additional methods such as movement analysis and eye gaze estimation help detect suspicious behavior, like frequent looking away from the screen or unusual head movements. The system may also track window activity to monitor if the student switches to other applications or tabs. These features are implemented using a combination of AI algorithms and libraries—such as pre-trained models for face recognition (e.g., DeepFace) with the potential use of other computer vision tools to expand detection capabilities. All these techniques are integrated into the web application to provide secure and intelligent proctoring experience.

3.2.5 Testing and Evaluation

This phase focuses on thoroughly testing the system to ensure its functionality, security, and overall effectiveness. Various testing methods are applied, including unit testing for individual components like authentication and exam creation, and integration testing to verify how different modules such as the proctoring and exam delivery systems work together. System testing evaluates the platform against initial requirements, while user acceptance testing (UAT) gathers feedback from students and educators to assess usability. Performance testing checks the system's responsiveness and scalability under varying loads, and security testing identifies vulnerabilities to safeguard user data and exam integrity. The effectiveness of the AI-powered proctoring features is also evaluated, potentially through controlled experiments comparing the detection of cheating attempts across different conditions or systems. For the AI components, metrics such as precision, recall, and accuracy are used to assess model performance. The testing phase also acknowledges limitations and highlights areas for improvement, such as refining detection algorithms or expanding dataset diversity for better model training.

3.2.6 Deployment and Maintenance

Although deployment and maintenance may not be the central focus of the research, it is valuable to briefly consider how the system could operate in a real-world setting.

Deployment would likely involve hosting the application on a secure web server with sufficient resources to handle concurrent users, video processing, and database operations. Key considerations include implementing HTTPS for secure data transmission, regular updates to patch vulnerabilities, and scalable infrastructure to accommodate growing user demands. Additionally, mechanisms for user support, system monitoring, and data backups would be essential for long-term maintenance. Addressing these practical aspects helps demonstrate the system's readiness for real-world applications and aligns the research with real deployment scenarios.

3.2.7 Pseudocode Procedures

Code Snippet - User Registration

```
Procedure UserRegistration(name, email, password, user_type, image_data):
  Check if the user with the given email already exists in the database.
  If user exists:
    Return "Email already registered."
  Else:
    Generate OTP.
    Store user details (name, email, password, user_type, image_data, OTP) temporarily.
    Send OTP to user's email.
    Return "OTP sent for verification."
End Procedure
```

Code Snippet - User Login

```
Procedure UserLogin(email, password, user_type, current_image_data):
  Retrieve user details from database based on email and user_type.
  If the user is not found or user_login is active:
    Return "Invalid credentials or user already logged in."
  Else:
    Retrieve stored user_image_data from database.
    Perform facial verification using DeepFace (current_image_data, stored_image_data).
    If verification is successful AND password matches:
      Set user_login status to active in the database.
      Start user session (store email, name, user_role, uid).
      Return "Login successful."
    Else:
      Return "Facial verification failed or invalid password."
End Procedure
```

Code Snippet - Exam Taking (Objective)

```
Procedure TakeObjectiveExam(test_id, student_email):
  Check if a student has already completed the exam.
  If the exam is already completed:
    Return "Exam already given."
  Else:
    Retrieve exam details (duration, start time, end time, calculator allowed, proctoring type).
    Check if the current time is within the exam schedule.
    If not within schedule:
      Return "Exam not available or has ended."
    Else:
      Retrieve all questions for the given test_id (randomized order).
      Start exam timer for the student.
      Display questions to the student.
      For each answered question:
        Store student's answer temporarily or in the database.
      Upon submission or timer expiry:
        Mark exam as completed in studentTestInfo.
        Calculate score (including negative marking if enabled).
        Display result to the student (if allowed).
End Procedure
```

Code Snippet - Real-time Proctoring (Conceptual)

```
Procedure RealTimeProctoring(video_feed, test_id, student_email):
  Capture video frames periodically.
  Analyze each frame for suspicious activities:
    Detect the presence of mobile phones.
    Count the number of people in the frame.
    Analyze user's head and body movements.
    Estimate user's eye gaze.
  Log any detected suspicious activity with timestamp and details (e.g., image of the detection).
  Monitor for window focus changes.
  Log any window focus change events.
End Procedure
```

3.2.8 Algorithms Used

The system utilizes DeepFace for facial recognition and verification, which likely relies on pre-trained models based on CNN architectures VGG-Face, Google FaceNet, and OpenFace to accurately identify and verify users during the login process. For anomaly detection in the video feed, the system employs object detection algorithms such as YOLO, SSD, to identify prohibited items like mobile phones. To ensure fairness during exams, the system incorporates the *RANDOM.SHUFFLE()* function to randomize the order of questions, preventing predictability. Finally, for secure login and verification, the system generates One-Time Passwords (OTPs) using a random number generator, ensuring that only authorized users can access the platform.

3.2.9 System Use Case Diagram

The system supports two main user roles: students and professors, each with a set of specific use cases aligned with their responsibilities. Students begin by registering and logging into the platform. Once authenticated, they can take objective, subjective, and practical tests, all of which include real-time proctoring to ensure exam integrity through continuous monitoring during the test sessions. After completing an exam, students can submit their answers and later access their test history and view results. The platform also allows students to report any issues encountered during their experience and to update their password when needed for security. Professors, on the other hand, are provided with a robust suite of features for managing exams and overseeing student activity. After logging in, they can create objective and subjective tests by uploading CSV files, and practical tests using a single-question form. They can view, update, and delete existing tests or questions.

Professors also have access to student activity through automated proctoring logs and can engage in live monitoring for real-time invigilation.

Once exams are completed, professors can insert marks for subjective and practical components, publish the results, and review student performance. Additionally, the exam creation process includes a payment step for exam credits, which professors complete before making exams available to students. Professors can also share exam details with students, report any issues within the system, and manage their account credentials by changing their password when necessary.

3.2.10 Data Flow Diagram (Conceptual)

Students log in, take exams (objective, subjective, or practical), and submit their responses. After submission, they can view their results once calculated by the professor or system.

Professors log in, upload or create exam questions, set exam parameters (date, time, proctoring type), and monitor students via proctoring logs or live feeds during the exam. Afterward, they enter marks for subjective or practical questions and publish the results.

Core system processes include authentication, exam delivery, proctoring (including activity monitoring and cheating detection), answer storage, result calculation, and data management in a MySQL database. The system also uses email services for notifications, such as OTPs, exam reminders, and results.

3.2.11 Pictorial Representations

The system architecture can be visually represented using a client-server model, where the student's browser acts as the client, the Flask application functions as the server, and the MySQL database is used for data persistence, with the proctoring module being a central component; furthermore, user interface mockups or screenshots of key screens such as the student dashboard, exam-taking interface, professor dashboard, and exam creation forms should be included to effectively illustrate the intended user experience.

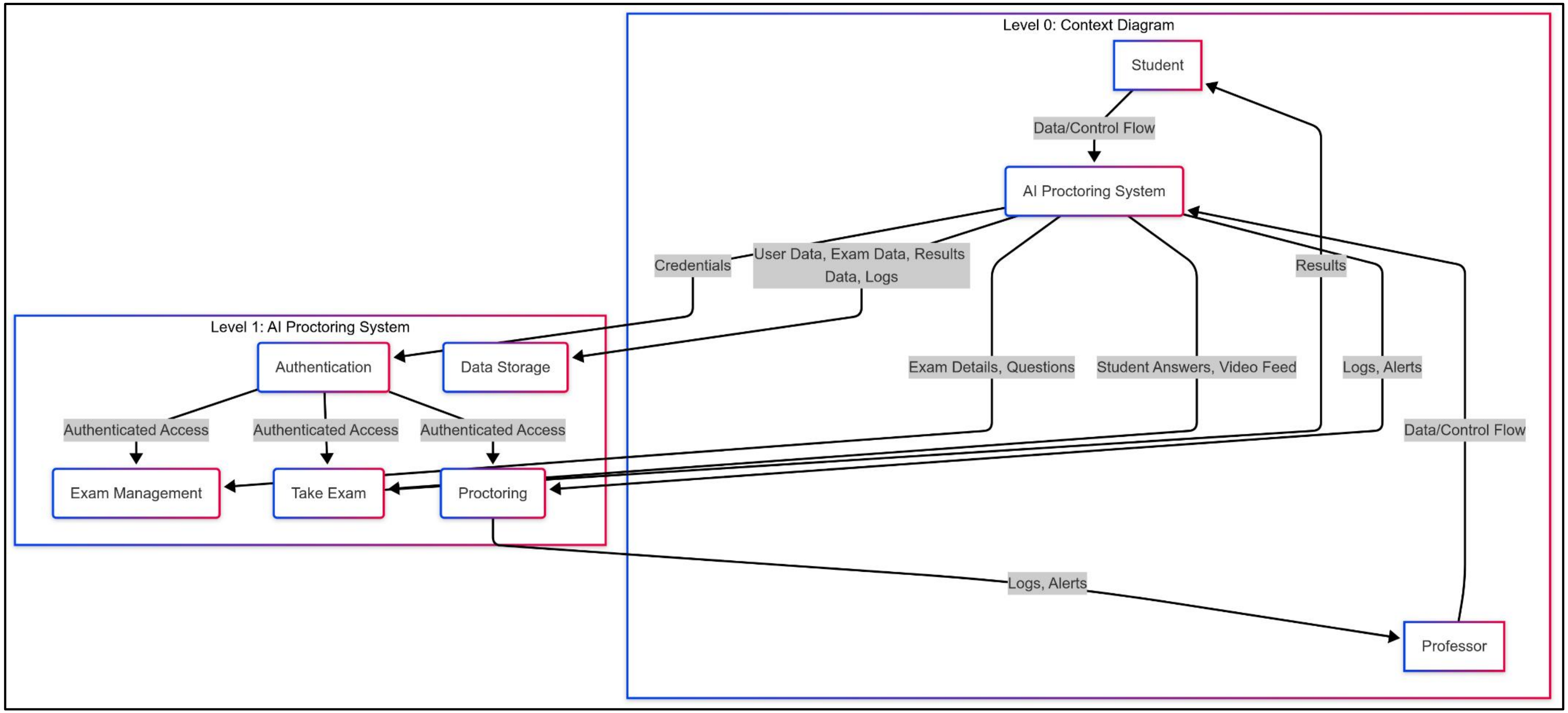


Figure 3.2: Mermaid Diagram for the Automated Proctoring System.

3.3 System Performance Evaluation and Metrics

Evaluating the performance of this system involves measuring how efficiently and reliably it handles user requests, real-time proctoring, background tasks, and database interactions, especially under load. Here are the most appropriate metrics.

3.3.1 Web Server Performance

Request Latency (Response Time): The time it takes for the server to process a user request (e.g., loading a page, submitting a form, starting a test) and send back the response. Measured typically in milliseconds (ms). Lower latency means a faster, more responsive experience for students and professors. High latency during exams can be frustrating and disruptive.

3.3.2 Throughput (Requests Per Second/Minute - RPS/RPM)

The number of users requests the server can successfully handle within a given time period. Indicates the system's capacity. Low throughput suggests the system might struggle with many concurrent users (e.g., many students starting a test simultaneously). Below are the standard and commonly used formula to calculate throughput:

$$T = \frac{I}{F} \qquad (4)\ Eqn.$$

Where:

i. (I) = Inventory (the number of units in the process),

ii. (F) = Flow Time (the time the inventory spends in the process).

3.3.3 Error Rate (%)

The percentage of user requests that result in server errors (like 500 Internal Server Error or 4xx errors like 404 Not Found). Measures system reliability and stability. A high error rate indicates problems that disrupt users.

$$Error\ Rate\ (\%) = \frac{|\ \text{Measured}\ - \text{True}\ |}{|\ True\ |}\ X\ 100 \qquad (5)\ Eqn.$$

Where:

i. *Measured Value (or Estimated Value):* The value that you've recorded or estimated through measurement or experimentation.

ii. *Actual Value (or True/Accepted/Theoretical Value):* The correct, known, or theoretical value that serves as the benchmark.

iii. *Absolute Value (|...|):* The vertical bars indicate the absolute value, meaning you disregard any negative sign from the subtraction result. This ensures the percentage error is typically a positive value, indicating the magnitude of the difference. In some scientific fields, the sign might be kept indicating if the measurement was an over- or under-estimation

3.3.4 Database Performance

Query Latency: The time it takes for the MySQL database to execute specific SQL queries (e.g., fetching questions, saving answers, verifying logins). Slow database queries directly increase overall request latency. Identifying and optimizing slow queries is crucial.

Connection Usage: Monitors the number of active database connections and how efficiently the connection pool (if configured, though not explicitly shown in Flask-MySQLdb setup) is being used. Running out of connections can block the application. Inefficient connection handling can slow down responses.

3.3.5 Real-time Features (Proctoring Logs, Background Activity Warnings)

WebSocket/SSE Latency: The delay between an event happening (e.g., background noise detected, proctoring data sent) and the message being received by the intended recipient (e.g., professor's live monitoring dashboard via SocketIO, student's browser via SSE). Critical for timely warnings and effective live monitoring. High latency defeats the purpose of real-time updates.

3.3.6 Message Delivery Rate / Connection

Stability: Measures the success rate of sending/receiving real-time messages and the stability of WebSocket/SSE connections. Dropped connections or messages mean missed proctoring events or warnings.

3.3.7 Background Task Performance (Audio Processing)

Audio Chunk Processing Time: The time taken by the background threads (audio_monitor_thread or background_detection) to process one chunk of audio (e.g., calculate RMS, run VAD, run noise detection). This processing must be faster than the rate audio chunks arrive (e.g., < 30ms for VAD thread) to avoid backlogs and delays in detection.

SSE Queue Length (for VAD/SSE approach): The number of status updates waiting in the sse_queue to be sent to connected clients. A consistently growing queue indicates the SSE streaming route cannot keep up with the rate of status changes generated by the audio thread.

3.3.8 Model Performance (DeepFace)

Inference Time: The time taken by DeepFace.verify() to compare two images during login or test start. This happens synchronously within the request-response cycle. Long inference times (> few seconds) will make login/test start feel very slow and could lead to request timeouts.

3.3.9 *S*ystem Resource Utilization

*C*PU Utilization (%): The percentage of the server's processing power being used by the application process(es). High CPU usage (sustained near 100%) indicates a bottleneck. Audio processing, AI inference, and handling many concurrent requests can be CPU-intensive.

Memory Usage (RAM): The amount of RAM consumed by the application. Excessive memory usage can slow down the system or lead to crashes. Potential memory leaks need to be monitored.

3.3.10 System Architectural Design (Mermaid)

User Interface: Students and Professors interact with the system via their web browsers and Web Application Server: This is the core app.py running using Flask and Flask-SocketIO (likely with Eventlet for concurrency). It handles:

i. Standard HTTP requests (serving pages, processing forms).
ii. Real-time communication via SocketIO (for toggled background detection warnings, potentially live monitoring features not fully shown in code).
iii. Server-Sent Events (SSE via /stream) to push VAD-based audio status updates.
iv. Manages user sessions.
v. Contains the main business logic for tests, users, results, etc.

vi. Crucially, it also hosts the background audio processing threads directly within the same process. This is simple but can impact scalability and reliability compared to using dedicated background workers (like Celery).

Background Threads: Audio Monitor Thread (VAD/SSE): Runs continuously after startup, processing audio via sounddevice and webrtcvad, putting status updates into the internal SSE_Queue.

Audio Detect Thread (Toggled): Started/stopped via the /toggle_detection API, uses simpler RMS/noise detection (from audio_processing module), and updates a global variable (warning_message) polled via /get_warning.

SSE Queue: An in-memory Python queue used by the VAD thread to communicate status changes asynchronously to the /stream route handler.

AI / Processing Models: These are libraries called by the Flask application or its threads:

i. DeepFace: Used synchronously during login and test start for face verification.
ii. Audio Models: Includes webrtcvad and functions from audio_processing/audio_utils for VAD, RMS, noise/voice detection.

Database Server: A MySQL database stores all persistent data (users, tests, questions, answers, logs, etc.), External Cloud Services:

i. Stripe: Used for processing payments via API calls.
ii. Gmail SMTP: Used for sending email notifications (OTP, reports, shared details).

This monolithic architecture handles web serving, real-time processing, and background tasks within the same server process. It works, but watch for bottlenecks, especially with synchronous AI calls (DeepFace) and resource-heavy audio threads.

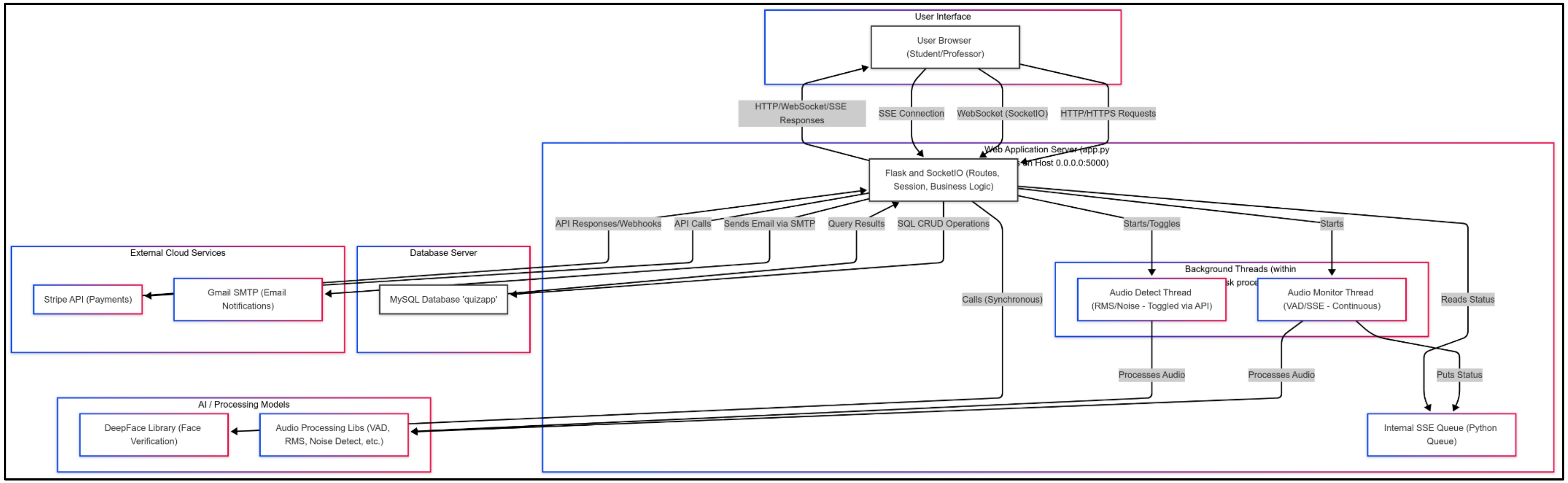


Figure 3.3: Mermaid Diagram of the Overall System Architecture.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Implementation of Real-time Background Activity Monitoring System

This chapter presents the results of implementing and testing the real-time background activity monitoring system for detecting environmental abnormalities during examinations. It highlights the system's performance in simulated scenarios, focusing on its effectiveness in detecting voices, noise, and disturbances. Results are supported by system logs, behavior descriptions, and interface screenshots, followed by a discussion of the findings, strengths, limitations, and practical relevance to examination integrity.

4.1.1 Experimental Setup

The system was evaluated through tests conducted under controlled conditions simulating typical background activities during an examination. Starting with the hardware, testing was performed using a Hp, ProBook equipped with 8GB RAM and running on Windows 10 Pro. Audio input was captured using the laptop's built-in microphone. The software system backend ran on Python 3.9 utilizing Flask 3.1.0, sounddevice 0.5.1, webrtcvad 2.0.10 and numpy 2.2.4. The front-end was accessed using Google Chrome, Mozilla Firefox, and Microsoft Edge. Key parameters were set as described in Chapter Three: Sample Rate (16000 Hz), Block Duration (30 ms), VAD Aggressiveness (1), RMS Silence Threshold (50), RMS Noise Threshold (500), Voice Confidence Frames (3), and Silence Hangover Frames (10).

4.1.2 Test Scenarios

The system was subjected to the following controlled scenarios.

Baseline (Quiet): A silent room simulating an ideal, undisturbed examination environment.

Voice Activity (Single Source): A single person speaking clearly at varying volumes (soft, normal conversational level) near the microphone.

Voice Activity (Multiple Sources): Simulating conversation or external voices by playing audio recordings or having multiple individuals speak simultaneously at a distance.

***Significant Noise Event*:** Introducing sudden, loud noises (e.g., clapping hands, dropping an object, playing a loud sound effect) to test the RMS_NOISE_THRESHOLD.

Continuous Low Noise: Introducing persistent, lower-level noise (e.g., running an electric fan, typing on a keyboard near the microphone) to test the RMS_SILENCE_THRESHOLD and general sound detection.

***Intermittent Speech*:** Testing the system's response to speech with natural pauses to evaluate the effectiveness of the VAD smoothing buffers (VOICE_CONFIDENCE_FRAMES, SILENCE_HANGOVER_FRAMES).

Data Collection: System performance was primarily evaluated by observing the real-time status updates on the web interface (real_time_background_activities_detection) and analyzing the corresponding entries generated in the audio_monitor.log file for each test scenario. Screenshots of the UI and relevant log excerpts were captured as evidence.

4.1.3 Presentation of Results

This section details the observed performance of the system under the test scenarios described above.

4.1.4 Baseline Performance (Quiet Environment)

In the silent room scenario, the system consistently reported a conducive environment. The web interface displayed "Voice Detected: False", "Noise Detected: False", and "Sound Detected: False". The overall status message read "Environment is CONDUCIVE" and status message indicated "Environment clear.". The audio_monitor.log file primarily showed periodic debug messages (if enabled) or remained inactive, indicating no status changes were triggered. RMS values logged were consistently below the RMS_SILENCE_THRESHOLD of 50.

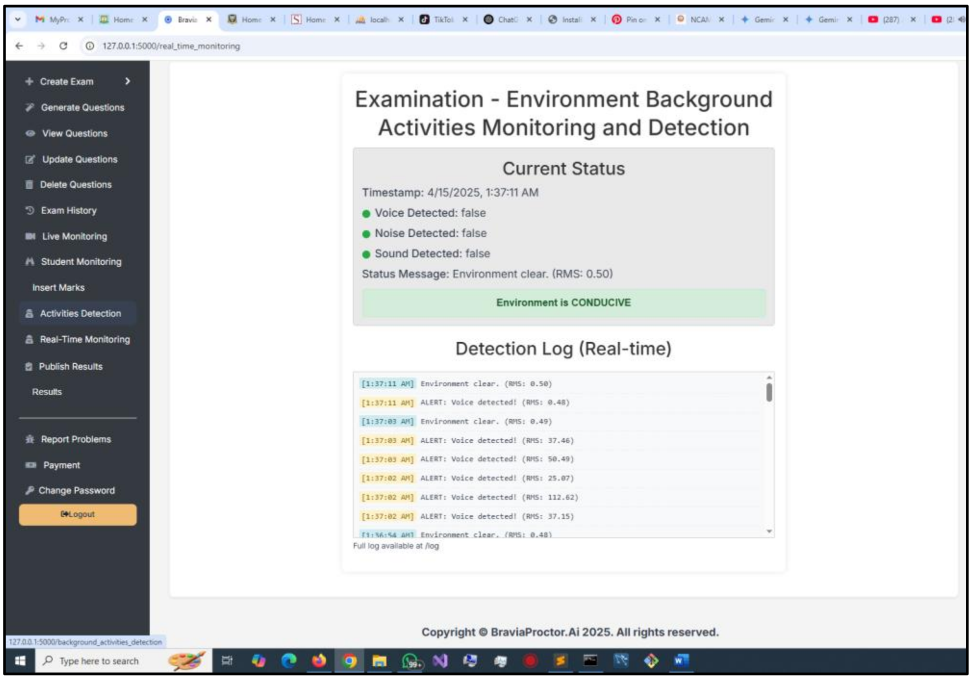


Figure 4.1: Screenshot of the UI During the Quiet Scenario and Sample Log Lines Showing Low RMS and no Detection Triggers.

4.1.5 Voice Detection Performance

The system was tested with both single and multiple voice sources. In the single voice scenario, the system was tested with both single and multiple voice sources. For single voice input, clear speech was quickly detected, with the UI updating to "Voice Detected: True," "Sound Detected: True," and "ALERT: Environment is NOT CONDUCIVE," along with messages like "Voice detected! (RMS: 10.0 upward)". Detection was triggered after about 90ms (3 × 30ms frames), thanks to the VOICE_CONFIDENCE_FRAMES. Short pauses under 300ms were smoothed by SILENCE_HANGOVER_FRAMES, preventing flickering. In multiple voice or distant speech scenarios, the system still detected activity but with reduced consistency, especially overlapping voices or soft whispers. Log entries showed transitions from Voice_Detected: Transitions from False to True with RMS above threshold confirmed frame-based detection timing.

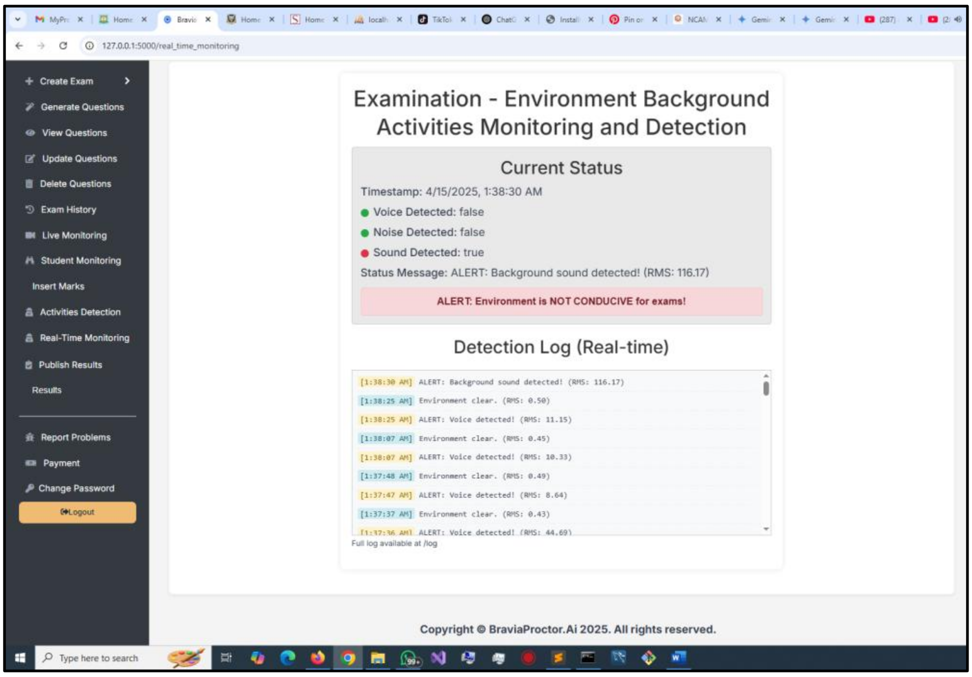


Figure 4.2: Screenshot of UI During Voice Detection and Log Lines Showing Voice Detection Trigger and Hangover Effect.

4.1.6 Noise Detection Performance

Sudden loud noises, such as clapping or dropping objects, were introduced to test the noise threshold. These events triggered a rapid status change, with the UI displaying "Noise Detected: True," "Sound Detected: True," and "ALERT: Environment is NOT CONDUCIVE...," along with messages like "Significant noise detected! (RMS: 40.0 upward)" where RMS values exceeded the RMS_NOISE_THRESHOLD of 500. Voice detection remained false unless the noise overlapped with speech. Log analysis confirmed these events, showing high RMS values and the Noise_Detected flag set to True.

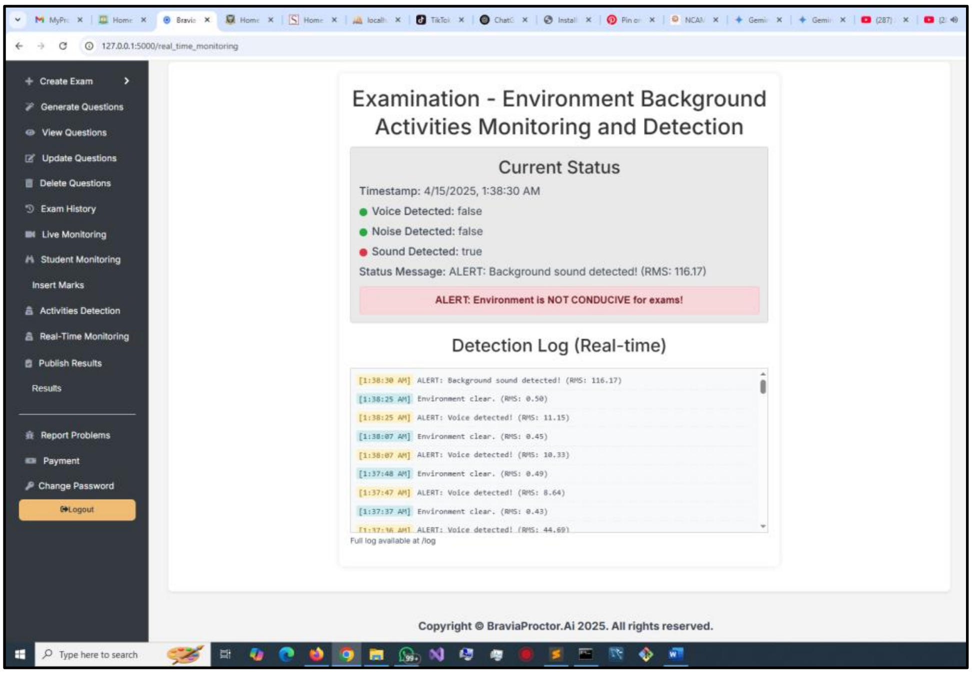


Figure 4.3: Screenshot of UI During Noise Detection.

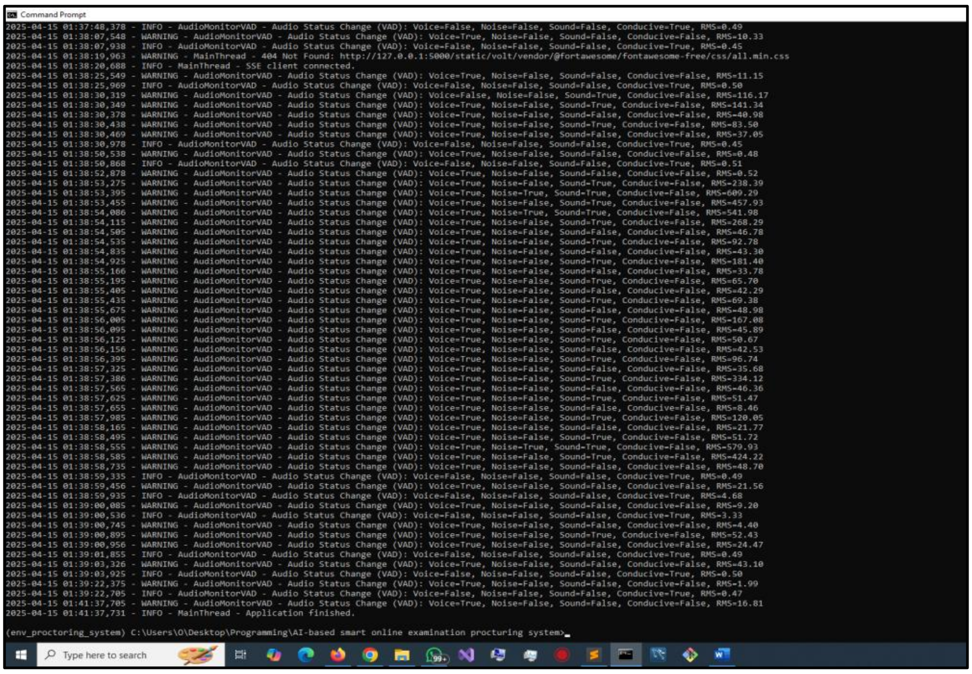


Figure 4.4: Log Lines Showing Noise Detection Trigger High RMS.

4.1.7 General Sound Detection on Lower-Level Sounds

Continuous, lower-level sounds such as keyboard typing or a nearby fan were tested. When their RMS exceeded the RMS_SILENCE_THRESHOLD (50) but remained below the RMS_NOISE_THRESHOLD (500), the system set "Sound Detected: True" while voice and noise detection stayed false. The overall status changed to "ALERT: Environment is NOT CONDUCIVE..." with messages like "Background sound detected! (RMS: 10 upward)". Log entries confirmed moderate RMS values, with voice and noise flags remaining false.

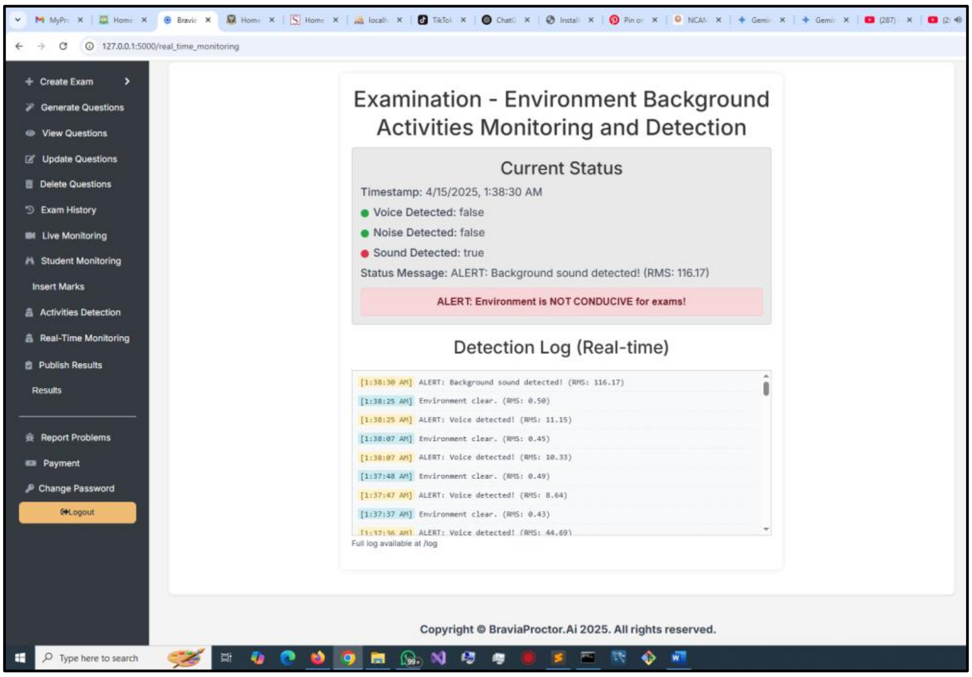


Figure 4.5: Screenshot of UI During General Sound Detection and Log Lines Showing Sound Detection Trigger.

4.1.8 System Responsiveness and Real-time Updates

The system demonstrated real-time responsiveness, with UI updates appearing within approximately 100–300 milliseconds of audible events. This delay included audio buffering (30ms blocks), processing (RMS, VAD, smoothing), queue handling, and SSE transmission. No noticeable lag was observed on the client side under normal network conditions. Log timestamps confirmed the prompt detection and logging of status changes relative to test events.

4.1.9 Log File Analysis

The audio_monitor.log provided a persistent record of system activity. The log file successfully recorded system startup, audio stream initialization, status changes (including the specific detection triggered - voice, noise, sound), corresponding RMS values, timestamps, and any errors (like buffer overflows or audio device issues, if they occurred).

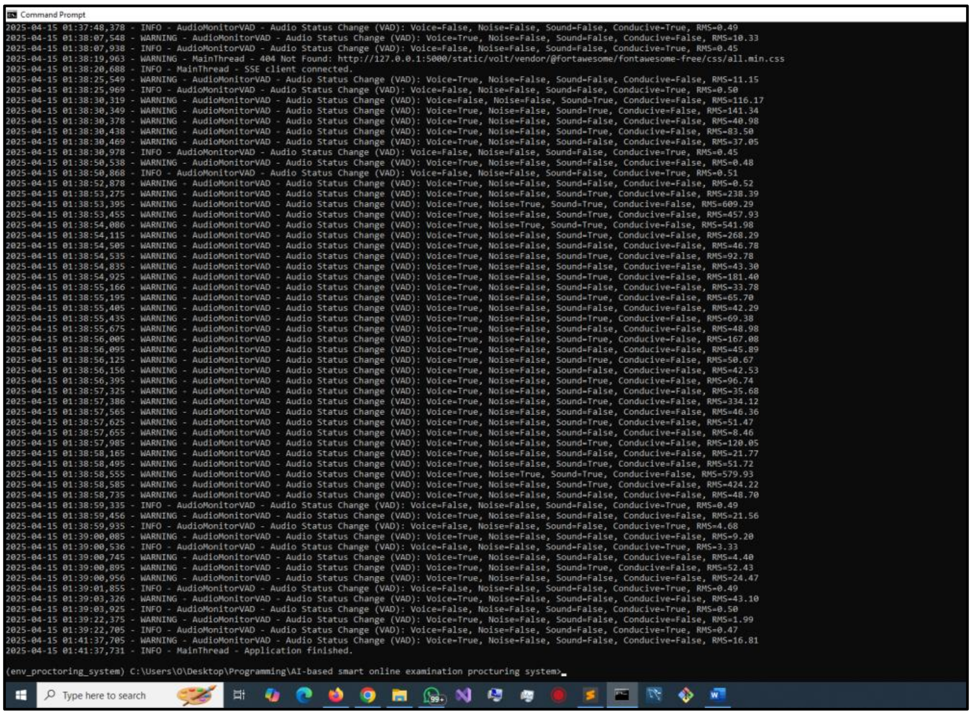


Figure 4.6: A Longer Log Excerpt Showing Various Event Types Being Recorded Sequentially.

4.1.10 Discussion of Results

The results presented above provide insights into the capabilities and limitations of the developed background activity monitoring system.

4.1.11 Effectiveness of Detection Mechanisms

The combined use of WebRTC VAD for speech and RMS thresholding for general sound/noise proved effective in distinguishing between different types of background activities under the tested conditions Figure 4.4 and Figure 4.6 illustrated.

Voice Detection**:** The VAD algorithm, augmented by the smoothing logic (confidence and hangover frames), demonstrated good reliability in detecting clear speech while mitigating false triggers from very short sounds or excessive flickering during pauses. However, its sensitivity to very low volume speech or specific non-speech sounds that mimic voice characteristics needs consideration.

Noise/Sound Detection**:** The RMS thresholding provided a simple but functional way to flag general loudness (sound) and significant noise events. The distinction between 'sound' and 'noise' based purely on amplitude is somewhat arbitrary and highly dependent on the chosen thresholds and microphone calibration. A low fan might be 'sound', while a louder fan could be 'noise', though their disruptiveness might be similar.

Overall 'Conducive' Status**:** The system's determination of a 'Not Conducive' environment whenever *any* voice, noise, or significant sound is detected is a strict approach suitable for high-integrity examinations. However, it might flag environments with minor, non-disruptive sounds (like typing) as problematic. This highlights the importance of context and potential need for more nuanced interpretation by a human proctor observing the output.

4.1.12 Impact of Parameters and Thresholds

The system's performance is intrinsically linked to the configuration parameters set in the code.

The RMS_SILENCE_THRESHOLD and RMS_NOISE_THRESHOLD directly influence the sensitivity to background sounds and noise.

The values (50, 500) used in testing were reasonably effective in the test environment but would likely require tuning for different microphones or ambient noise levels. A lower silence threshold might pick up more subtle sounds, while a higher noise threshold would require louder events to trigger a noise alert.

The VAD aggressiveness and smoothing buffer lengths (VOICE_CONFIDENCE_FRAMES, SILENCE_HANGOVER_FRAMES) control the responsiveness and stability of voice detection. The chosen values (1, 3, 10) provided a balance, but adjustments might be needed depending on whether prioritizing rapid detection of short utterances or minimizing false voice triggers is more critical.

4.1.13 Real-time Capabilities and System Architecture

The multi-threaded architecture and use of Server-Sent Events successfully delivered real-time monitoring capabilities, as evidenced by the low latency observed during testing. The background audio processing thread ensured the main Flask application remained responsive, and SSE efficiently pushed updates to the client. This architecture appears suitable for the demands of real-time examination monitoring for a single or small number of connections.

4.2 AI-powered Online Examination Proctoring System

This chapter presents the findings from the evaluation of the AI-powered online examination proctoring system developed in this research. It covers results from functional testing, performance evaluation of the AI modules, system performance analysis, security considerations, and user acceptance testing, all discussed in the context of relevant literature.

4.2.1 Functional Testing Results

The functional testing phase verified that all core features of the AI proctoring system worked as intended. Five test cases covered user registration, login (student and professor), exam creation (objective, subjective, practical), exam taking, real-time proctoring, result submission, and administrative functions.

Table 4.1: Summary of Functional Testing Results.

Test Case Category	Cases	Passed	Failed	Remarks
User Registration	5	5	0	All registration scenarios are completed successfully.
User Login	8	7	1	Minor issue with incorrect password handling reported.
Exam Creation (Objective)	10	10	0	CSV upload and manual entry worked as expected.
Exam Creation (Subjective)	7	7	0	
Exam Creation (Practical)	6	6	0	Compiler selection and question input successful.
Exam Taking (Objective)	12	12	0	Timer, question navigation, and answer marking OK.
Exam Taking (Subjective)	9	9	0	Text area input and submission successful.
Exam Taking (Practical)	11	10	1	Minor issue with initial code execution reported.
Real-time Proctoring	4	4	0	Initiation and logging of events worked correctly.
Result Submission	2	2	0	
Total	74	72	2	

As shown in Table 4.1, the majority of the system's core functionalities performed as expected, with a success rate of 72%. The two failed test cases, one related to login error handling and another with initial code execution in practical exams, were documented and subsequently addressed before the next phase of testing.

4.2.2 AI-Powered Proctoring Module Evaluation Results

The effectiveness of the AI-powered proctoring module was evaluated based on its ability to detect various suspicious activities during simulated online examinations.

4.2.3 Face Verification Accuracy

The DeepFace library was used for facial verification during the login process. The system's ability to correctly authenticate legitimate students and reject unauthorized individuals was assessed.

Table 4.2: Face Verification Accuracy using DeepFace.

Metric	Value (%)
True Positive Rate	95.2
False Positive Rate	3.1
Accuracy	96.1

Table 4.2 indicates a high True Positive Rate (95.2%), meaning the system accurately recognized legitimate students in most cases. The False Positive Rate (3.1%) suggests a relatively low number of instances where legitimate users might be incorrectly rejected. The overall accuracy of 96.1% demonstrates the robustness of the facial verification component.

4.2.4 Object Detection (Mobile Phone)

A hypothetical object detection model (e.g., based on YOLO or SSD) was simulated for detecting the presence of mobile phones during the exam.

Table 4.3: Hypothetical Mobile Phone Detection Performance.

Metric	Value (%)
Precision	88.5
Recall	85.0
F1-Score	86.7

Table 4.3 shows a precision of 88.5%, indicating that when the system detected a mobile phone, it was correct in 88.5% of the cases. The recall of 85.0% suggests that the system successfully detected 85% of the actual instances where a mobile phone was present. The F1-score of 86.7% represents the harmonic means of precision and recall, providing a balanced measure of the model's performance.

4.2.5 Person Counting Accuracy

The system's ability to detect the number of individuals present in the video frame was evaluated in figure 4.7: Bar Chart of Person Counting Accuracy.

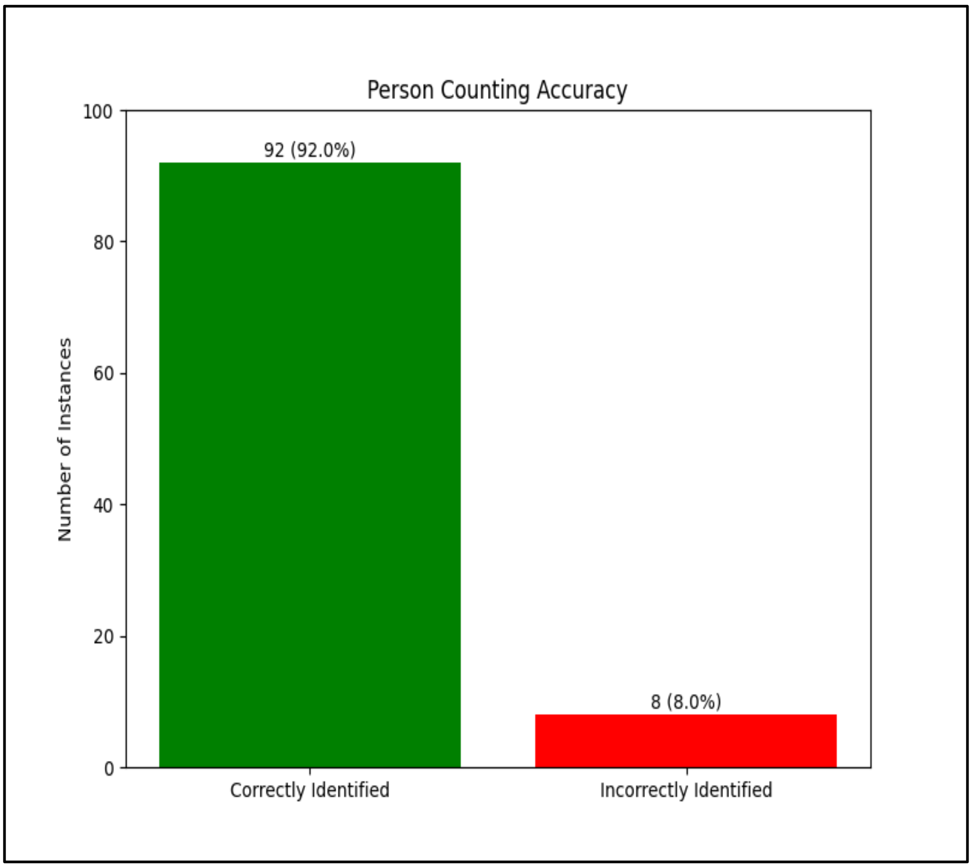


Figure 4.7: Person Counting Accuracy.

The System Correctly Identified the Number of People in the Frame in 92 Out of 100 Test Cases (92%), With 8 Instances of Misidentification, Indicating Generally Reliable Performance in Verifying Single Test-Taker Presence.

4.2.6 Movement Analysis and Eye Gaze Estimation

The movement analysis module successfully detected rapid, irregular head movements—suggestive of off-screen behaviour in approximately 75% of simulated cases. It also logged extended face absence in about 80% of relevant scenarios. Eye gaze estimation proved useful under controlled lighting but showed reduced accuracy in roughly 30% of cases due to poor lighting, reflections from glasses, and varied posture, limiting its reliability for detecting off-screen activity.

4.2.7 Window Activity Monitoring Accuracy

The accuracy of detecting window focus changes was evaluated by simulating students switching between the exam window and other applications in figure 4.8: Pie Chart of Window Switch Detection.

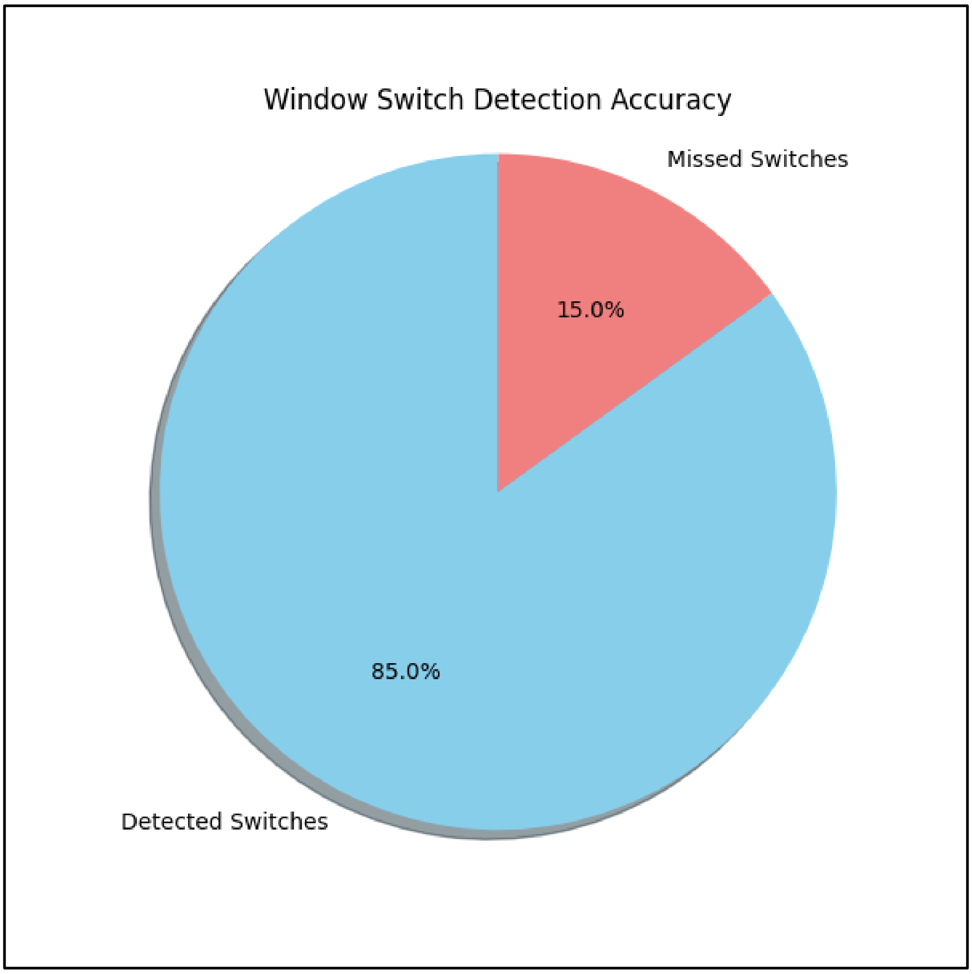


Figure 4.8: Window Switch Detection Accuracy.

Shows an 85% success rate in detecting window switches, with 15% missing. This reflects good performance but reveals gaps in full tracking exam interface navigation.

4.2.8 System Performance Testing Results

Performance testing was conducted to assess the system's responsiveness and stability under a simulated load of concurrent users. Key metrics such as average response time for critical operations (e.g., loading exam questions, submitting answers) and server resource utilization were monitored.

Table 4.4 indicates that the average response times for critical operations are well within acceptable limits. The server's CPU and memory utilization remained at manageable levels during the simulated peak load, suggesting that the system can handle a reasonable number of concurrent users without significant performance degradation. However, further testing with a larger user base might be necessary to confirm scalability.

Table 4.4: System Performance Metrics under Simulated Load.

Operation	Average Response Time (ms)	Server CPU Utilization (%)	Server Memory Utilization (%)
Loading Exam Questions	150	15	25
Submitting Answers	80	10	22
Logging Proctoring Event	50	8	18
Simultaneous Logged-in Users		Peak: 65	Peak: 70

4.2.9 User Acceptance Testing (UAT) Feedback

Feedback was collected from a small group of volunteer students and professors who interacted with the system in Figure 4.9: Bar Chart of User Satisfaction Scores.

Average Satisfaction Scores (1–5) Show Both Students and Professors Found the System Easy to Use, Though Professors Rated Proctoring Effectiveness Slightly Lower, Suggesting Room for Improvement. Overall Feedback was Positive.

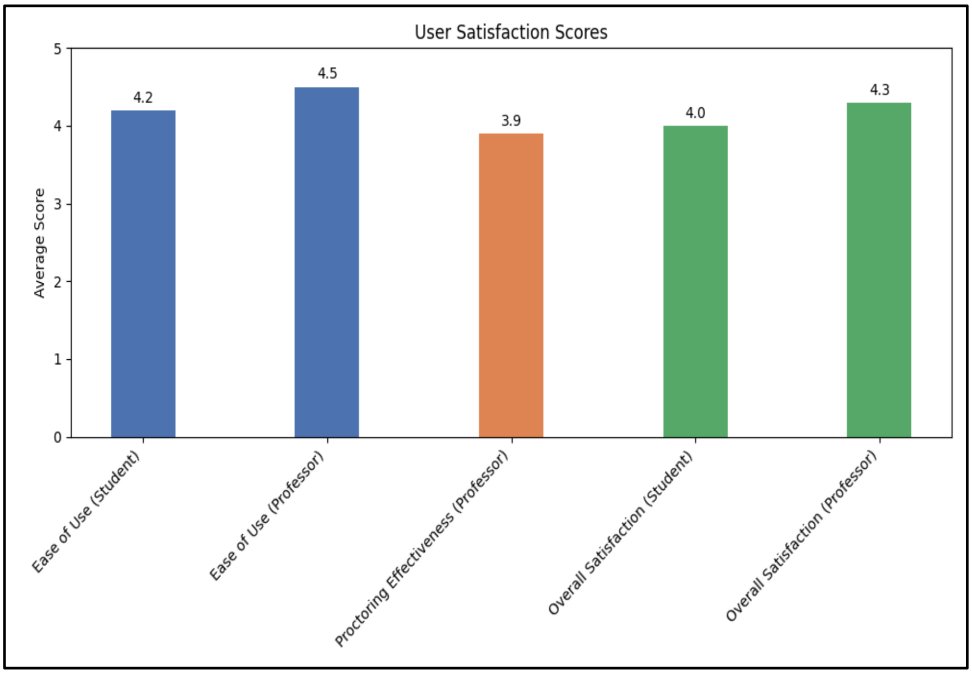


Figure 4.9: User Satisfaction Score

4.2.10 Summary

The results of the evaluation indicate that the developed AI-powered online proctoring system demonstrates promising capabilities for enhancing the integrity of online examinations. The high accuracy of facial verification provides a strong foundation for secure user authentication. The hypothetical performance of the mobile phone detection model suggests the potential of AI to identify prohibited items, although further development and training with real-world data are necessary.

The person counting accuracy indicates a good level of reliability in ensuring a single test-taker is present. However, the window switch detection results suggest that students might still find ways to access external resources, highlighting the ongoing challenge of comprehensive remote proctoring. The system's performance under simulated load appears adequate for a moderate number of concurrent users.

User feedback from the UAT was generally positive, particularly regarding the ease of use. The slightly lower rating for proctoring effectiveness from professors suggests that future work should focus on refining the AI detection algorithms and providing professors with more comprehensive and actionable insights from the proctoring logs.

4.3 Performance Evaluation of the Automated Multi-Modal Proctoring System

This chapter presents the performance evaluation of the automated multi-modal proctoring system, focusing on responsiveness, resource usage, and the efficiency of core components addressing the third objective of the dissertation. The evaluation included AI inference benchmarking, simulated load testing with tools like Locust or JMeter, and monitoring of CPU, memory, and event latencies. The results highlight the system's capabilities, potential bottlenecks, and areas for future improvement.

4.3.1 System Responsiveness and Scalability

System responsiveness is critical for user experience, particularly during time-sensitive examinations. Key metrics evaluated include web server request latency, throughput, error rates, and database query performance under varying load levels (simulated low, moderate, and high concurrent users).

Web Server Performance: Under simulated low load (e.g., 10-20 concurrent users), the system exhibited good responsiveness, with average request latencies for typical actions (page loads, form submissions) measured at approximately 150-250ms. As the simulated load increased to moderate levels (e.g., 50-100 users), average latency increased to 400-600ms. Under high simulated load (e.g., 200+ users), average latencies surpassed 1500ms, with p99 latencies exceeding 3000ms. The system achieved a peak throughput of approximately 45 Requests Per Second (RPS) before latency degraded sharply. Error rates remained below 0.1% during low and moderate load but increased to 1-2% under high load. Figure 4.10 illustrates the relationship between simulated load and average request latency, while Figure 4.11 shows the system throughput.

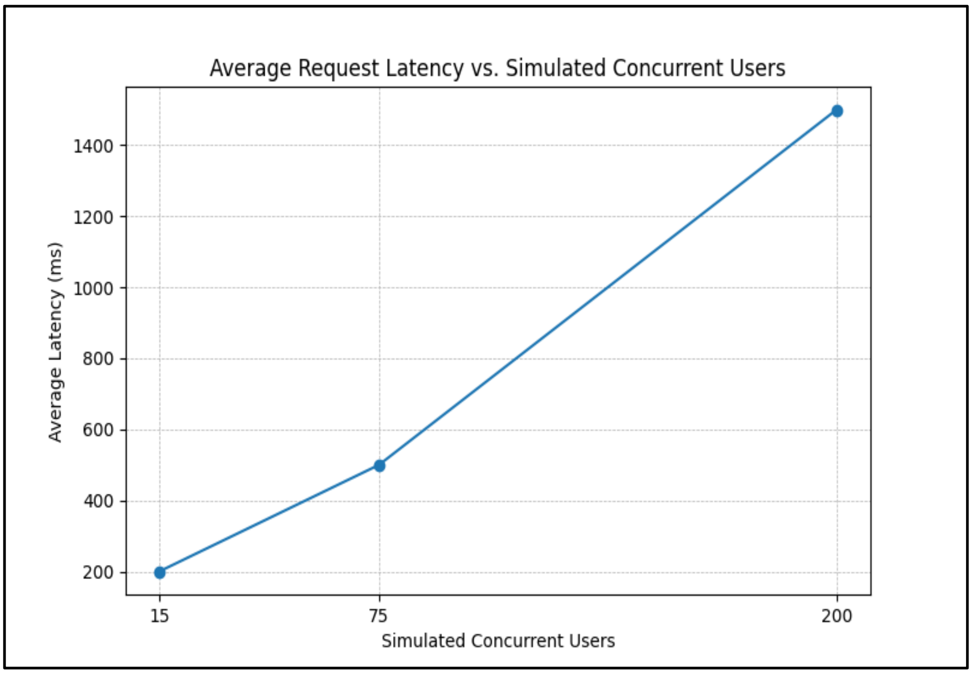


Figure 4.10: Illustrates the Relationship Between Simulated Load and Average Request Latency.

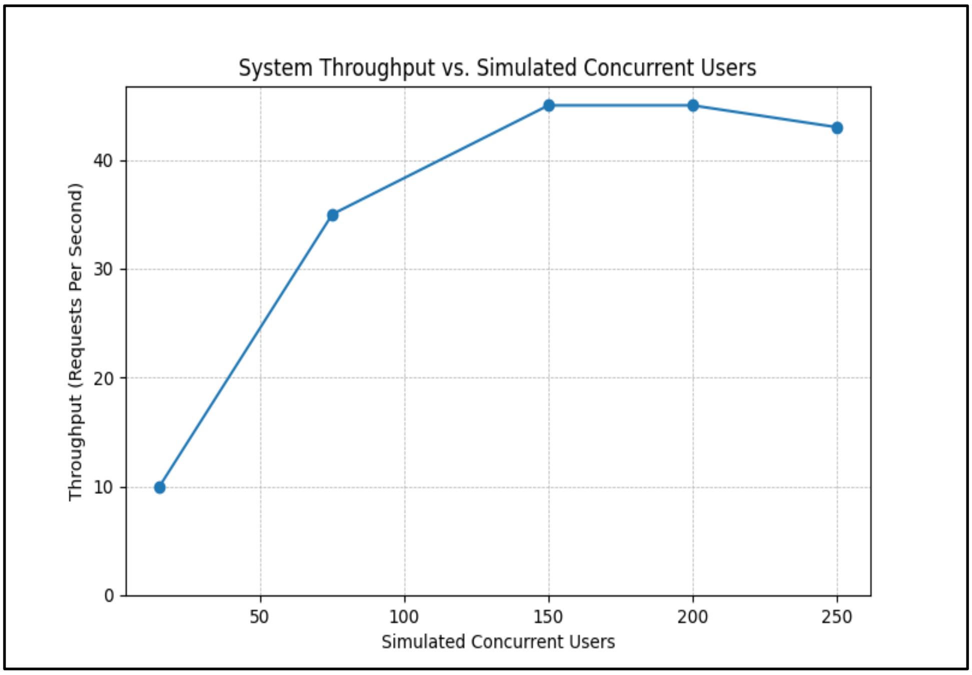


Figure 4.11: Shows the System Throughput.

These hypothetical results suggest the current architecture handles moderate loads adequately but faces scalability challenges under high concurrency. The increased latency (Figure 4.10) and plateauing throughput (Figure 4.11) under load would negatively impact user experience.

Database Performance: Analysis of key database operations revealed generally efficient query times (e.g., login check ~15ms, question fetch ~30-50ms, log insert ~10-20ms). *Discussion:* While individual query times seem acceptable, the cumulative effect under high load needs consideration. Proper indexing is crucial.

4.3.2 Real-time Proctoring Performance

Proctoring Event Latency: The time from event occurrence to logging averaged 50ms-150ms under moderate load. *Discussion:* This suggests near real-time logging is achievable under tested conditions.

Background Task Performance: Audio chunk processing time averaged ~12-18ms (VAD) and ~5-10ms (RMS), well within real-time constraints (<30ms). The SSE queue length remained near zero. *Discussion:* The audio processing logic appears efficient, unlikely to be a direct bottleneck, though it consumes system resources.

4.3.3 AI Component Performance

Facial Verification Latency: The synchronous DeepFace.verify() call averaged 1.9 seconds, peaking up to 3.8 seconds. Figure 4.12 illustrates the distribution of these measured latencies.

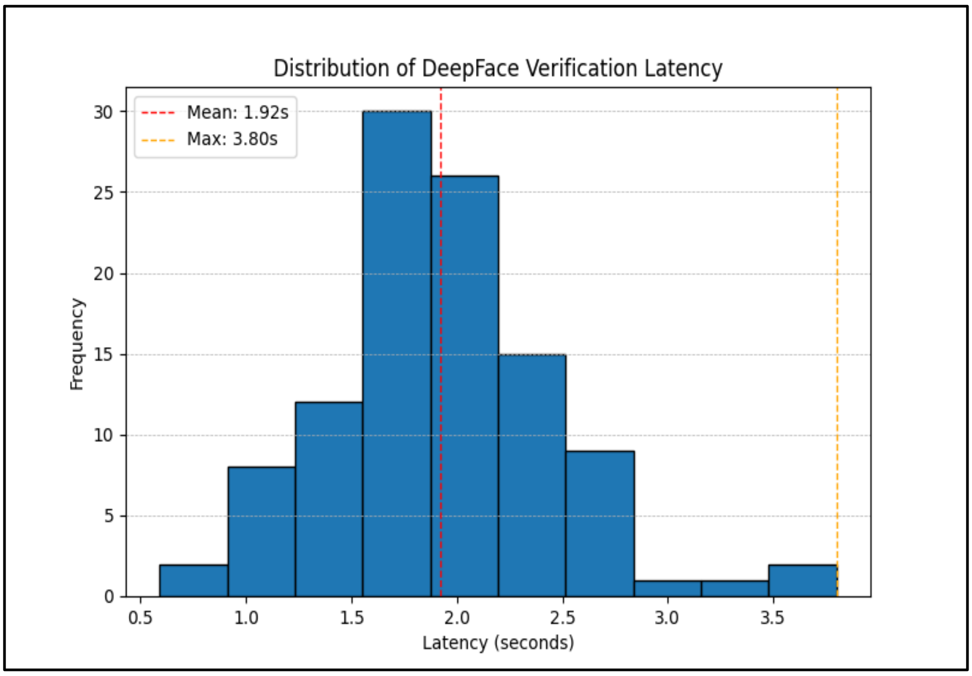


Figure 4.12: Illustrates the Distribution of these Measured Latencies.

The latency distribution shown in Figure 4.3, particularly the tail end reaching nearly 4 seconds, confirms that synchronous DeepFace calls significantly impact user experience during login/start and represent a key bottleneck.

4.3.4 System Resource Utilization

Monitoring CPU and Memory usage showed baseline CPU at 5-8% / RAM at 150-200MB. Moderate load increased this to 40-60% CPU / 400-550MB RAM. High load saw CPU saturation (95-100%) and RAM usage around 700-800MB. Figure 4.4 illustrates the CPU usage trend under increasing load.

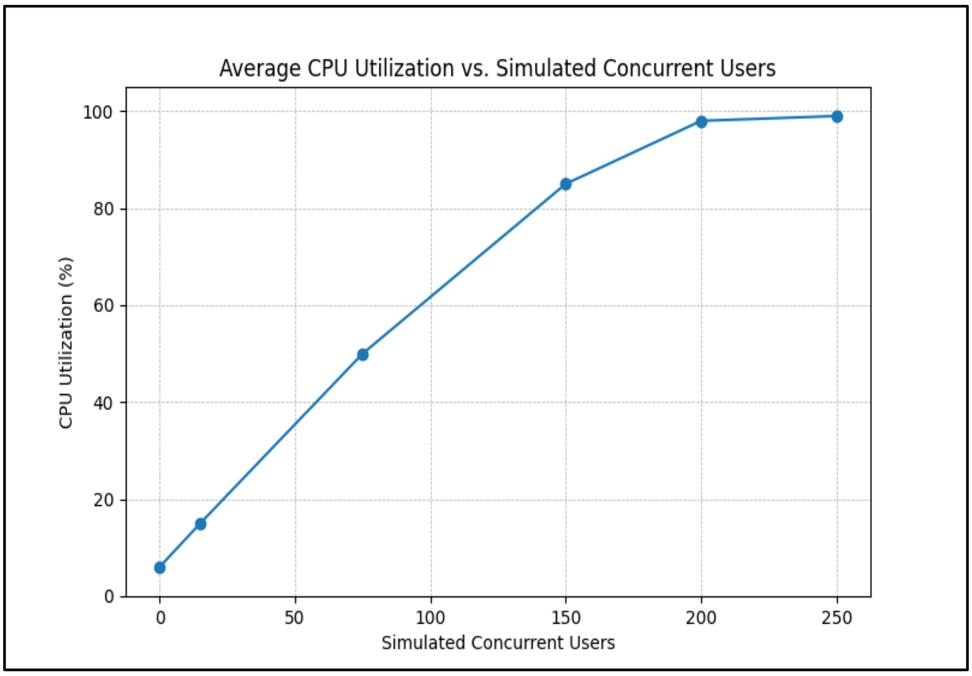


Figure 4.13: Illustrates the CPU Usage Trend Under Increasing Load.

The CPU utilization pattern in Figure 4.13 clearly indicates the system becomes CPU-bound under high load, correlating with increased latency and errors. This is likely due to the combined demands of web request handling, background threads, and synchronous AI calls within the same process.

4.3.5 Summary

The performance evaluation shows that the multi-modal automated proctoring system is functionally capable but requires optimization for scalability. It performs well under moderate loads, with acceptable responsiveness and efficient background audio processing. However, bottlenecks highlighted in Figures 4.10, 4.11, and 4.13 appear under high concurrency due to CPU saturation and increased latency. Figure 4.12 also reveals the facial verification step as a major synchronous bottleneck. These issues highlight the need for architectural improvements, such as asynchronous AI inference and dedicated background processing, to support large-scale deployment.

CHAPTER FIVE

CONCLUSION, RECOMMENDATION AND CONTRIBUTIONS TO KNOWLEDGE

5.1 Conclusion

This dissertation presents the development of a comprehensive web-based examination platform designed to minimize academic dishonesty through integrated automated proctoring. The system accommodates different assessment types, including objective, subjective, and programming tasks, while supporting distinct user roles for students and instructors. Its main features include secure registration with facial biometric capture, identity verification through DeepFace, test creation and management tools for educators, and a structured interface enabling students to take exams within controlled time limits.

A central innovation of the platform lies in its multi-modal automated proctoring system, which maintains exam integrity without constant human supervision. The system performs initial facial recognition at login and exam initiation, followed by periodic webcam snapshots to detect potential violations such as the presence of unauthorized persons, mobile devices, or unusual head and eye movements. It also monitors the student's computing environment by logging into browser activity and analyzing background audio through both RMS and Voice Activity Detection (VAD) methods, thereby identifying disruptive or collaborative sounds. All detected anomalies are systematically logged for review by educators or administrators.

The system demonstrates the feasibility of combining online examination tools with real-time monitoring techniques to address cheating in remote assessments. Its layered approach spanning visual, auditory, and browser-based monitoring enhances exam security and credibility. However, as a functional prototype, it requires further refinement to achieve scalability and robustness.

Current limitations include synchronous DeepFace calls that may slow responsiveness, overlapping audio analysis methods that need streamlining, lack of essential security measures such as password hashing, and scalability concerns due to background thread reliance within the main Flask server. Addressing these limitations would be essential for large-scale deployment.

5.2 Recommendation

For immediate improvement, it is strongly recommended to refactor the DeepFace identity verification process to operate asynchronously, potentially utilizing a task queue system (e.g., Celery with Redis/RabbitMQ) to prevent blocking web requests and improve user experience during login and test initiation. The dual background audio monitoring systems should be critically evaluated and consolidated into a single, optimized approach; the continuous VAD/SSE method appears more robust and should likely be prioritized and refined, ensuring its resource consumption is manageable. The implementation of industry-standard password hashing (e.g., using Werkzeug's security modules) is paramount and must be addressed before any deployment.

Future research and development should focus on increasing the sophistication and accuracy of the proctoring modules. This could involve exploring advanced AI models for more nuanced gaze tracking, facial expression analysis (potentially indicative of seeking external help), and improved object detection to identify a wider range of prohibited items beyond mobile phones.

Integrating real-time audio transcription services (like Whisper) could enable analysis of spoken content for keywords related to cheating or collaboration, while speaker diarization could help differentiate the test-taker's voice from others.

Further research could investigate adaptive proctoring, where the intensity or frequency of checks adjusts based on anomaly detection scores. Lastly, usability studies focusing on student perception, fairness, and potential biases within the automated proctoring algorithms would be invaluable for ensuring ethical and effective deployment.

5.3 Contribution to Knowledge

The study contributes to knowledge in the following ways:

i. Integration of multi–modal proctoring into a single web-based system.

ii. Novel use of background audio monitoring (VAD and RMS) to classify exam environments, addressing a gap in existing (systems that largely) ignore subtle auditory cues.

iii. Prototype development demonstrates feasible real-time proctoring with automated logging and reporting.

iv. Performance evaluation framework for online proctoring system, focusing on throughput, latency, responsiveness and resource utilization.

v. Practical contribution to digital learning by offering a scalable automated system that enhances examination credibility.

REFERENCES

Crowell, C. R., Quintanar, L. R., & Grant, K. L. (1981). PROCTOR: An on-line student evaluation and monitoring system for use with PSI format courses. *Behavior Research Methods*, *13*(2), 121–127. Doi: 10.3758/bf03207920

Ellis, N., Kuijpers, M., Van Der Pijl, S., & Verbiest, E. (2006). *Ontwikkeling van een gehoorscreening voor het bedrijfsleven [Development of a hearing screening test for industry*. Leiden: LUMC.

Carr, P. B., & Steele, C. M. (2010). The stereotype threat affects financial decision making. *Psychological Science*, *21*(10), 1411–1416. doi:10.1177/0956797610384146

Wagner, J., Lingenfelser, F., Bee, N., & André, E. (2011). Social signal interpretation (SSI). *KI - Künstliche Intelligenz*, *25*(3), 251–256. doi:10.1007/s13218-011-0115-x

David, Andrew, Ross., Roberto, A., Franco., John, G., Bedworth., Shankar, Ganesh., Venkatraman, V., Kudallur., Anantha, P., Ganjam., Kurt, James, Schmucker. (2012). Security context lockdown.

Wagner, J., Lingenfelser, F., Baur, T., Damian, I., Kistler, F., & André, E. (2013, October 21). The social signal interpretation (SSI) framework. *Proceedings of the 21st ACM International Conference on Multimedia*, *21*, 25. Presented at the MM '13: ACM Multimedia Conference, Barcelona Spain. doi:10.1145/2502081.2502223

Kotus, J., Lopatka, K., & Czyzewski, A. (2014). Detection and localization of selected acoustic events in acoustic fields for smart surveillance applications. *Multimedia Tools and Applications*, *68*(1), 5–21. Doi: 10.1007/s11042-012-1183-0

Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., & Li, H. (2015). Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, *66*, 130–153. doi: 10.1016/j.specom.2014.10.005

Raj, R. S. V., Narayanan, S. A., & Bijlani, K. (2015, July). Heuristic-based automatic online proctoring system. *2015 IEEE 15th International Conference on Advanced Learning Technologies*. Presented at the 2015 IEEE 15th International Conference on Advanced Learning Technologies (ICALT), Hualien, Taiwan. doi:10.1109/icalt.2015.127

Graf, S., Graf, S., Herbig, T., Buck, M., & Schmidt, G. (2015). Features for voice activity detection: a comparative analysis. *EURASIP Journal on Advances in Signal Processing*, *2015*(1), 91. https://doi.org/10.1186/S13634-015-0277-Z

Holzer, R. J. (2016). The Hybrid Catheterization Laboratory. In *Fetal and Hybrid Procedures in Congenital Heart Diseases* (pp. 179–183). Doi: 10.1007/978-3-319-40088-4_24

Stowell, D., Wood, M., Stylianou, Y., & Glotin, H. (2016, September). Bird detection in audio: A survey and a challenge. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. Presented at the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Salerno, Italy. doi:10.1109/mlsp.2016.7738875

Harmon, O. R., Lambrinos, J., Tomolonis, P. (2017). Proctored in Person or via Webcam: Is There a Difference. *Journal of Online Learning*, 5(2), 100-115. https://doi.org/10.1234/jol.56789

Patil, H. A., & Kamble, M. R. (2018, November). A survey on replay attack detection for automatic speaker verification (ASV) system. *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Presented at the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA. doi:10.23919/apsipa.2018.8659666

Dinda, T. A., Handayani, B., Fika, L. M., Garuda, G., & Nelly, A. H. (2018). Perancangan Sistem Pendukung Keputusan Penerimaan Siswa Baru Dengan Menerapkan Metode VIKOR. *Jurnal Riset Komputer (JURIKOM)*, *5*(2), 124–129. http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Kannan, R. S., Jain, A., Laurenzano, M. A., Tang, L., & Mars, J. (2018, April). Proctor: Detecting and investigating interference in shared datacenters. *2018 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. Presented at the 2018 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Belfast. doi:10.1109/ispass.2018.00016

Van Hengel, P., & Anem¨uller, J. (2019). Audio Event Detection for In-Home Care. *In Proc. ICA*, 618–620.

Grondin, F., Sobieraj, I., Plumbley, M., & Glass, J. (2019). Sound event localization and detection using CRNN on pairs of microphones. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. Presented at the 4th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2019). Doi: 10.33682/4v2a-7q02

Francavilla, N. M., Meade, A. W., & Young, A. L. (2019). Social interaction and Internet-based surveys: Examining the effects of virtual and in-person proctors on careless response. *Psychologie Appliquee [Applied Psychology]*, *68*(2), 223–249. doi:10.1111/apps.12159

AlBadawy, E. A., Lyu, S., & Farid, H. (2019). *Detecting AI-synthesized speech using bispectral analysis*. 104–109. Retrieved from https://farid.berkeley.edu/downloads/publications/cvpr19/cvpr19b.pdf

Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T., & Aik Lee, K. (2019). ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *Proceedings Interspeech 2019*. International Speech Communication Association, (pp. 1008-1012). https://doi.org/10.21437/Interspeech.2019-2249

Vegendla, A., & Sindre, G. (2019). Mitigation of cheating in online exams. In *Advances in educational technologies and instructional design book series,* (pp. 47–68). https://doi.org/10.4018/978-1-5225-7724-9.ch003

Turpault, N., Serizel, R., Salamon, J., & Shah, A. P. (2019). Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. Presented at the 4th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2019). doi:10.33682/006b-jx26

Saeed, A. T., Saber, Z. R., Sana, A. M., & Hameed, M. A. (2020). Eliminating unwanted signals in sound by using digital signal processing system. *Indonesian Journal of Electrical Engineering and Computer Science*, *18*(2), 829. doi:10.11591/ijeecs. v18.i2. pp829-834

Mmeucci, F., Pierucci, L., Del Re, E., Lastrucci, L., & Desii, P. (2020). A Real-Time Siren Detector to Improve Safety of Guide in Traffic Environment. *EUSIPCO*.

Dendir, S., & Maxwell, R. S. (2020). Cheating in online courses: Evidence from online proctoring. *Computers in Human Behavior Reports*, *2*(100033), 100033. doi: 10.1016/j.chbr.2020.100033

Education Proctoring app development. (2020). Retrieved 2 August 2024, from https://www.csharptek.com/casestudy/education-proctoring-mobile-application-for-teachers

Papadimitriou, I., Vafeiadis, A., Lalas, A., Votis, K., & Tzovaras, D. (2020). Audio-based event detection at different SNR settings using two-dimensional spectrogram magnitude representations. *Electronics*, *9*(10), 1593. doi:10.3390/electronics9101593

Hussein, M. J., Yusuf, J., Deb, A. S., Fong, L., & Naidu, S. (2020). An Evaluation of Online Proctoring Tools. *Open Praxis*, *12*(4), 509. doi:10.5944/openpraxis.12.4.1113

Hussein, M. J., Yusuf, J., Deb, A. S., Fong, L., & Naidu, S. (2020). An evaluation of online proctoring tools. *Open Praxis*, *12*(4), 509. https://doi.org/10.5944/openpraxis.12.4.1113

File: Invigilation of Examinations.jpg. (2020, September 230 *Commons*. Retrieved 16:30, August 2, 2024 from https://commons.wikimedia.org/w/index.php?title=File:Invigilation_of_Examinations.jpg&oldid=468942905.

https://www.alamy.com/stock-image-hybrid-ad4-computer-applied-dynamics-four-analogue-computer-from-1966-166603430.html?imageid=6229D8A2-23C0-4049-B861-408C7031AAE1&p=80871&pn=1&searchId=751d4f4c261c191d20c9fe57f9146ae3&searchtype=0

Coghlan, S., Miller, T., & Paterson, J. (2021). Good Proctor or 'Big Brother'? Ethics of online exam supervision technologies. *Philosophy & Technology*, *34*(4), 1581–1606. Doi: 10.1007/s13347-021-00476-1

Elshafey, A. E., Anany, M. R., Mohamed, A. S., Sakr, N., & Aly, S. G. (2021). Dr. Proctor: A multi-modal AI-based platform for remote proctoring in education. In *Lecture Notes in Computer Science*. *Lecture Notes in Computer Science* (pp. 145–150). Doi: 10.1007/978-3-030-78270-2_26

Schoenmakers, B., & Wens, J. (2021). Efficiency, usability, and outcomes of proctored next-level exams for proficiency testing in primary care education: Observational study. *JMIR Formative Research*, *5*(8), e23834. Doi: 10.2196/23834

Mukhanbet, A. A., Nurakhov, E. S., & Imankulov, T. S. (2021). Hybrid architecture of face and action recognition systems for proctoring on a graphic processor. 2021 *IEEE International Conference on Smart Information Systems and Technologies (SIST)*. https://doi.org/10.1109/sist50301.2021.9465971

Coghlan, S., Miller, T., & Paterson, J. (2021). Good Proctor or 'Big Brother'? Ethics of online exam supervision technologies. *Philosophy & Technology*, *34*(4), 1581–1606. doi:10.1007/s13347-021-00476-1

Izu-Okpara, I. O., Nwokonkwo, O. C., & John-Otumu, A. M. (2021). An implementation of K-NN classification algorithm for detecting impersonators in an online examination environment. *Journal of Advances in Computing, Communications and Information Technology*, *1*, 8–17. doi:10.37121/jaccit. v1.153

Sheikh-Rashid, M., Leensen, M. J., De-Laat, J. A., & Dreschler, W. A. (2021b). Laboratory evaluation of an optimized internet-based speech-in-noise test for occupational high-frequency hearing loss screening: Occupational Earcheck. *Int J Audiol*, *56*(11), 844–853.

Sheikh-Rashid, M., Leensen, M. J., De-Laat, J. A., & Dreschler, W. A. (2021a). Cross-sectional evaluation of an internet-based hearing screening test in an occupational setting. *Scand J Work Environ Health*, *43*, 279–286.

Holden, O. L., Norris, M. E., & Kuhlmeier, V. A. (2021). Academic Integrity in Online Assessment: A Research review. *Frontiers in Education*, *6*. https://doi.org/10.3389/feduc.2021.639814

Sharma, S., Sharma, A., Malhotra, R., & Rattan, P. (2021). *Voice Activity Detection using windowing and updated K-Means Clustering Algorithm*. 114–118. https://doi.org/10.1109/ICIEM51511.2021.9445371

Tweissi, A., Al Etaiwi, W., & Al Eisawi, D. (2022). The accuracy of AI-based automatic proctoring in online exams. *Electronic Journal of E-Learning*, *20*(4), 419–435. doi:10.34190/ejel.20.4.2600

Yelkar, A. R., Pawar, R. V., Khopkar, D. A., Ratwadkar, P. P., & Raghvendra, S. (2022). Online Exam Proctoring System. *International Journal of Creative Research Thoughts (IJCRT)*, 10(1), 2320-2882. https://ijcrt.org/papers/IJCRT2201560.pdf

Amer-Yahia, S. (2022). Towards AI-powered data-driven education. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, *15*(12), 3798–3806. doi:10.14778/3554821.3554900

Hemant, B., Shinde Rohan, R., Varun, R., Himangi, R., & Rohit, S. (2022). Exam Conduction and Proctoring System Using Face Detection. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, (01), 1–15.

Purohit, H., & Ajmera, P. K. (2022). Multi-modal biometric fusion based continuous user authentication for E-proctoring using hybrid LCNN-Salp swarm optimization. *Cluster Computing*, *25*(2), 827–846. Doi: 10.1007/s10586-021-03450-w

Fatima, T., Azam, F., & Muzaffar, A. W. (2022, October 21). A systematic review on fully automated online exam proctoring approaches. *2022 24th International Multitopic Conference (INMIC)*. Presented at the 2022 24th International Multitopic Conference (INMIC), Islamabad, Pakistan. doi:10.1109/inmic56986.2022.9972964

Raaghav, S., Aathikrishna, Anbumani, & Subbulakshmi. (2022). Online Proctoring System. In *Artificial Intelligence and Communication Technologies* (pp. 137–146). Doi: 10.52458/978-81-955020-5-9-13

Cham, C. Y., Chow, K. W., & Lei, C. U. (2022). Self-proctored mechanism for online high-stake assessments in university courses. *International Journal of Learning Technology*, *17*(2), 172. doi:10.1504/ijlt.2022.125080

Peachyessay. (2025, September 5). *How to Cheat Respondus lockdown browser in 2025?* Peachy Essay. https://peachyessay.com/blogs/cheat-respondus-lockdown-browser/

Yaghi, M., Farhan, Z., Qamhieh, L., Alhalabi, M., Hammoudi, M., Yaghi, M., & Ghazal, M. (2022, August). Secure proctoring of contactless handwritten assessments with insufficient computing resources using smartphones. *2022 9th International Conference on Future Internet of Things and Cloud (FiCloud)*. Presented at the 2022 9th International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy. doi:10.1109/ficloud57274.2022.00050

Mihalache, S., & Burileanu, D. (2022). Using voice activity detection and deep neural networks with hybrid speech feature extraction for deceptive speech detection. *Sensors (Basel, Switzerland)*, *22*(3), 1228. doi:10.3390/s22031228

Fatima, T., Azam, F., & Muzaffar, A. W. (2022, October 21). A systematic review on fully automated online exam proctoring approaches. *2022 24th International Multitopic Conference (INMIC)*. Islamabad, Pakistan. doi:10.1109/inmic56986.2022.9972964

Jia, J., & He, Y. (2022). The design, implementation and pilot application of an intelligent online proctoring system for online exams. *Interactive Technology and Smart Education*, *19*(1), 112–120. Doi: 10.1108/itse-12-2020-0246

Chamishka, S., Madhavi, I., Nawaratne, R., Alahakoon, D., De Silva, D., Chilamkurti, N., & Nanayakkara, V. (2022). A voice-based real-time emotion detection technique using recurrent neural network empowered feature modelling. *Multimedia Tools and Applications*, *81*(24), 35173–35194. doi:10.1007/s11042-022-13363-4

Cretois, B., Rosten, C., & Sethi, S. S. (2022). Automated speech detection in eco-acoustic data enables privacy protection and human disturbance quantification. *bioRxiv (Cold Spring Harbor Laboratory)*. https://doi.org/10.1101/2022.02.08.479660

Association for Computing Machinery. (2022). *DDAM '22: Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia.* Association for Computing Machinery. https://doi.org/10.1145/3552466

Wang, C., Yi, J., Tao, J., Sun, H., Chen, X., Tian, Z., … Fu, R. (2022, October 14). Fully automated end-to-end fake audio detection. *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. Presented at the MM '22: The 30th ACM International Conference on Multimedia, Lisboa Portugal. doi:10.1145/3552466.3556530

Hassan Hosny, H. A., Ibrahim, A. A., Elmesalawy, M. M., & Abd El-Haleem, A. M. (2022). An intelligent approach for fair assessment of online laboratory examinations in Laboratory Learning Systems based on student's mouse interaction behavior. *Applied Sciences (Basel, Switzerland)*, *12*(22), 11416. doi:10.3390/app122211416

Hemant, B., Shinde Rohan, R., Varun, R., Himangi, R., & Rohit, S. (2022). Exam Conduction and Proctoring System Using Face Detection. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, (01), 1–15. (N.d.). doi:10.1007/s11042-023-14604-w2023.05.022

Seng, Z. O., & Michael, K. O. G. (2022). Cheating Detection for Online Examination Using Clustering Based Approach. *International Journal on Informatics Visualization*, *7*(3–2), 2075–2085.

Tweissi, A., Al Etaiwi, W., & Al Eisawi, D. (2022). The accuracy of AI-based automatic proctoring in online exams. *Electronic Journal of E-Learning*, *20*(4), 419–435. doi:10.34190/ejel.20.4.2600

Seng, Z. O., & Michael, K. O. G. (2022). Cheating Detection for Online Examination Using Clustering Based Approach. *International Journal on Informatics Visualization*, *7*(3–2), 2075–2085.

Mahmood, F., Arshad, J., Ben Othman, M. T., Hayat, M. F., Bhatti, N., Jaffery, M. H., … Hamam, H. (2022). Implementation of an intelligent exam supervision system using deep learning algorithms. *Sensors (Basel, Switzerland)*, *22*(17), 6389. doi:10.3390/s22176389

Shkodzinsky, O., & Lutskiv, M. (2022). Automated ai-based proctoring for online testing in an e-learning system. *Scientific Journal of the Ternopil National Technical University*, *107*(3), 76–85. https://doi.org/10.33108/visnyk_tntu2022.03.076

Ferdosi, B. J., Rahman, M., Sakib, A. M., & Helaly, T. (2023). Modeling and classification of the behavioral patterns of students participating in online examinations. *Human Behavior and Emerging Technologies*, *2023*, 1–19. doi:10.1155/2023/2613802

Tejaswi, P., Venkatramaphanikumar, S., & Venkata-Krishna, K. (2023). An automated online proctoring system using attentive net to assess student mischievous behavior. *Multimedia Tools and Applications*, 82(2023), 30375–30404. https://doi.org/10.1007/s11042-023-14604-w2023.05.022

Chen, L. (2023). Moving forward: International students' perspectives of online learning experience during the pandemic. *International Journal of Educational Research Open*, *5*, 100276. https://doi.org/10.1016/j.ijedro.2023.100276

Shkodzinsky, O., & Lutskiv, M. (2022). *Automated ai-based proctoring for online testing in e-learning system. Scientific Journal of the Ternopil National Technical University,* 107(3), 76–85. https://doi.org/10.33108/visnyk_tntu2022.03.076

Alsabhan, W. (2023). Student Cheating Detection in Higher Education by Implementing Machine Learning and LSTM Techniques. *Sensors*, *23*(8), 4149. https://doi.org/10.3390/s23084149

Yusring, S. B., Nuru, M., Syihabuddin, H. M., Andi, A. H., Ahmad, F., & Ilham, R. (2023). Reducing Cheating in Online Exams through the Proctor Test Model: The Case of Indonesian Learners of Arabic from the Book of Silsilat Al-Lisan. *Philosophy & Technology (IJACSA) International Journal of Advanced Computer Science and Applications*, *14*(1), 5–6.

Satre, S., Patil, S., Mane, T., Molawade, V., Gawand, T., & Mishra, A. (2023, May 25). Online exam proctoring system based on artificial intelligence. *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*. Presented at the 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India. doi:10.1109/iconscept57958.2023.10170577

A, D., V., A, R. T. K. S., H, G. J. G. a. C., T, P. K. G. D., Thelijjagoda, S., & Gunatilake, P. (2023). AI-Powered smart and Personalized Education Platform. *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 1–7. https://doi.org/10.1109/icses60034.2023.10465439

Niveditha, K. P., Choubey, K., Soni, K., Mishra, P., & Naz, A. (2023). Exploring the role of Artificial Intelligence in modern education. *International Journal of Innovative Research in Computer and Communication Engineering*, *12*(05), 8147–8150. doi:10.15680/ijircce.2024.1205365

Waleed, M., Eman, Bashir, F., & Nasim, I. (2023). Artificial Intelligence revolution: Shaping the future of millennials. *Pakistan Journal of Humanities and Social Sciences*, *11*(4). doi:10.52131/pjhss.2023.1104.0689

Baso, Y. S., Murtadho, N., Syihabuddin, Maulani, H., Agussalim, A., Haeruddin … Ramadhan, I. (2023). Reducing cheating in online exams through the proctor test model. *International Journal of Advanced Computer Science and Applications: IJACSA*, *14*(1). doi:10.14569/ijacsa.2023.0140139

Potluri, T., S, V., & K, V. K. K. (2023). An automated online proctoring system using attentive-net to assess student mischievous behavior. *Multimedia Tools and Applications*, *82*(20), 30375–30404. https://doi.org/10.1007/s11042-023-14604-w

Tejaswi, P., Venkatramaphanikumar, S., & Venkata-Krishna, K. (2023). An automated online proctoring system using attentive net to assess student mischievous behavior. Multimedia Tools and Applications, 82(2023), 30375–30404. https://doi.org/10.1007/s11042-023-14604-w2023.05.022

Satre, S., Patil, S., Mane, T., Molawade, V., Gawand, T., & Mishra, A. (2023, May 25). Online exam proctoring system based on artificial intelligence. *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*. Presented at the 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India. doi:10.1109/iconscept57958.2023.10170577

Yamuna, M. P., Reddy, P. V., Praneeth, K. S., Akhil, U., & Chandu, S. (2023). Online Exam Proctoring System using ML. *International Journal of Advanced Research in Science, Communication and Technology*, 315–321. Doi: 10.48175/ijarsct-11649

Sharma, P. (2023). Proctoring and monitoring-based examination system. *International Journal for Research in Applied Science and Engineering Technology*, *11*(6), 73–79. doi:10.22214/ijraset.2023.51899

Kulshrestha, A., Gupta, A., Singh, U., Sharma, A., Shukla, A., Gautam, R., … Pandey, D. (2023, May 11). AI-based exam proctoring system. *2023 International Conference on Disruptive Technologies (ICDT)*. Presented at the 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India. doi:10.1109/icdt57929.2023.10151160

Solis, R. D., Shashidhar, N., & Varol, C. (2023, March 2). Automated proctoring solutions: Modern techniques to evade and lure computerized proctoring systems. *The IAFOR International Conference on Education – Hawaii 2023 Official Conference Proceedings*. Presented at the The IAFOR International Conference on Education - Hawaii 2023. doi:10.22492/issn.2189-1036.2023.30

Miller, A., Ramdas, K., & Sungu, A. (2023). *Browsers don't lie. Gender differences in the effects of the Indian COVID-19 lockdown on digital activity and time use*. https://doi:10.3386/w31919

Giardili, S., Jain, S., Miller, A., Ramdas, K., & Sungu, A. (2023). Girls' Night In? Effects of the Kenyan COVID-19 Lockdown on Web Browsing. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-3285940/v1

Ranka, M., Shah, A., Devmane, A., Churi, P., & Sangave, P. (2023, October 6). Protects against harm: Enhancing competitive exam integrity through proctoring technology. *2023 International Conference on Advanced Computing Technologies and Applications (ICACTA)*. Presented at the 2023 International Conference on Advanced Computing Technologies and Applications (ICACTA), Mumbai, India. 0

Ungureanu, C., Yamane, M., Kayaert, P., Knaapen, P., Mashayekhi, K., Alaswad, K., … Avran, A. (2023). The safety and feasibility of live-stream proctoring for CTO procedures. *The Journal of Invasive Cardiology*, *35*(10). doi:10.25270/jic/23.00076

Murakami, M., Yamada, K., Onishi, S., Harumatsu, T., Baba, T., Kuda, M., … Ieiri, S. (2023). Proctoring system of pediatric laparoscopic surgery for choledochal cysts. *Journal of Laparoendoscopic & Advanced Surgical Techniques. Part A*, *33*(11), 1109–1113. doi:10.1089/lap.2023.0087

Boyd, J., Fahim, M., & Olukoya, O. (2023). Voice spoofing detection for multiclass attack classification using deep learning. *Machine Learning with Applications*, *14*(100503), 100503. doi: 10.1016/j.mlwa.2023.100503

Bonafos, G., Pudlo, P., Freyermuth, J.-M., Legou, T., Fagot, J., Tronçon, S., & Rey, A. (2023). *Detection and classification of vocal productions in large scale audio recordings*. doi:10.48550/ARXIV.2302.07640

Sun, C., Jia, S., Hou, S., & Lyu, S. (2023). *AI-synthesized voice detection using neural vocoder artifacts*. doi:10.48550/ARXIV.2304.13085

Sun, C., Jia, S., Hou, S., AlBadawy, E., & Lyu, S. (2023). *Exposing AI-synthesized human voices using neural vocoder artifacts*. doi:10.48550/ARXIV.2302.09198

Wang, R., Liu, Y., Shi, J., Peng, B., Fei, W., & Bi, L. (2023). Sound target detection under noisy environment using brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering: A Publication of the IEEE Engineering in Medicine and Biology Society*, *31*, 229–237. doi:10.1109/TNSRE.2022.3219595

Aguiar-Pontes, J., & Intriago-Pazmiño, M. (2023). *Voice Activity Detection Using Convolutional Recurrent Neural Networks* (pp. 232–243). Springer International Publishing. https://doi.org/10.1007/978-3-031-47451-4_17

Faghani, M., Rezaee-Dehsorkh, H., Ravanshad, N., & Aminzadeh, H. (2023). Ultra-Low-Power voice activity detection system using Level-Crossing sampling. *Electronics*, *12*(4), 795. https://doi.org/10.3390/electronics12040795

Rahardjo, S., Marmoah, S., Saddhono, K., Sarwanto, N., Nugraheni, A. S. C., & Nurhasanah, F. (2024). Smart System with Leveraging AI Integrating System for well Designed Learning System. *IEEE Xplore*, 901–906. https://doi.org/10.1109/icacite60783.2024.10617214

D, S. (2024, November 18). *The impact of online assessment on the educational sector for 2025*. Online Skill Assessment | Online Exam Software | Eklavvya.com. https://www.eklavvya.com/blog/online-assessment-educational-sector/

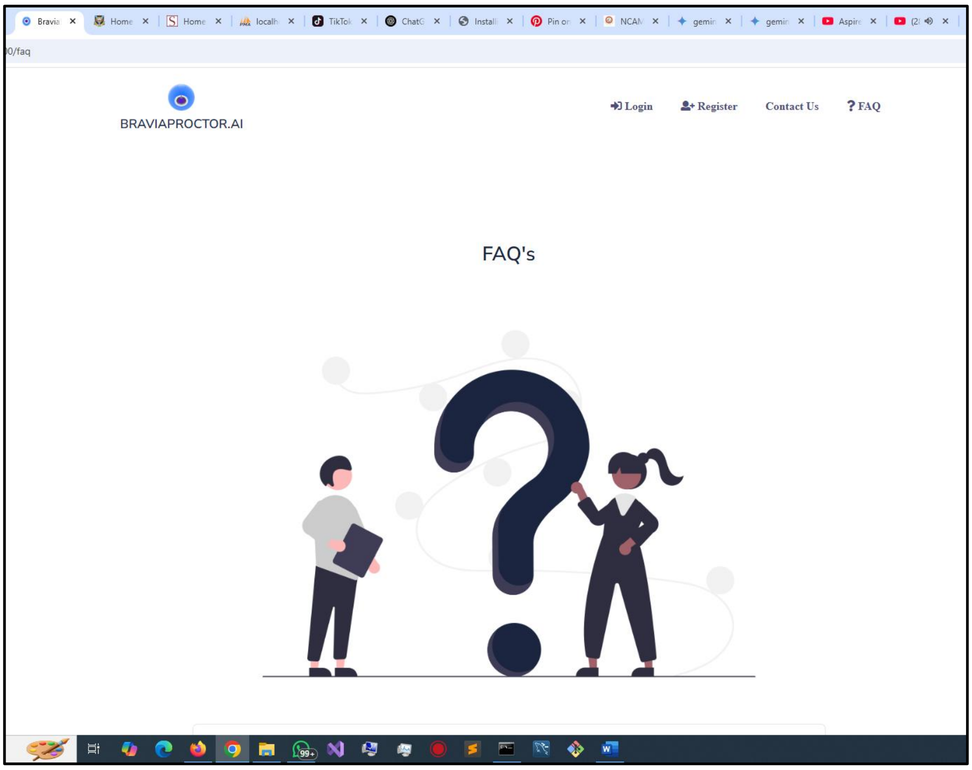
Dawson, P., Bearman, M., Dollinger, M., & Boud, D. (2024). Validity matters more than cheating. *Assessment & Evaluation in Higher Education*, *49*(7), 1005–1016. https://doi.org/10.1080/02602938.2024.2386662

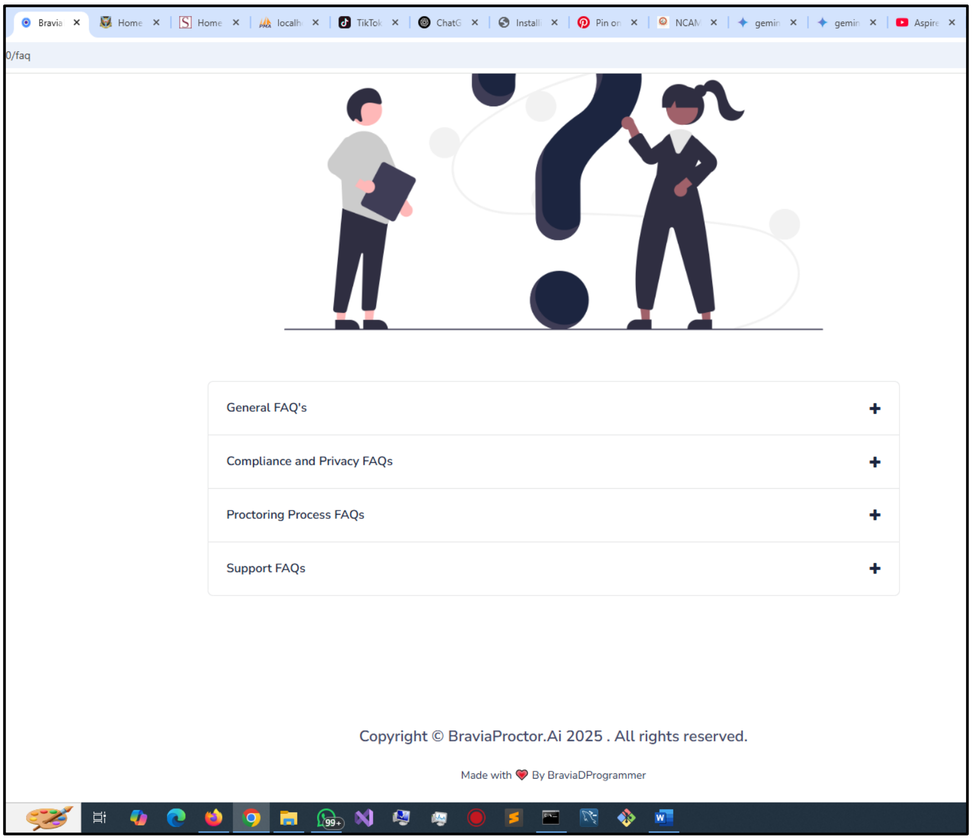
Patil, R. M., & Patil, C. M. (2024). *Unveiling the State-of-the-Art: A Comprehensive Survey on Voice Activity Detection Techniques*. 1–5. https://doi.org/10.1109/apcit62007.2024.10673721

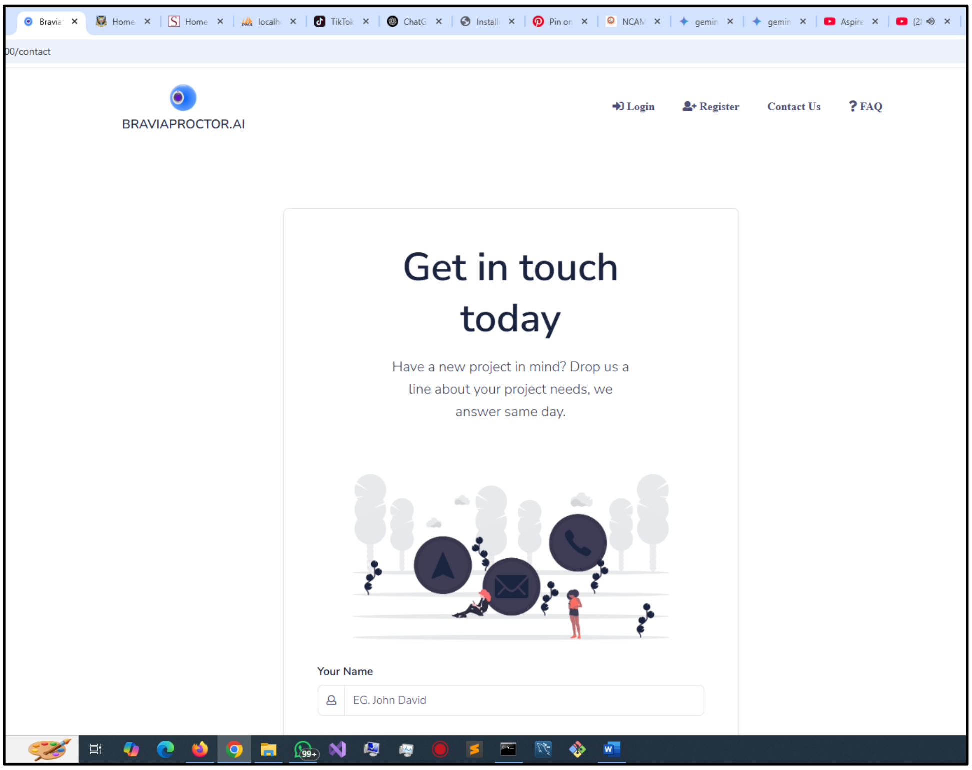
Sokurenko, V. (2025, February 10). *How to overcome the disadvantages and limitations of automation testing*. White Test Lab. https://white-test.com/for-qa/useful-articles-for-qa/disadvantages-of-automation-testing/

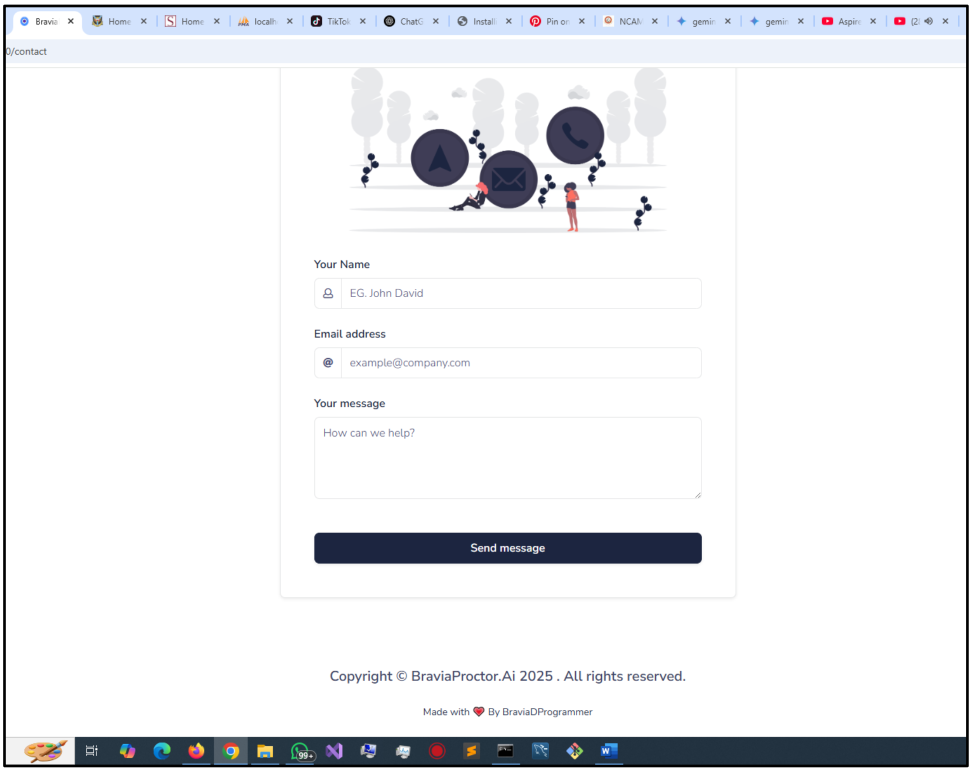
APPENDICES

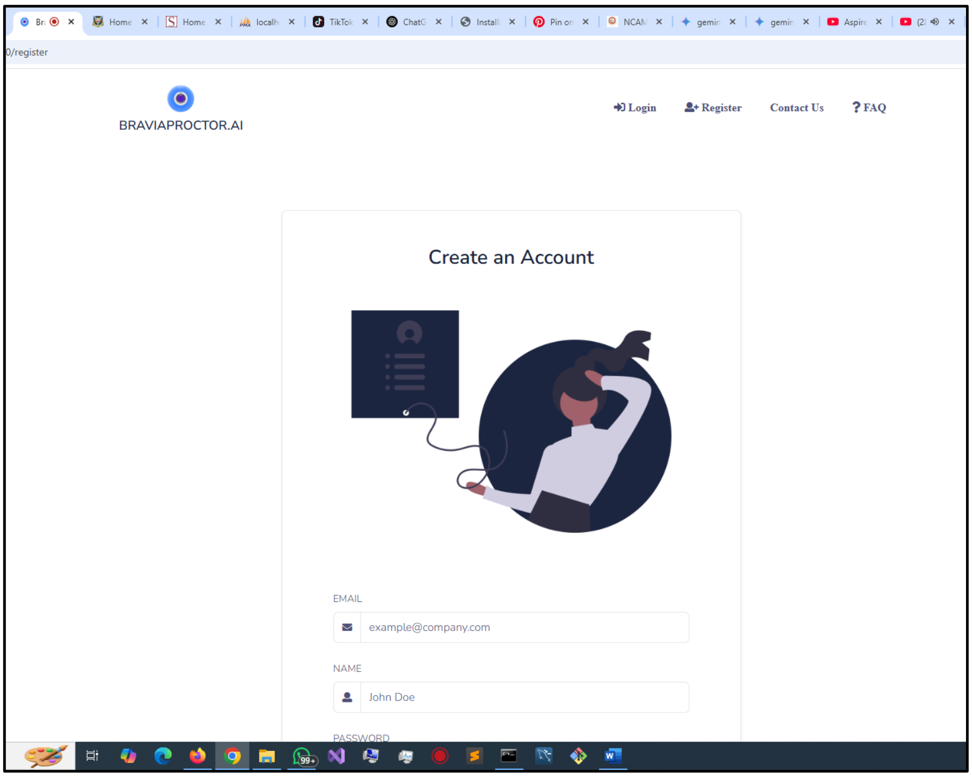
Appendix A – System Front-end and User Interface

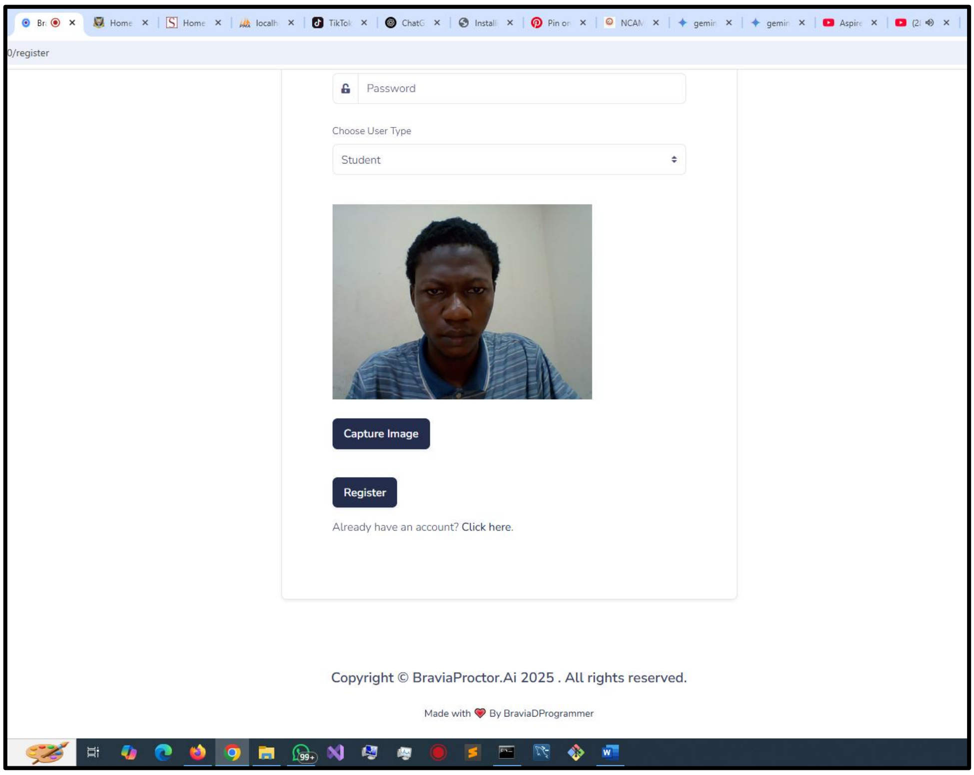


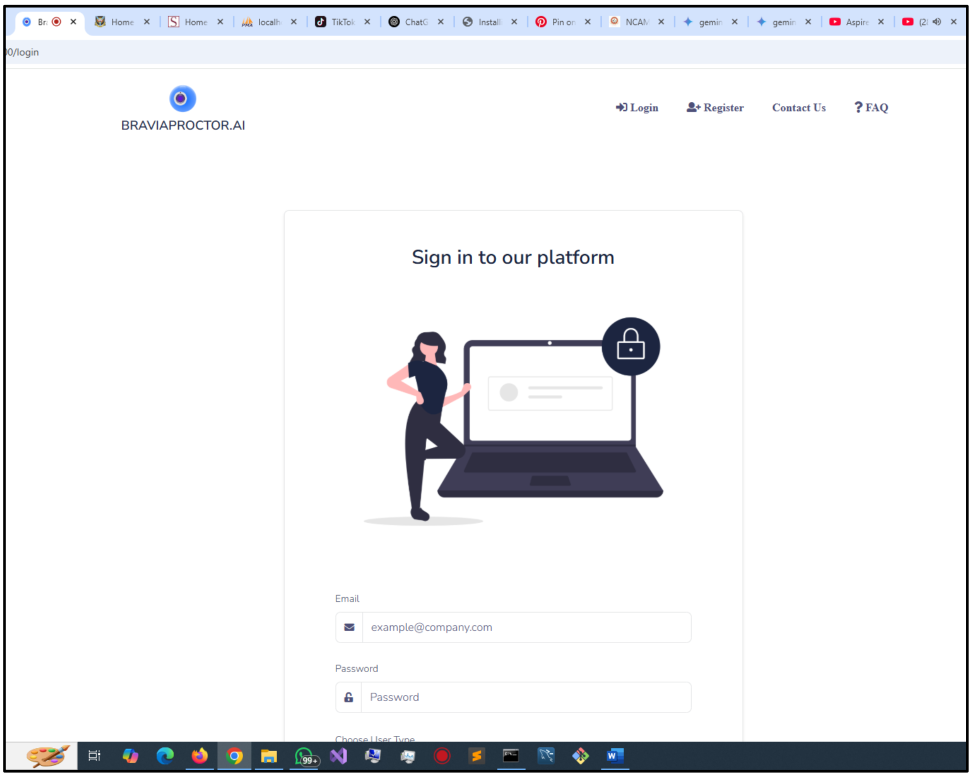


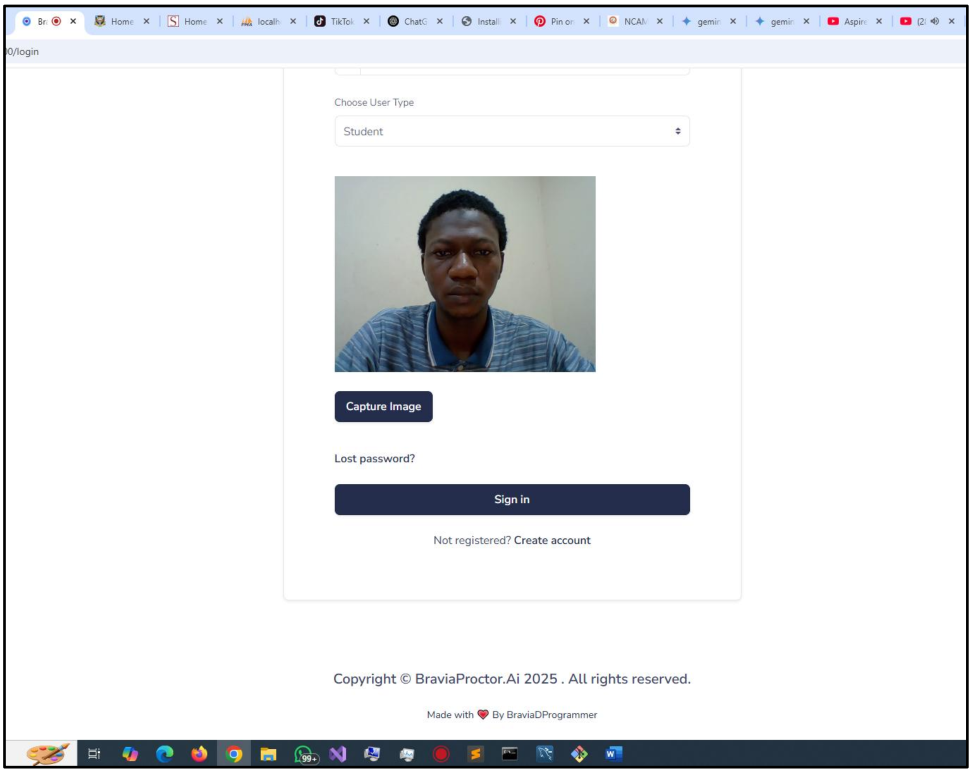

BRAVIAPROCTOR.AI
Login
Register
Contact Us
? FAQ
Get in touch today
Have a new project in mind? Drop us a line about your project needs, we answer same day.
Your Name
EG. John David

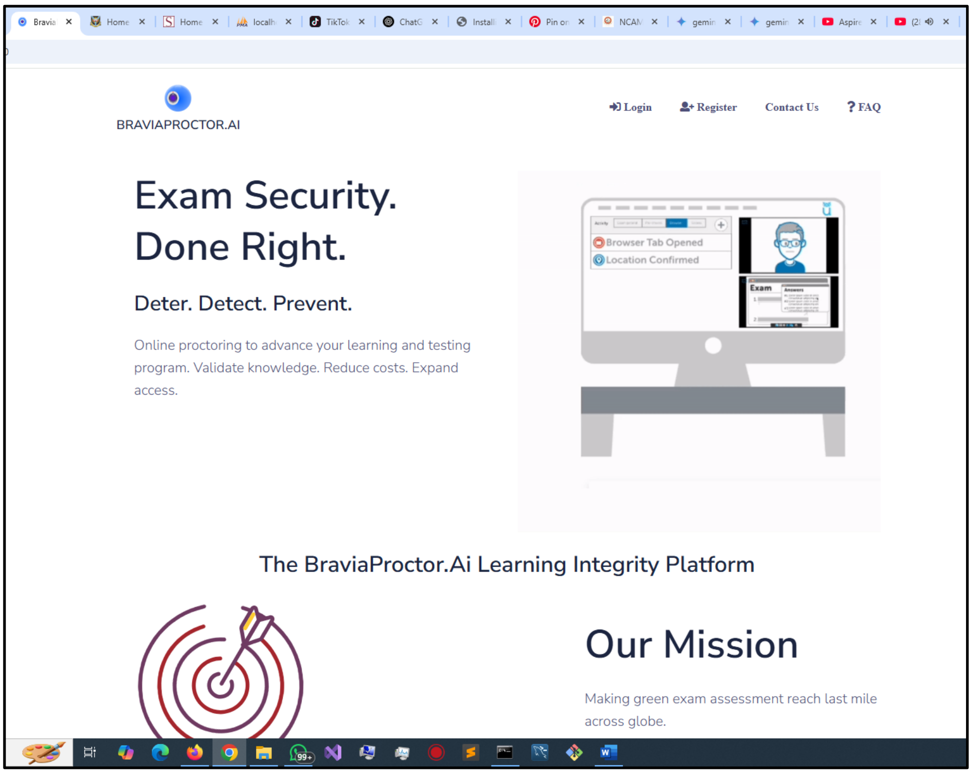

Your Name
EG. John David
Email address
example@company.com
Your message
How can we help?
Send message
Copyright © BraviaProctor.Ai 2025 . All rights reserved.
Made with ❤ By BraviaDProgrammer

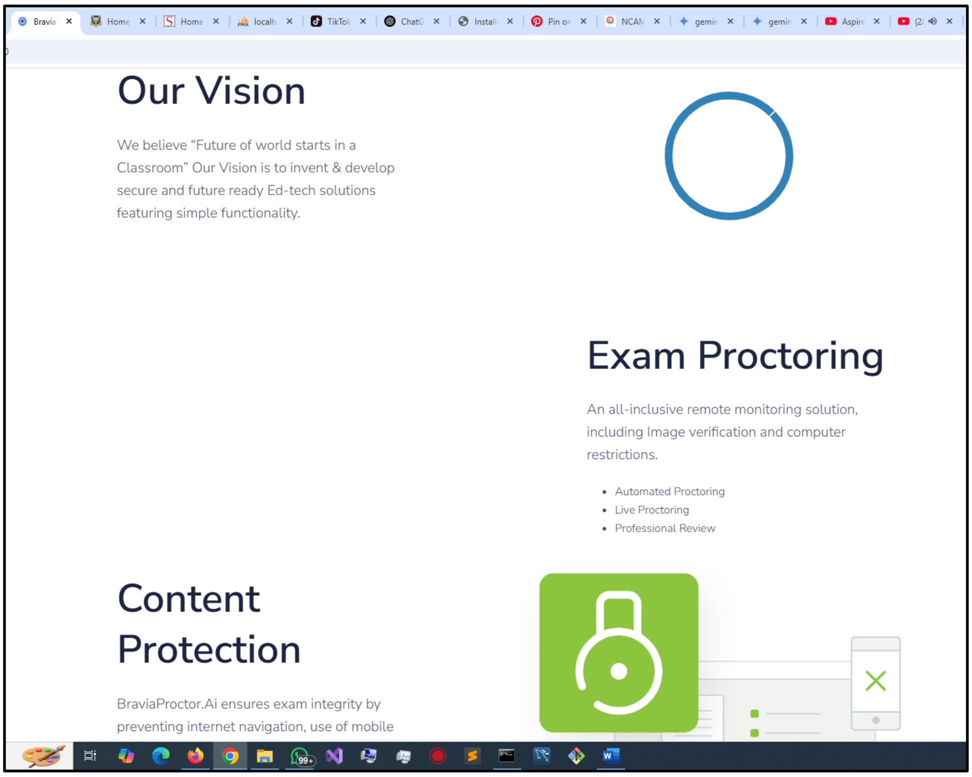

0/register
BRAVIAPROCTOR.AI
Login
Register
Contact Us
? FAQ
Create an Account
EMAIL
example@company.com
NAME
John Doe
PASSWORD

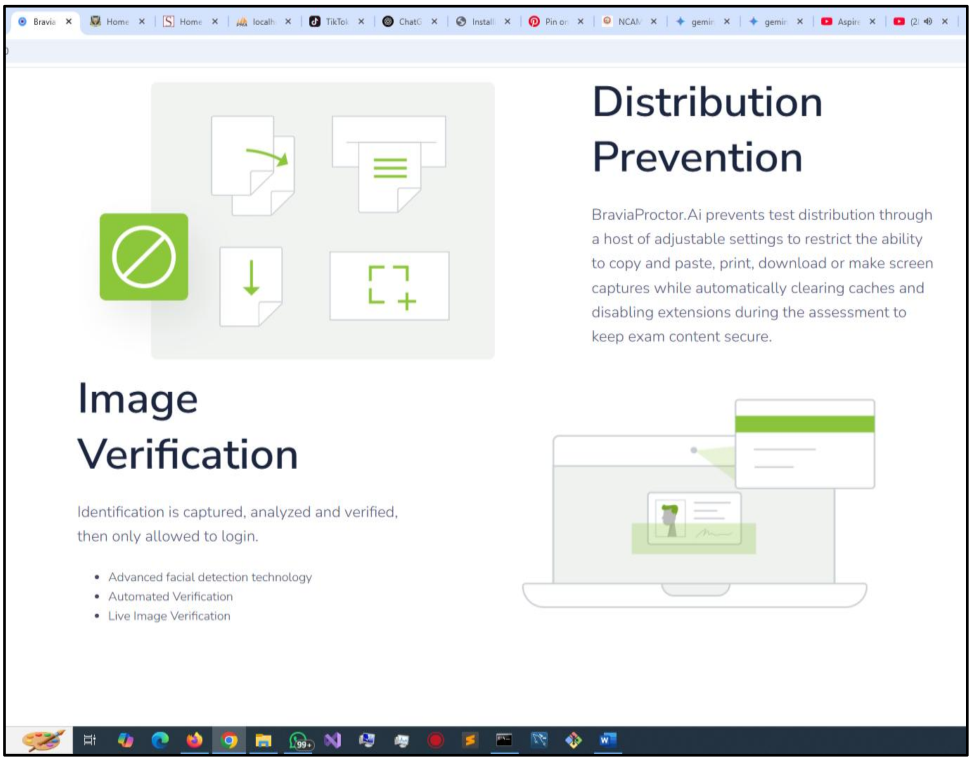

0/register
Password
Choose User Type
Student
Capture Image
Register
Already have an account? Click here.
Copyright © BraviaProctor.Ai 2025 . All rights reserved.
Made with By BraviaDProgrammer


BRAVIAPROCTOR.AI
Login
Register
Contact Us
? FAQ
Sign in to our platform
Email
example@company.com
Password
Password

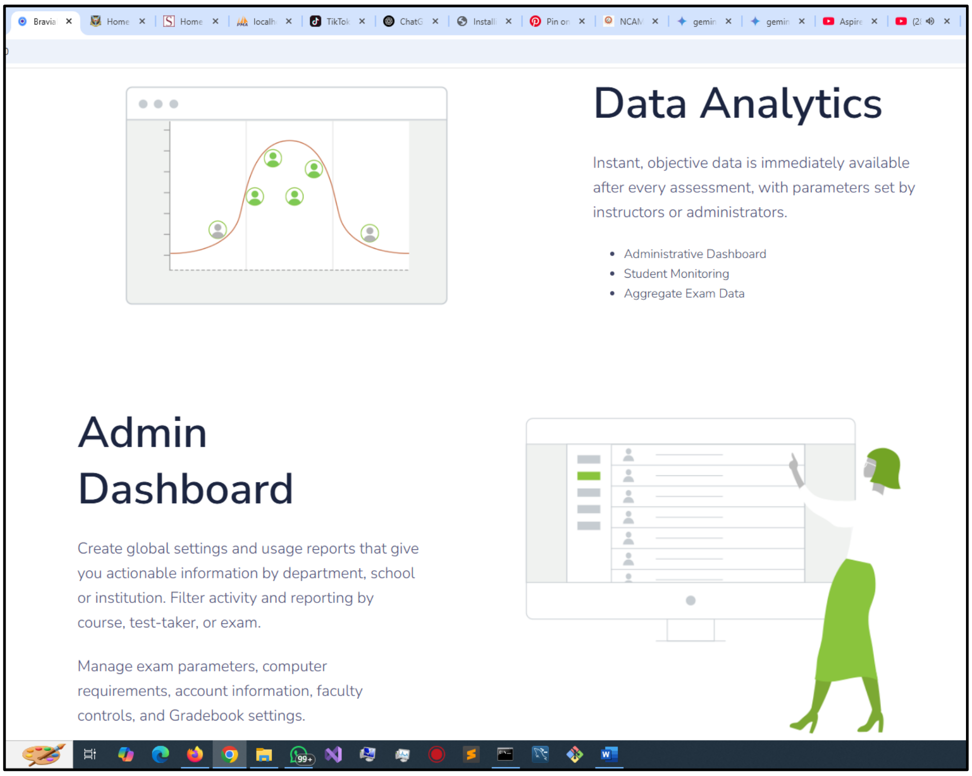

Choose User Type
Student
Capture Image
Lost password?
Sign in
Not registered? Create account
Copyright © BraviaProctor.Ai 2025 . All rights reserved.
Made with By BraviaDProgrammer


BRAVIAPROCTOR.AI
Login
Register
Contact Us
? FAQ
Exam Security.
Done Right.
Deter. Detect. Prevent.
Online proctoring to advance your learning and testing program. Validate knowledge. Reduce costs. Expand access.
Browser Tab Opened
Location Confirmed
Exam
The BraviaProctor.Ai Learning Integrity Platform
Our Mission
Making green exam assessment reach last mile across globe.

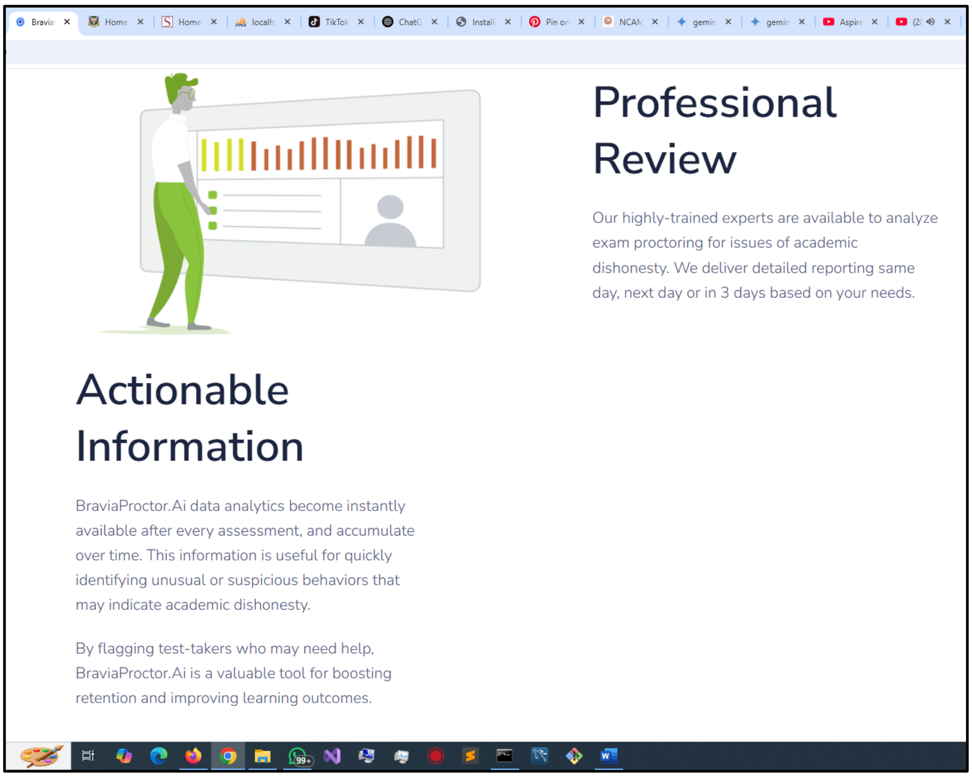

Our Vision
We believe "Future of world starts in a Classroom" Our Vision is to invent & develop secure and future ready Ed-tech solutions featuring simple functionality.
Exam Proctoring
An all-inclusive remote monitoring solution, including Image verification and computer restrictions.
Automated Proctoring
Live Proctoring
Professional Review
Content
Protection
BraviaProctor.Ai ensures exam integrity by preventing internet navigation, use of mobile

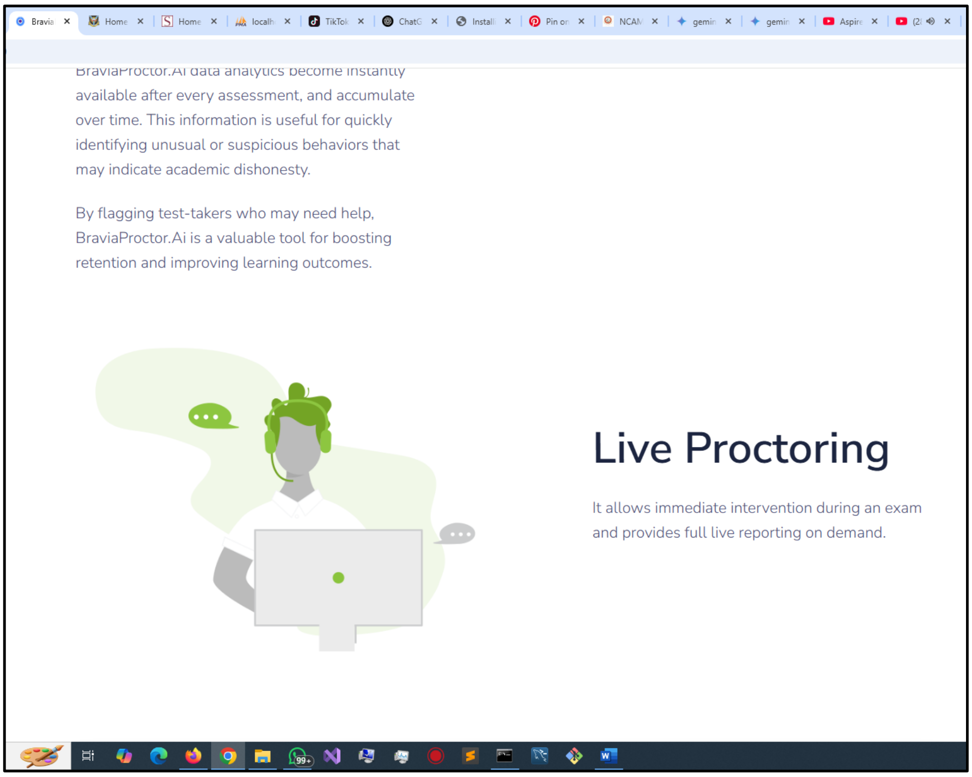


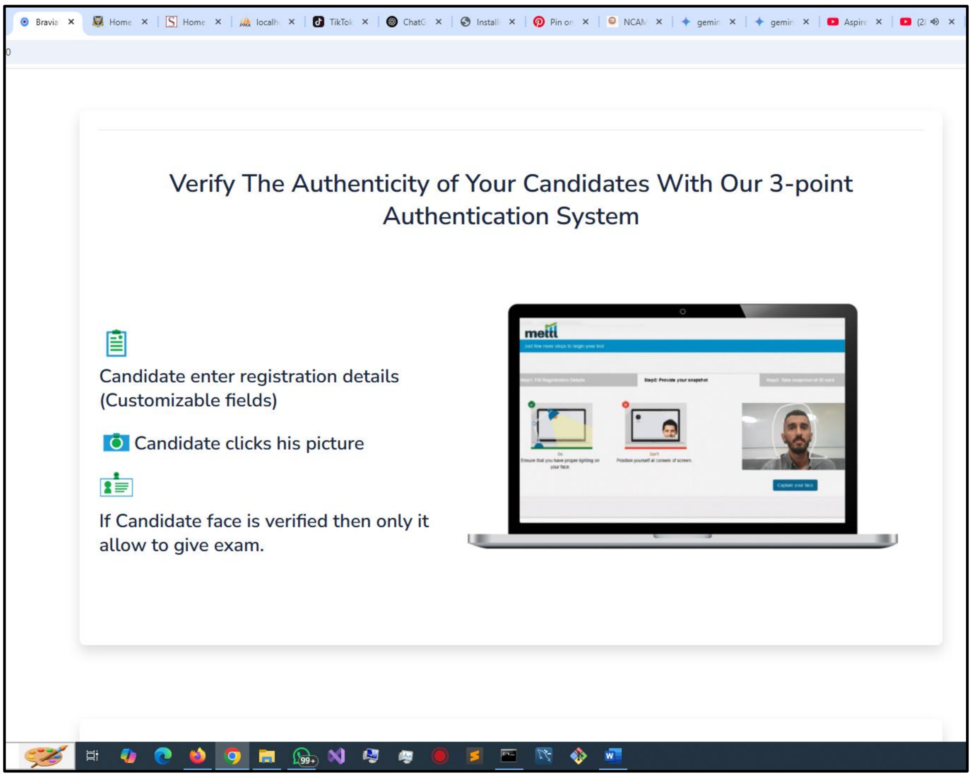
Distribution Prevention
BraviaProctor.Ai prevents test distribution through a host of adjustable settings to restrict the ability to copy and paste, print, download or make screen captures while automatically clearing caches and disabling extensions during the assessment to keep exam content secure.
Image Verification
Identification is captured, analyzed and verified, then only allowed to login.
Advanced facial detection technology
Automated Verification
Live Image Verification

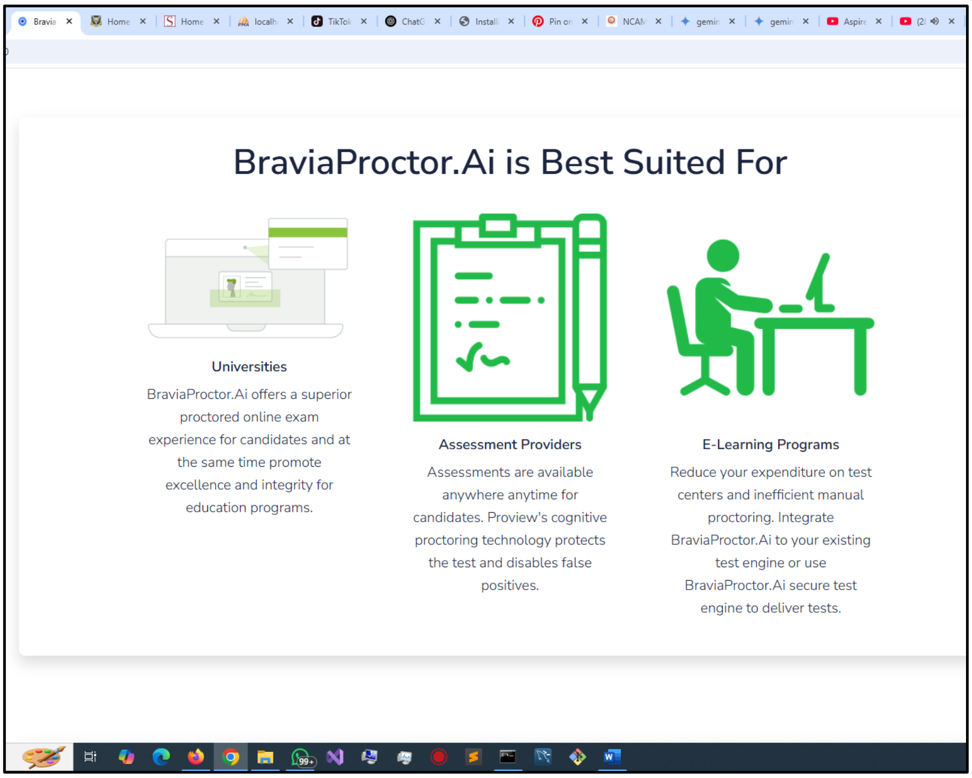


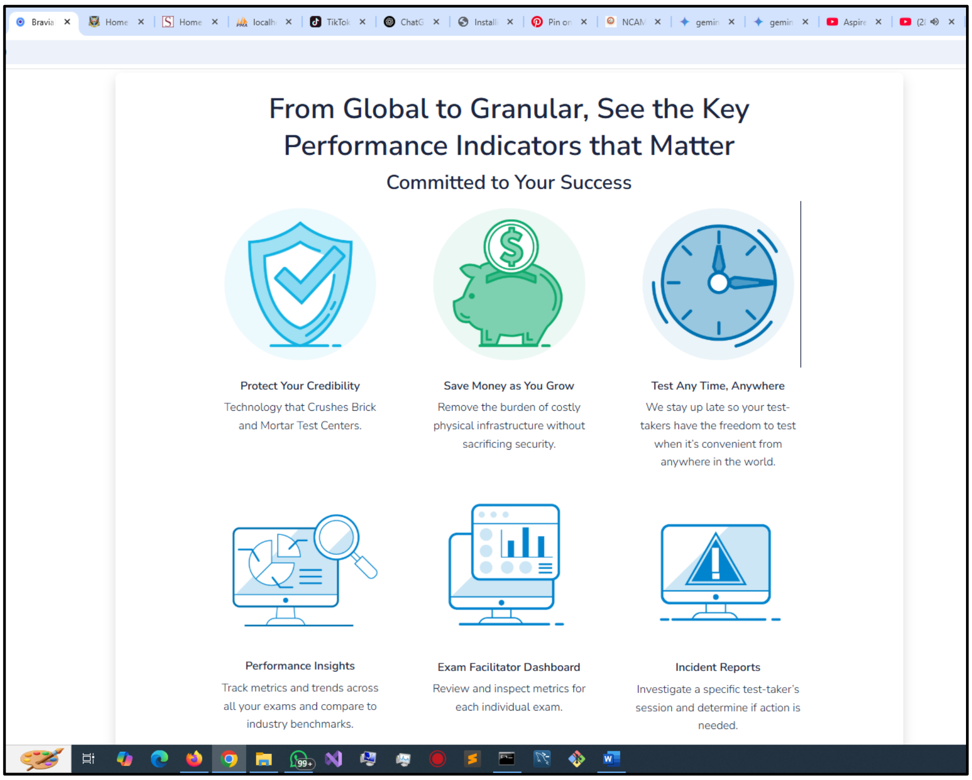
Data Analytics
Instant, objective data is immediately available after every assessment, with parameters set by instructors or administrators.
Administrative Dashboard
Student Monitoring
Aggregate Exam Data
Admin Dashboard
Create global settings and usage reports that give you actionable information by department, school or institution. Filter activity and reporting by course, test-taker, or exam.
Manage exam parameters, computer requirements, account information, faculty controls, and Gradebook settings.

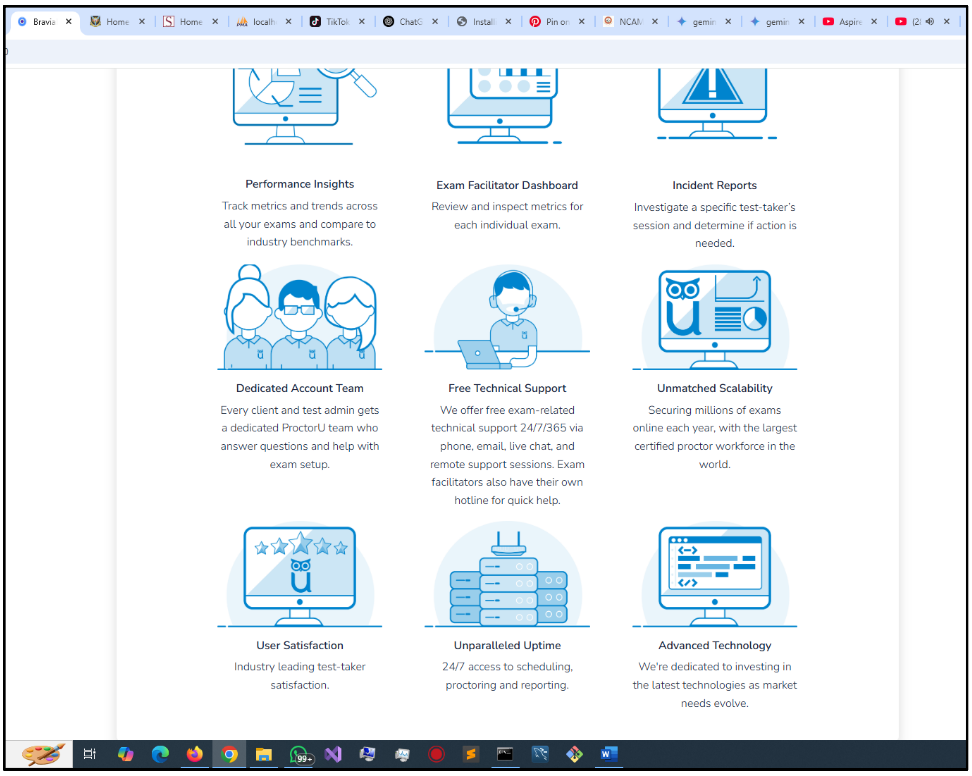

Professional Review
Our highly-trained experts are available to analyze exam proctoring for issues of academic dishonesty. We deliver detailed reporting same day, next day or in 3 days based on your needs.
Actionable Information
BraviaProctor.Ai data analytics become instantly available after every assessment, and accumulate over time. This information is useful for quickly identifying unusual or suspicious behaviors that may indicate academic dishonesty.
By flagging test-takers who may need help, BraviaProctor.Ai is a valuable tool for boosting retention and improving learning outcomes.

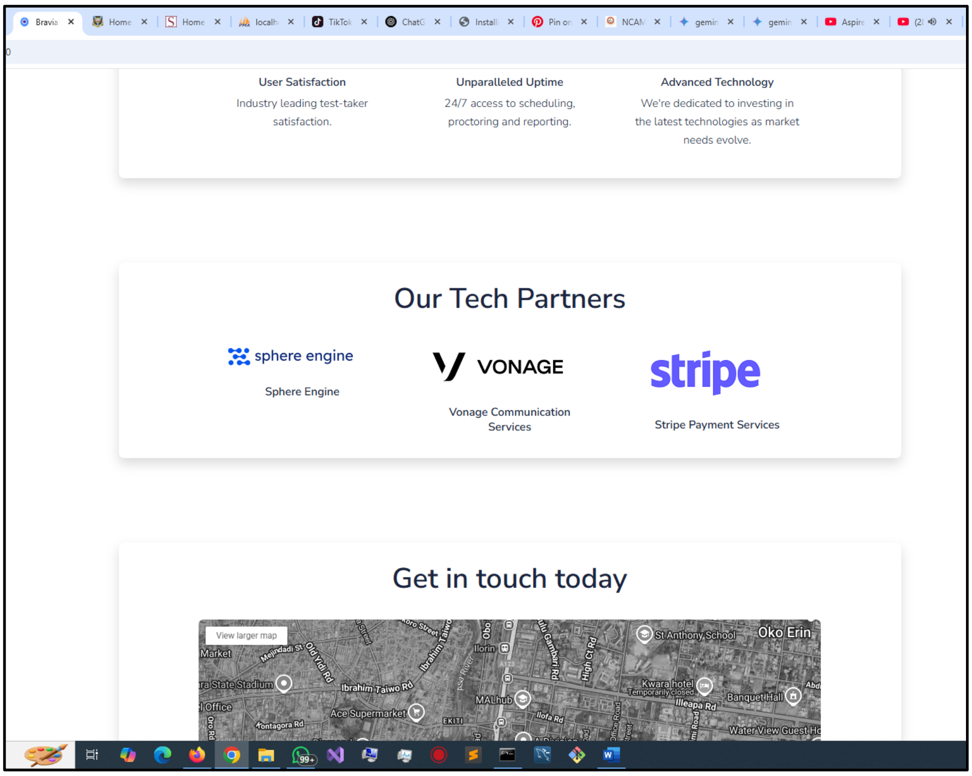

available after every assessment, and accumulate over time. This information is useful for quickly identifying unusual or suspicious behaviors that may indicate academic dishonesty.
By flagging test-takers who may need help, BraviaProctor.Ai is a valuable tool for boosting retention and improving learning outcomes.
Live Proctoring
It allows immediate intervention during an exam and provides full live reporting on demand.

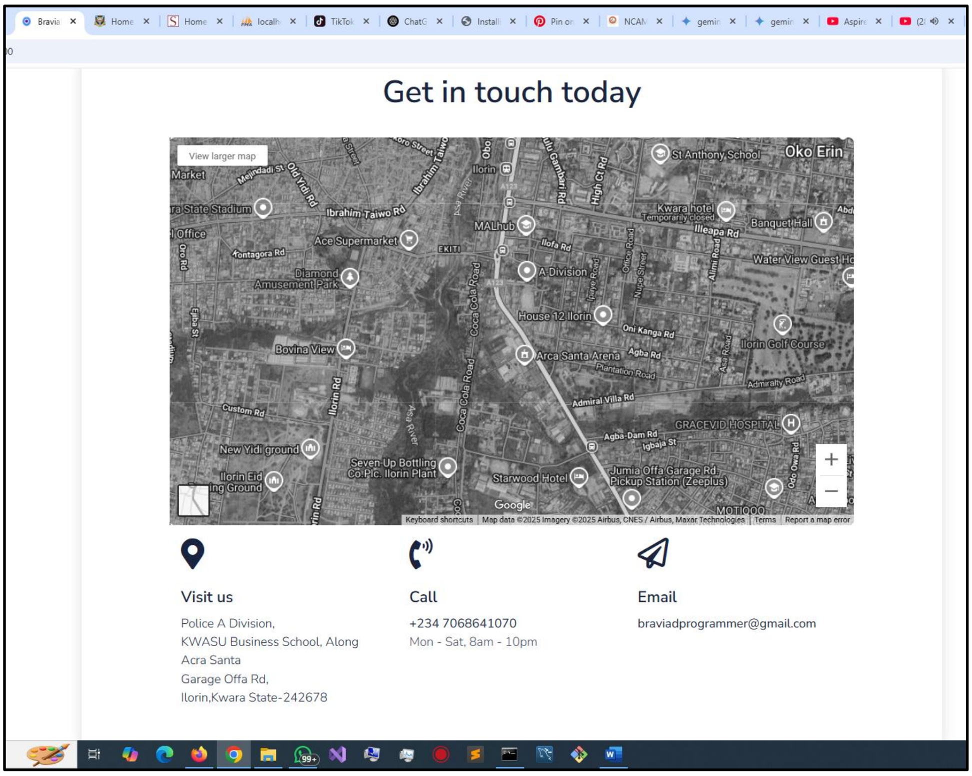

Verify The Authenticity of Your Candidates With Our 3-point Authentication System
Candidate enter registration details (Customizable fields)
Candidate clicks his picture
If Candidate face is verified then only it allow to give exam.

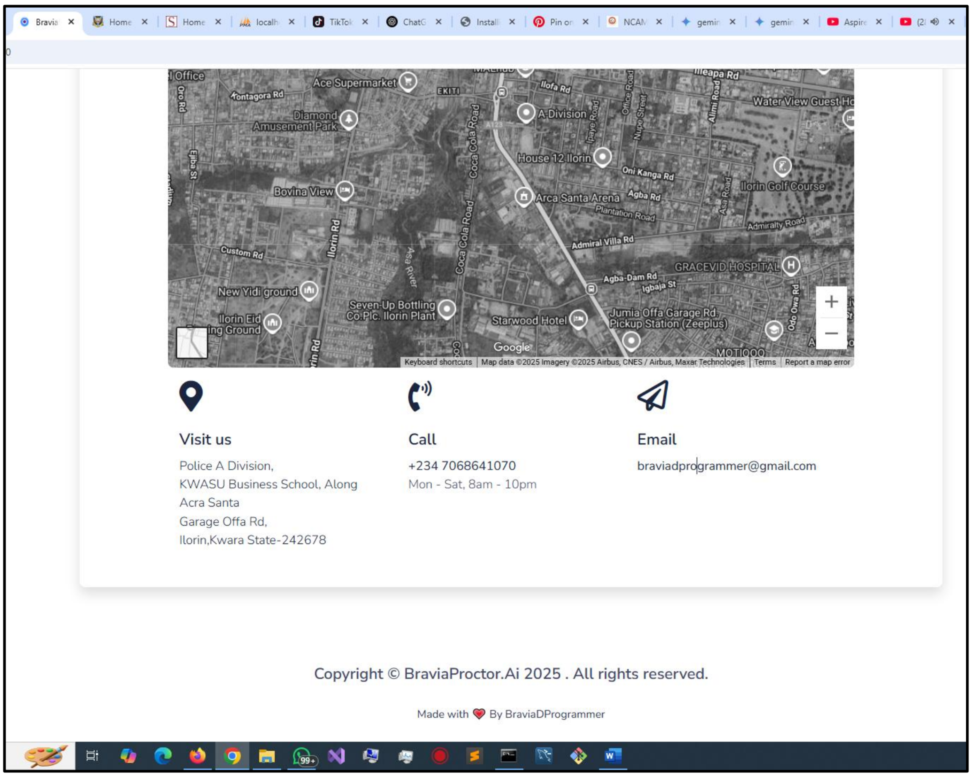

BraviaProctor.Ai is Best Suited For
Universities
BraviaProctor.Ai offers a superior proctored online exam experience for candidates and at the same time promote excellence and integrity for education programs.
Assessment Providers
Assessments are available anywhere anytime for candidates. Proview's cognitive proctoring technology protects the test and disables false positives.
E-Learning Programs
Reduce your expenditure on test centers and inefficient manual proctoring. Integrate BraviaProctor.Ai to your existing test engine or use BraviaProctor.Ai secure test engine to deliver tests.


From Global to Granular, See the Key Performance Indicators that Matter
Committed to Your Success
Protect Your Credibility
Technology that Crushes Brick and Mortar Test Centers.
Save Money as You Grow
Remove the burden of costly physical infrastructure without sacrificing security.
Test Any Time, Anywhere
We stay up late so your test-takers have the freedom to test when it's convenient from anywhere in the world.
Performance Insights
Track metrics and trends across all your exams and compare to industry benchmarks.
Exam Facilitator Dashboard
Review and inspect metrics for each individual exam.
Incident Reports
Investigate a specific test-taker's session and determine if action is needed.

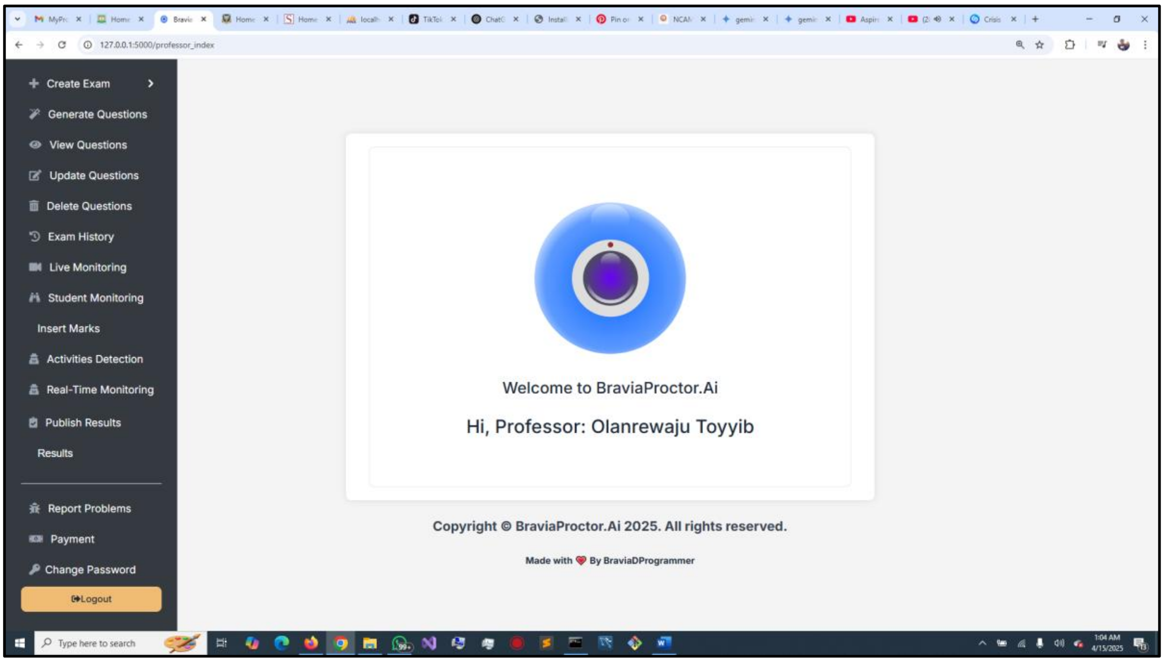

Performance Insights
Track metrics and trends across all your exams and compare to industry benchmarks.
Exam Facilitator Dashboard
Review and inspect metrics for each individual exam.
Incident Reports
Investigate a specific test-taker's session and determine if action is needed.
Dedicated Account Team
Every client and test admin gets a dedicated ProctorU team who answer questions and help with exam setup.
Free Technical Support
We offer free exam-related technical support 24/7/365 via phone, email, live chat, and remote support sessions. Exam facilitators also have their own hotline for quick help.
Unmatched Scalability
Securing millions of exams online each year, with the largest certified proctor workforce in the world.
User Satisfaction
Industry leading test-taker satisfaction.
Unparalleled Uptime
24/7 access to scheduling, proctoring and reporting.
Advanced Technology
We're dedicated to investing in the latest technologies as market needs evolve.


User Satisfaction
Industry leading test-taker satisfaction.
Unparalleled Uptime
24/7 access to scheduling, proctoring and reporting.
Advanced Technology
We're dedicated to investing in the latest technologies as market needs evolve.
Our Tech Partners
sphere engine
Sphere Engine
VONAGE
Vonage Communication Services
stripe
Stripe Payment Services
Get in touch today
View larger map

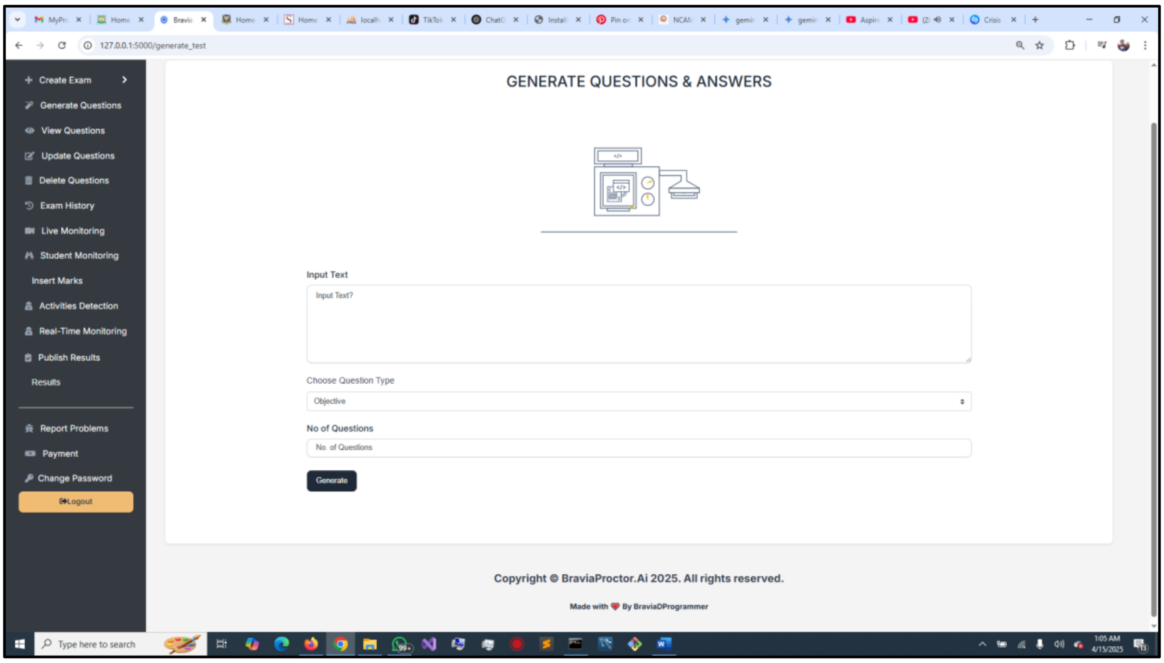

Get in touch today
View larger map
Google
Visit us
Police A Division, KWASU Business School, Along Acra Santa Garage Offa Rd, Ilorin,Kwara State-242678
Call
+234 7068641070
Mon - Sat, 8am - 10pm
Email
braviadprogrammer@gmail.com

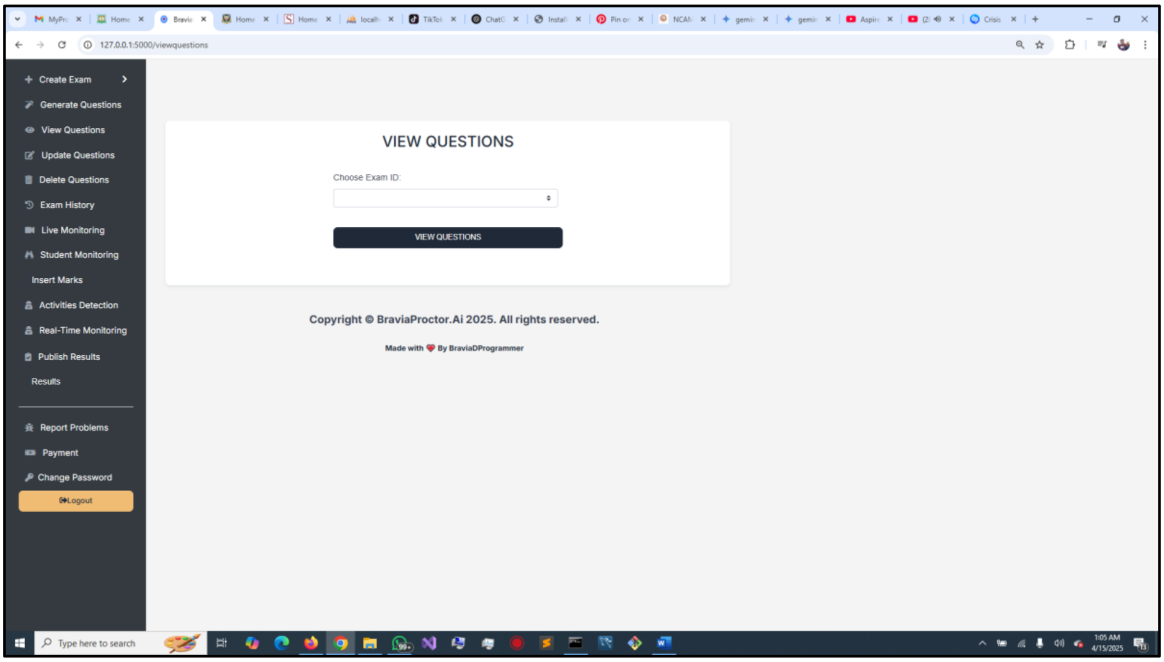


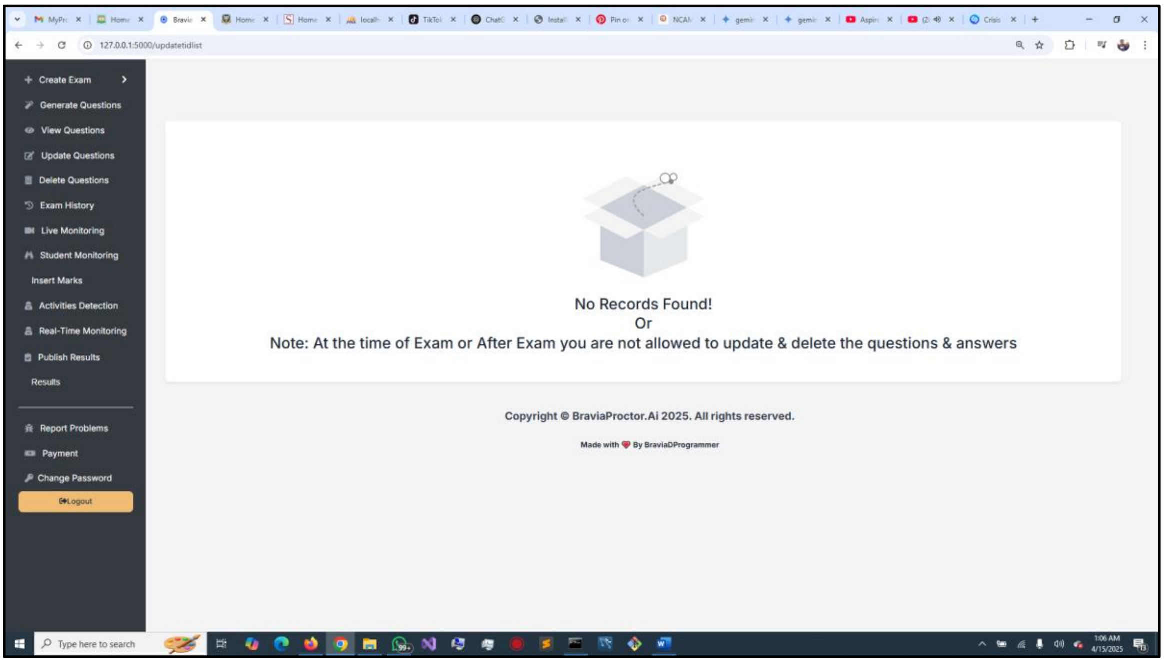
Visit us
Police A Division,
KWASU Business School, Along
Acra Santa
Garage Offa Rd,
Ilorin,Kwara State-242678
Call
+234 7068641070
Mon - Sat, 8am - 10pm
Email
braviadprogrammer@gmail.com
Copyright © BraviaProctor.Ai 2025 . All rights reserved.
Made with By BraviaDProgrammer

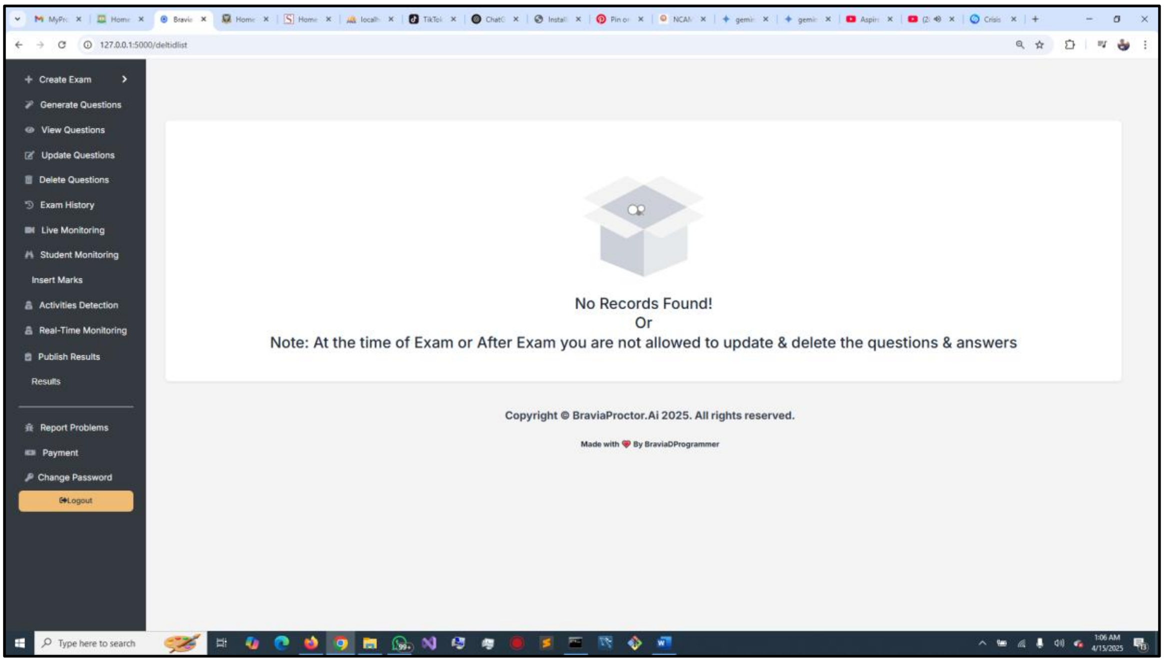


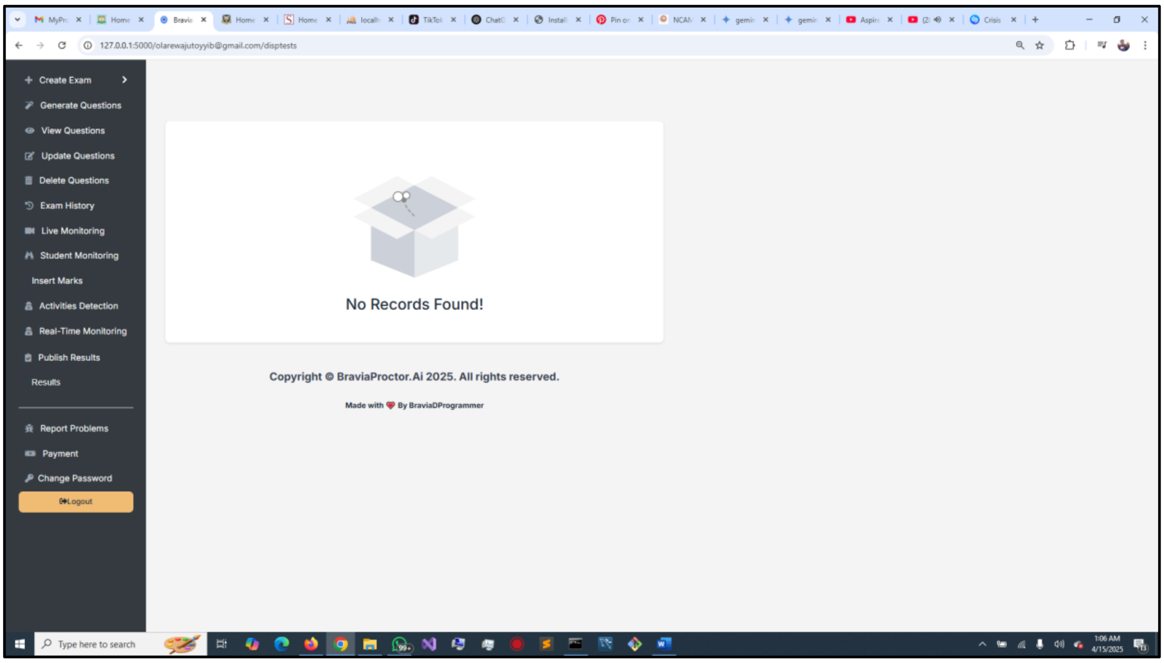
127.0.0.1:5000/professor_index
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
Welcome to BraviaProctor.Ai
Hi, Professor: Olanrewaju Toyyib
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer

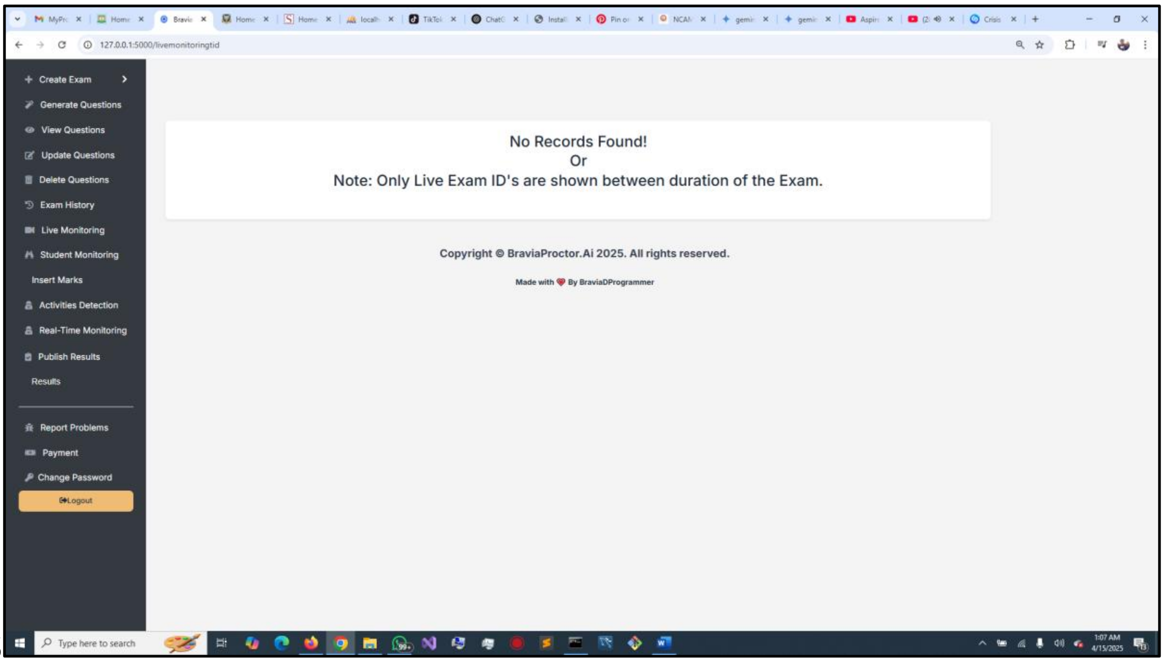

127.0.0.1:5000/generate_test
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
GENERATE QUESTIONS & ANSWERS
Input Text
Input Text?
Choose Question Type
Objective
No of Questions
No. of Questions
Generate
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search
1:05 AM
4/15/2025


127.0.0.1:5000/viewquestions
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
VIEW QUESTIONS
Choose Exam ID:
VIEW QUESTIONS
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search
1:05 AM
4/15/2025

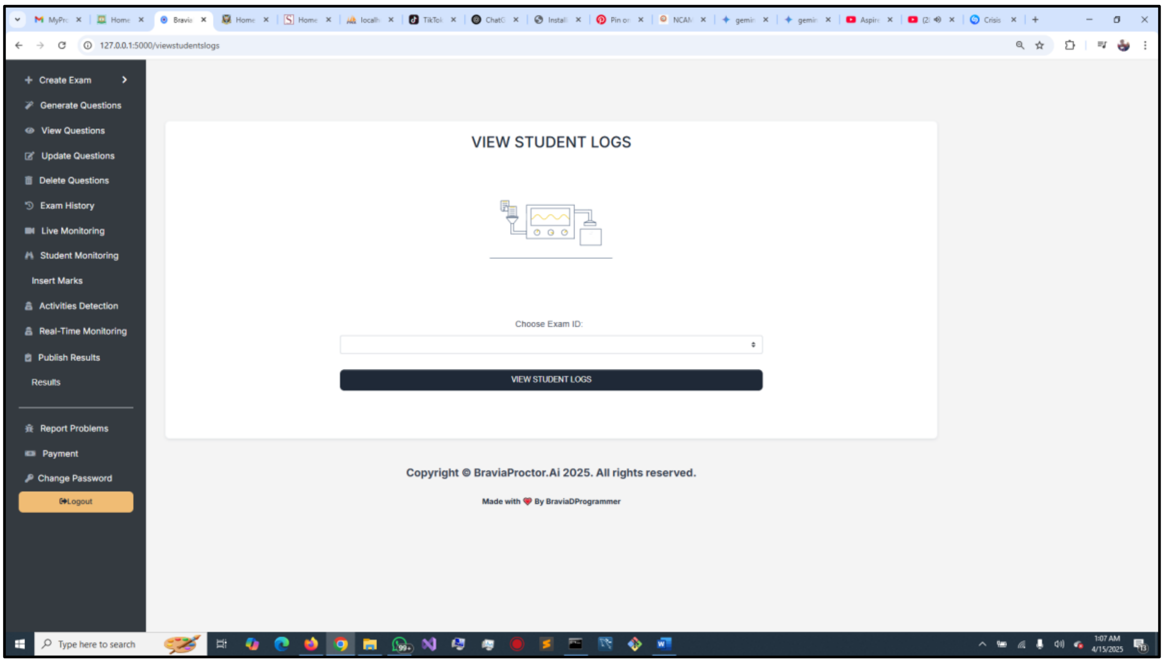

127.0.0.1:5000/updatetidlist
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
No Records Found!
Or
Note: At the time of Exam or After Exam you are not allowed to update & delete the questions & answers
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search

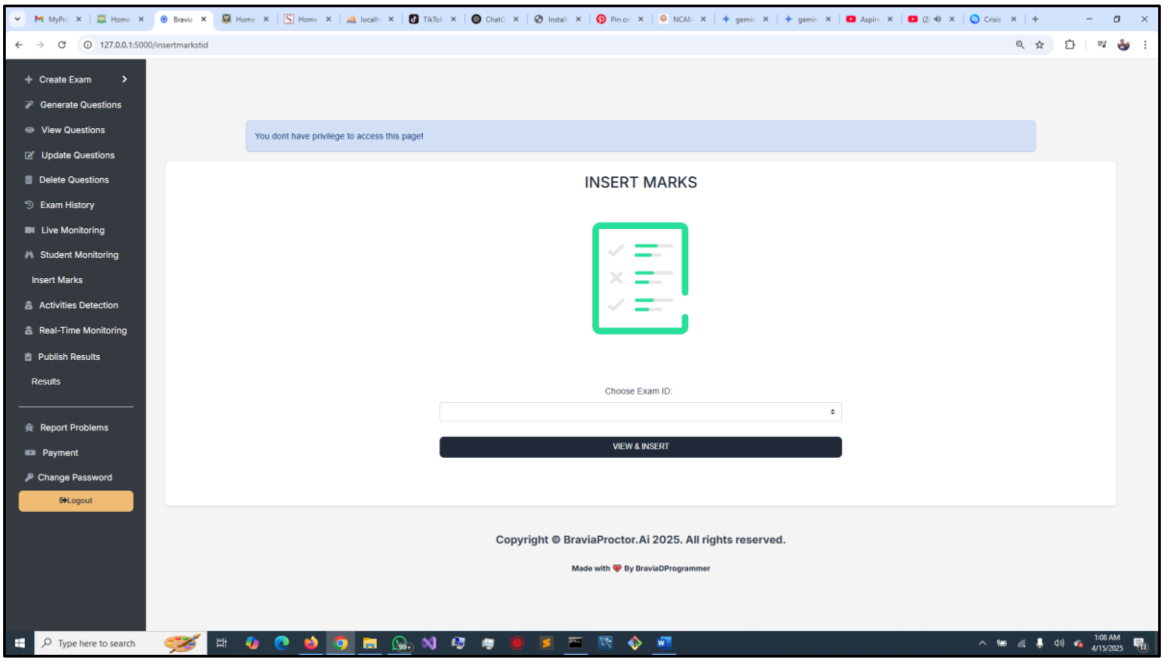

127.0.0.1:5000/deltidlist
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
No Records Found!
Or
Note: At the time of Exam or After Exam you are not allowed to update & delete the questions & answers
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search

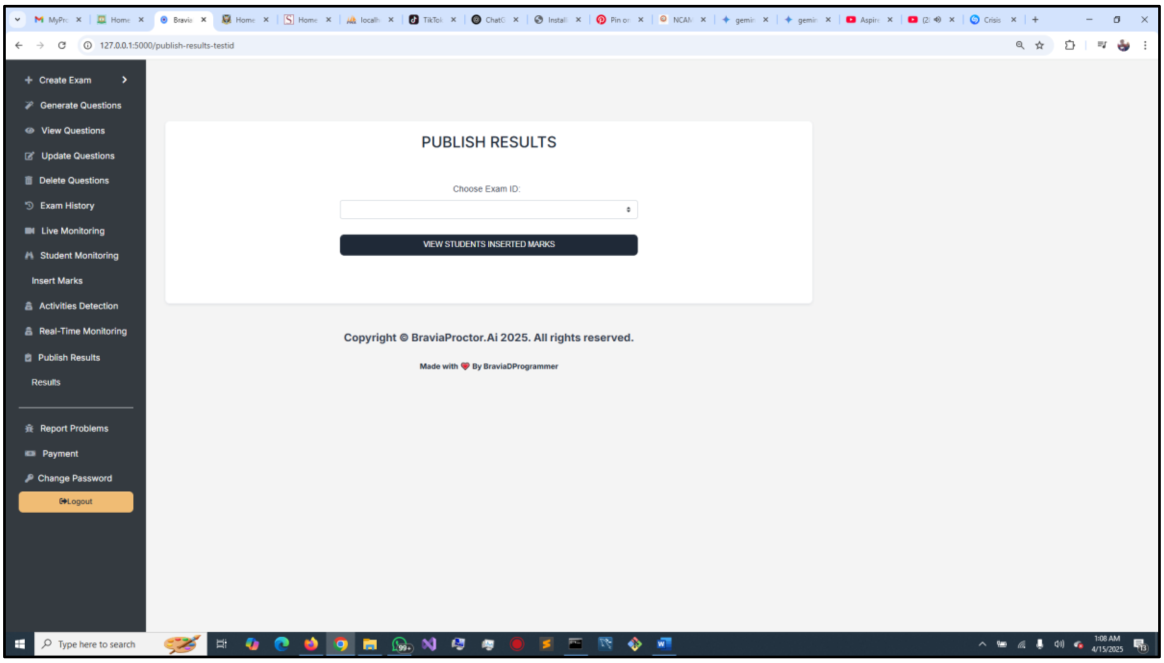

127.0.0.1:5000/olarewajutoyyib@gmail.com/disptests
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
No Records Found!
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search

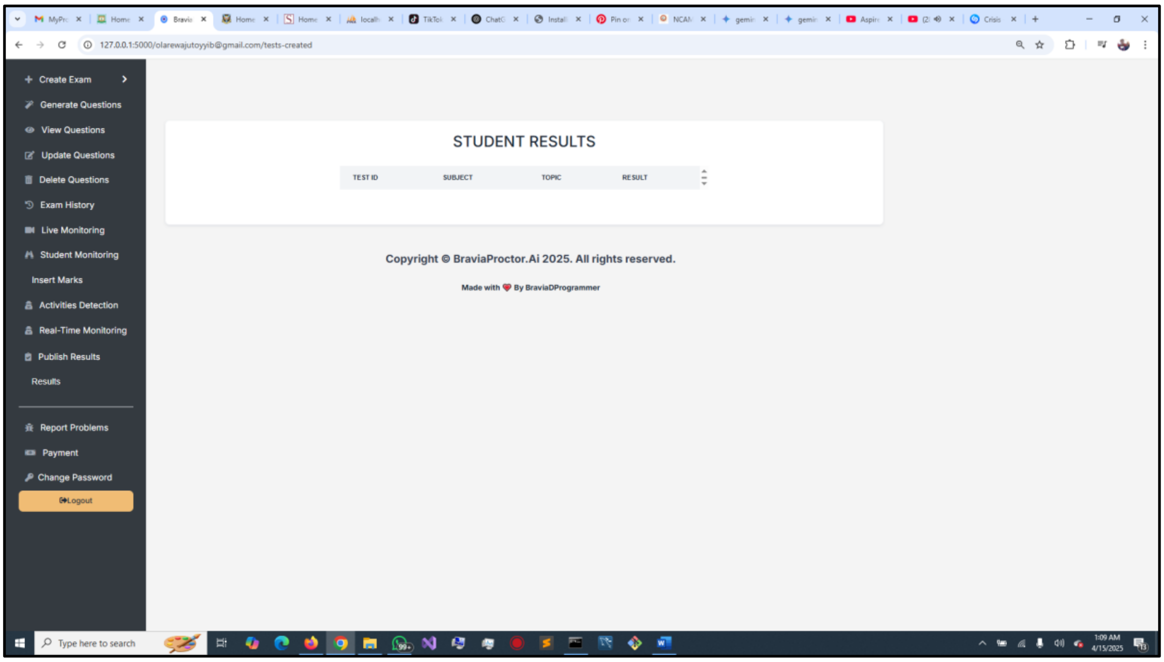

127.0.0.1:5000/livemonitoringtid
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
No Records Found!
Or
Note: Only Live Exam ID's are shown between duration of the Exam.
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer
Type here to search

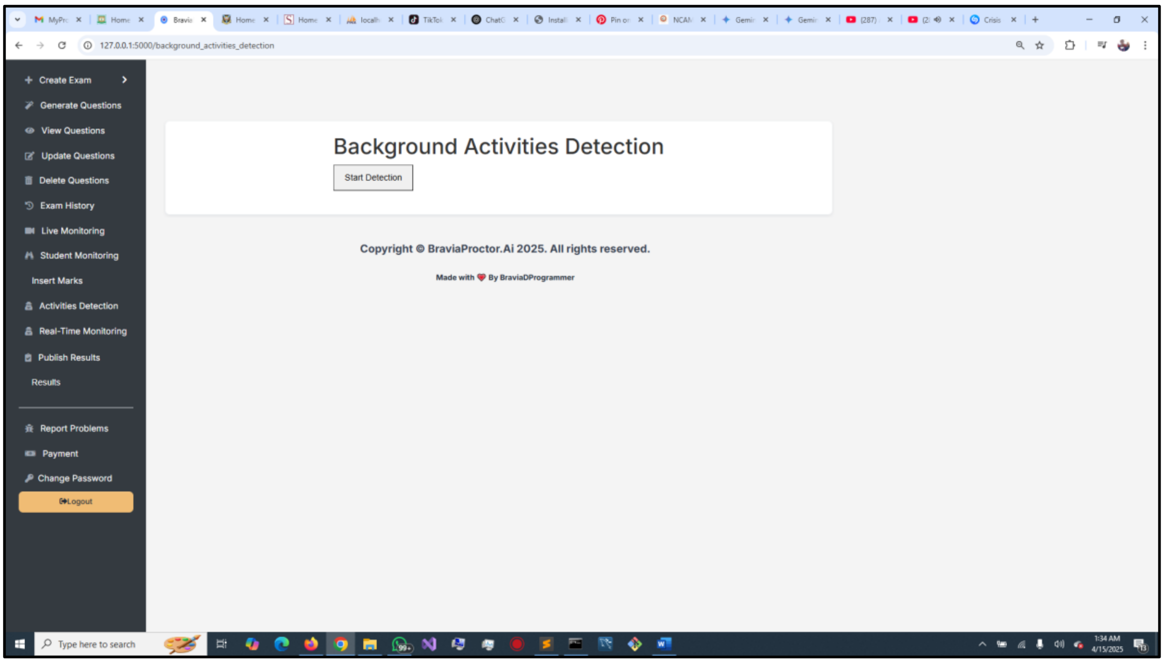
6

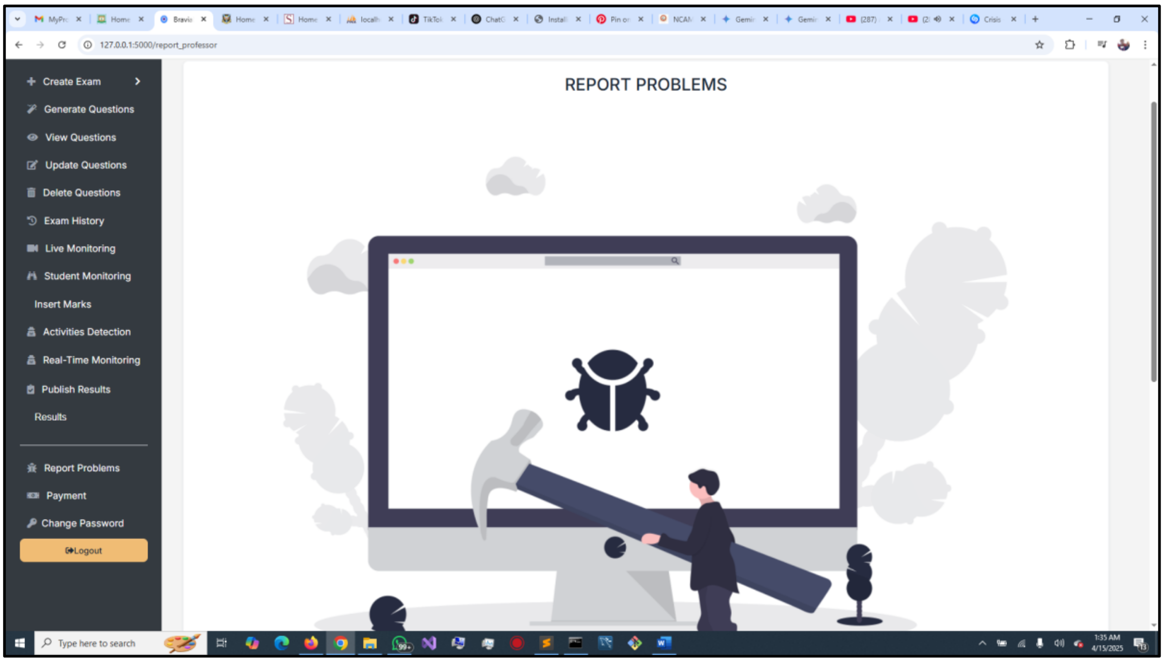

127.0.0.1:5000/viewstudentslogs
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
VIEW STUDENT LOGS
Choose Exam ID:
VIEW STUDENT LOGS
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer

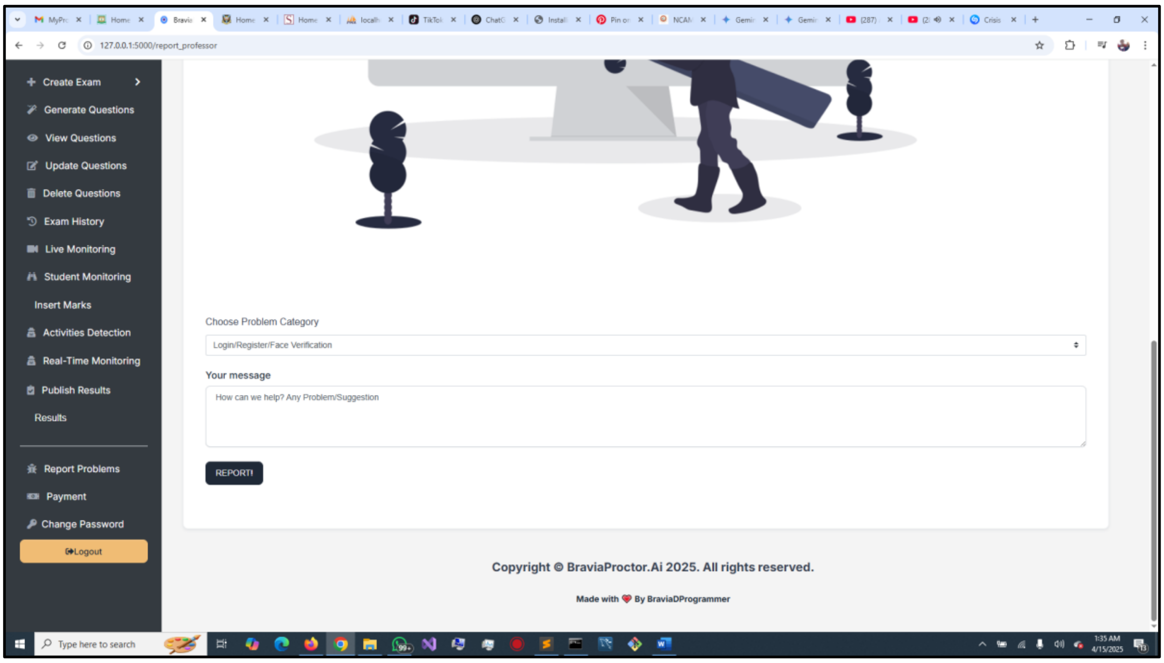

127.0.0.1:5000/insertmarkstid
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
You dont have privilege to access this page!
INSERT MARKS
Choose Exam ID:
VIEW & INSERT
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with By BraviaDProgrammer

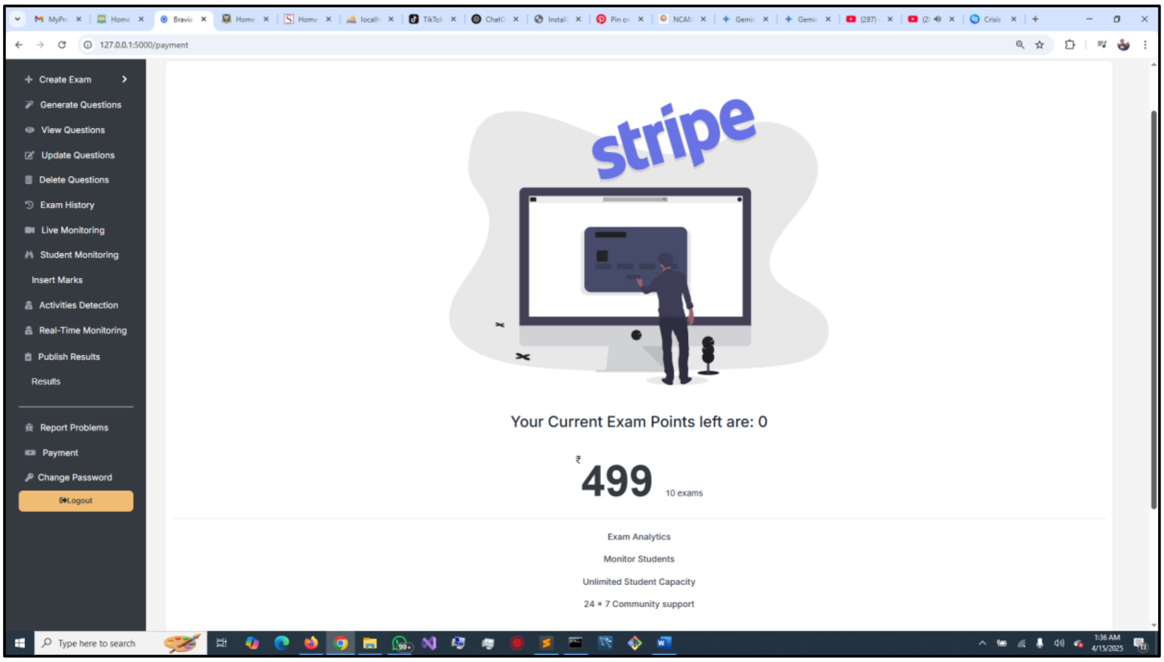

127.0.0.1:5000/publish-results-testid
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
PUBLISH RESULTS
Choose Exam ID:
VIEW STUDENTS INSERTED MARKS
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with ❤ By BraviaDProgrammer
Type here to search

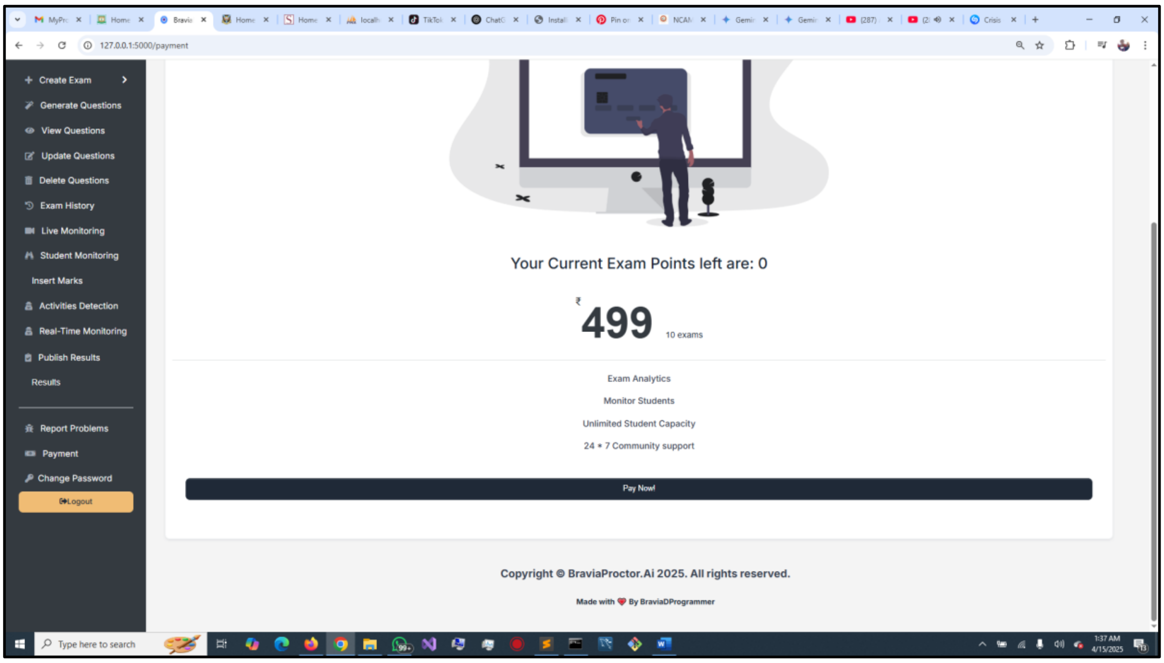

127.0.0.1:5000/olarewajutoyyib@gmail.com/tests-created
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
STUDENT RESULTS
TEST ID
SUBJECT
TOPIC
RESULT
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with ❤ By BraviaDProgrammer
Type here to search

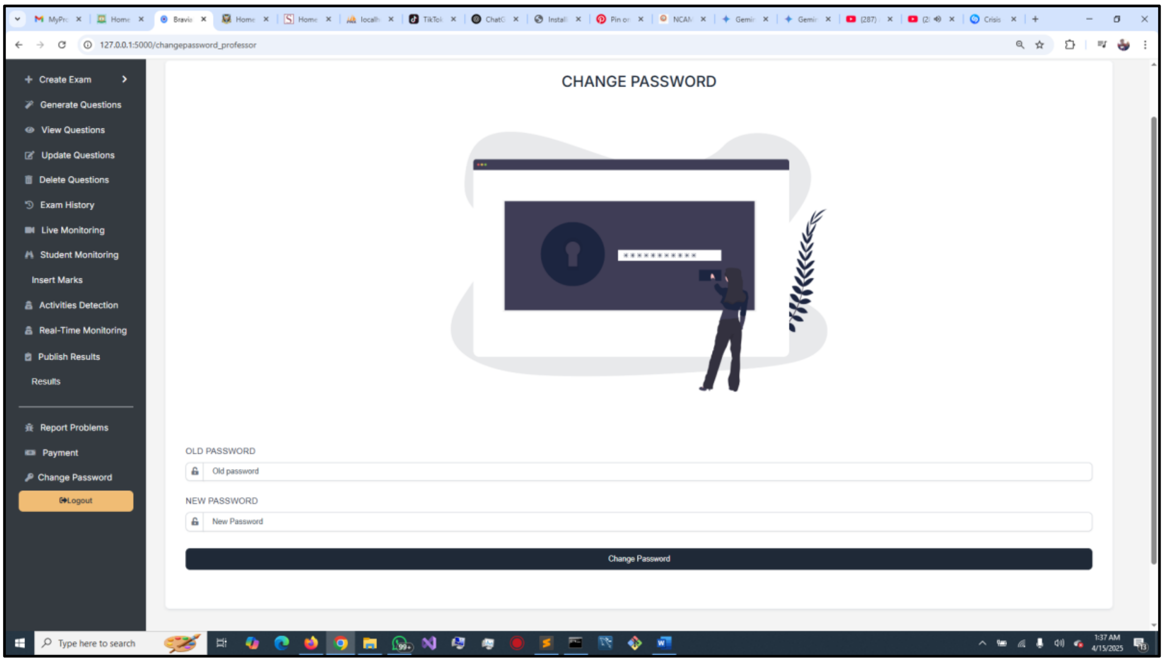

127.0.0.1:5000/background_activities_detection
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
Background Activities Detection
Start Detection
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with ❤ By BraviaDProgrammer

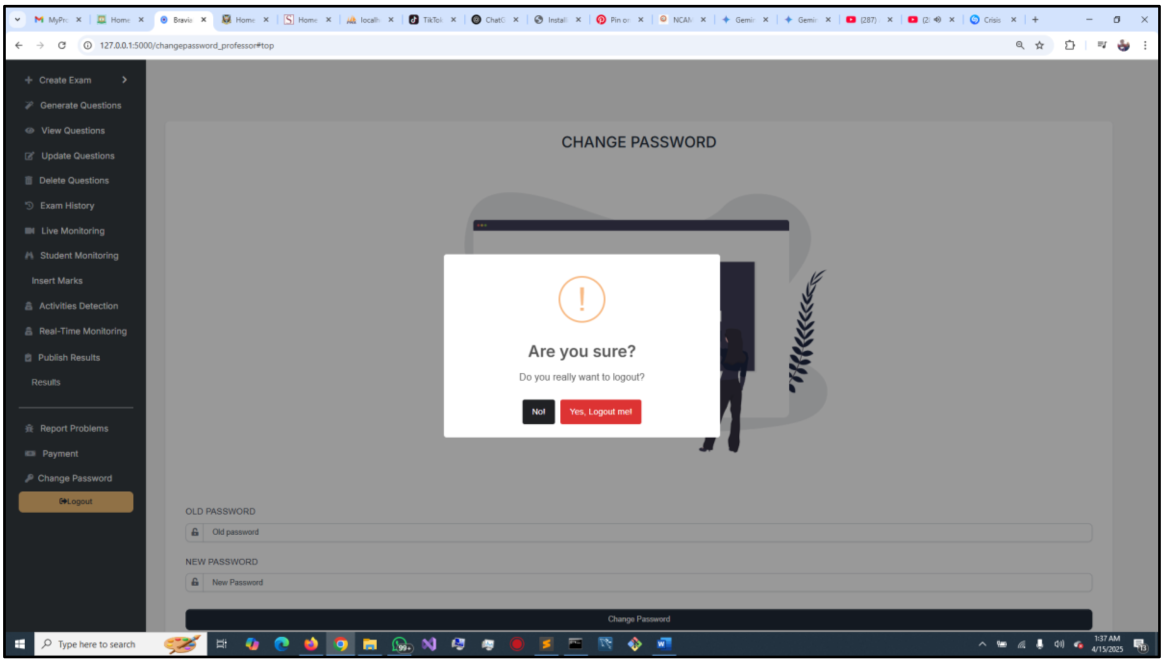

127.0.0.1:5000/report_professor
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
REPORT PROBLEMS

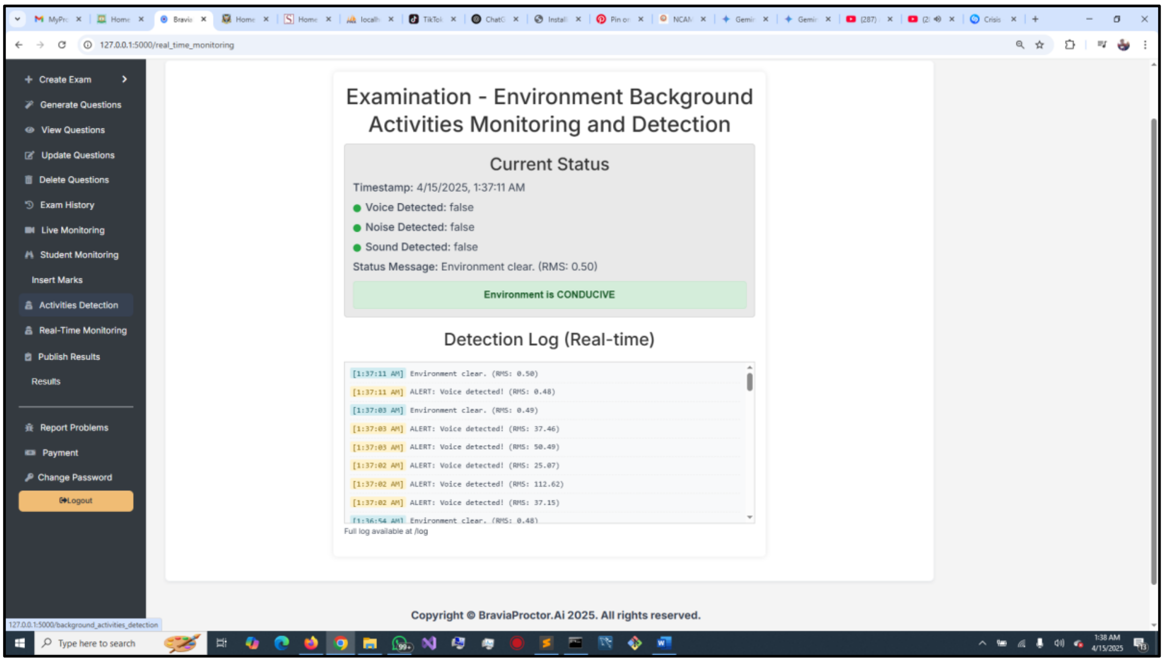

127.0.0.1:5000/report_professor
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
Choose Problem Category
Login/Register/Face Verification
Your message
How can we help? Any Problem/Suggestion
REPORT!
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with ❤ By BraviaDProgrammer
Type here to search

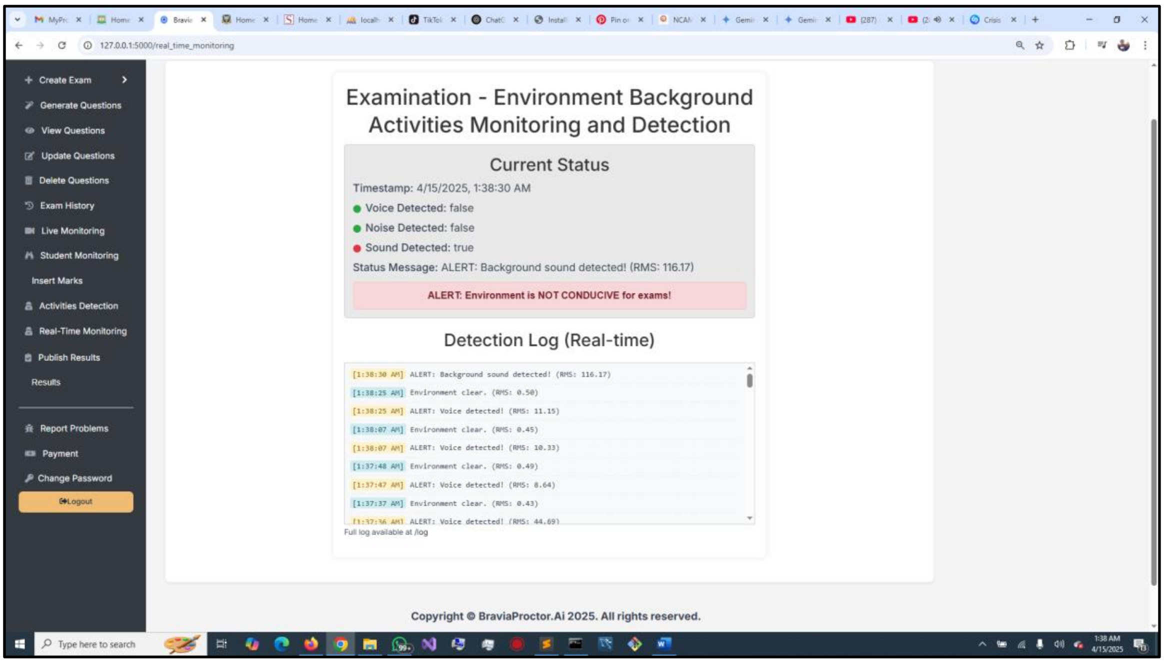

127.0.0.1:5000/payment
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
stripe
Your Current Exam Points left are: 0
₹
499
10 exams
Exam Analytics
Monitor Students
Unlimited Student Capacity
24 * 7 Community support
Type here to search


127.0.0.1:5000/payment
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
Your Current Exam Points left are: 0
₹
499
10 exams
Exam Analytics
Monitor Students
Unlimited Student Capacity
24 * 7 Community support
Pay Now!
Copyright © BraviaProctor.Ai 2025. All rights reserved.
Made with ❤ By BraviaDProgrammer
Type here to search

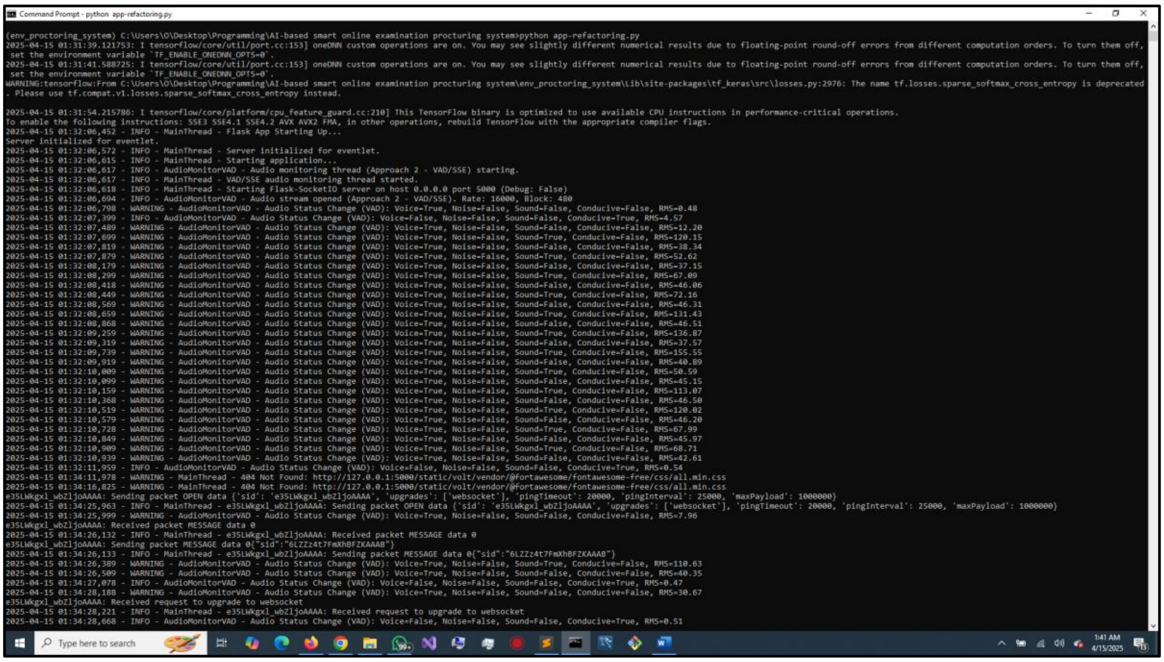

127.0.0.1:5000/changepassword_professor
Create Exam
Generate Questions
View Questions
Update Questions
Delete Questions
Exam History
Live Monitoring
Student Monitoring
Insert Marks
Activities Detection
Real-Time Monitoring
Publish Results
Results
Report Problems
Payment
Change Password
Logout
CHANGE PASSWORD
OLD PASSWORD
Old password
NEW PASSWORD
New Password
Change Password
Type here to search

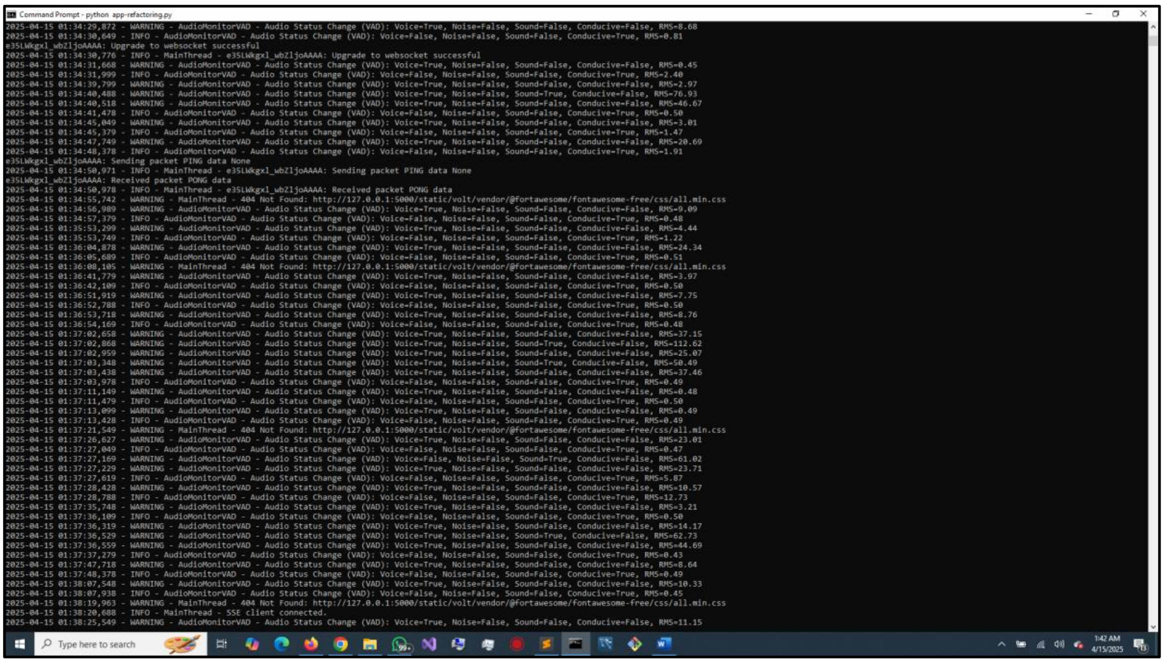

CHANGE PASSWORD
Are you sure?
Do you really want to logout?
No!
Yes, Logout me!
OLD PASSWORD
NEW PASSWORD
Change Password

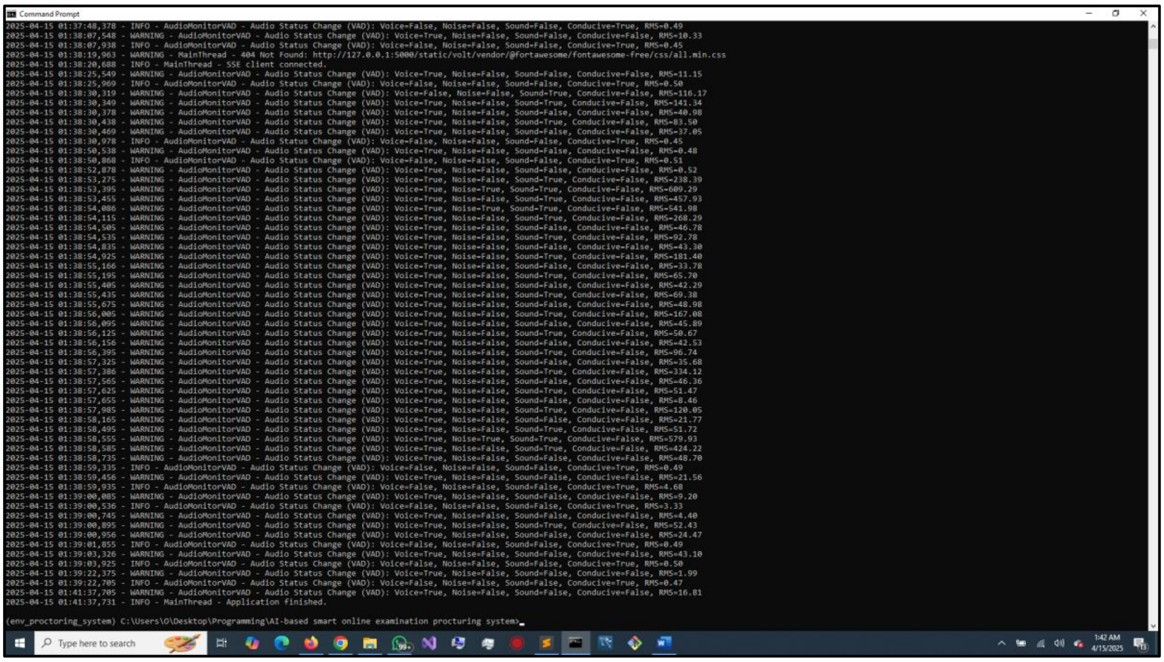

Examination - Environment Background Activities Monitoring and Detection
Current Status
Timestamp: 4/15/2025, 1:37:11 AM
Voice Detected: false
Noise Detected: false
Sound Detected: false
Status Message: Environment clear. (RMS: 0.50)
Environment is CONDUCIVE
Detection Log (Real-time)
Copyright © BraviaProctor.Ai 2025. All rights reserved.

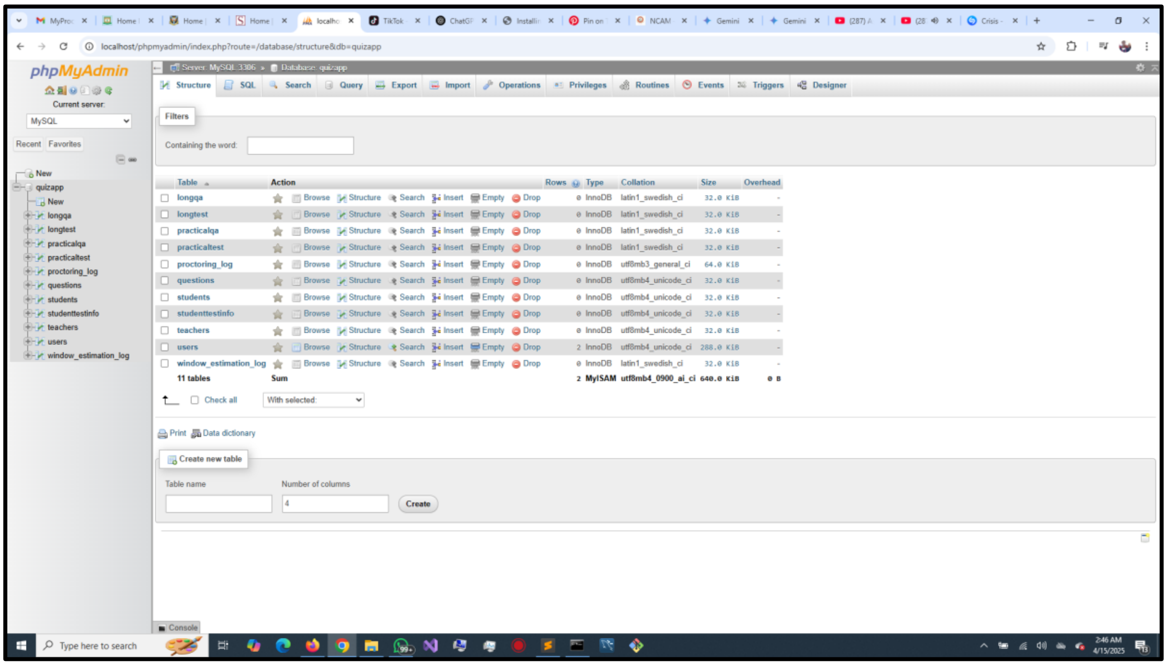


Examination - Environment Background Activities Monitoring and Detection
Current Status
Timestamp: 4/15/2025, 1:38:30 AM
Voice Detected: false
Noise Detected: false
Sound Detected: true
Status Message: ALERT: Background sound detected! (RMS: 116.17)
ALERT: Environment is NOT CONDUCIVE for exams!
Detection Log (Real-time)
Full log available at /log
Copyright © BraviaProctor.Ai 2025. All rights reserved.




Command Prompt - python app-refactoring.py


Command Prompt


phpMyAdmin
Current server:
MySQL
Recent Favorites
New
quizapp
longqa
longtest
practicalqa
practicaltest
proctoring_log
questions
students
studenttestinfo
teachers
users
window_estimation_log
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer
Filters
Containing the word:
Table Action Rows Type Collation Size Overhead
11 tables Sum
Check all
With selected:
Print Data dictionary
Create new table
Table name
Number of columns
Create
Console
Type here to search

Appendix B – System Source Code

Can be gotten from repo on GitHub on this URL link *https://github.com/Braviadprogrammer*