REMOTE E-PROCTORING SYSTEM

A PROJECT REPORT

Submitted by

MOHAN VAMSI A [REGISTER NO: 211416104150]
NITEESH B [REGISTER NO: 211416104169]
SAI ASHWIN D [REGISTER NO:211417104229]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

AUGUST 2021

BONAFIDE CERTIFICATE

Certified that this project report **"REMOTE E-PROCTORING SYSTEM"** is the bonafide work of "MOHAN VAMSI A(211417104150), NITEESH B (211417104169), SAI ASHWIN D (211417104229) " who carried out the project under my supervision.

SIGNATURE

Dr. S. MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mr. K.KAJENDRAN, M.C.A., M.E.,
SUPERVISOR
ASSOCIATE PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voice Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYA RAJESWARI**, **Dr.C.SAKTHIKUMAR, M.E.,Ph.D.,**and **Tmt. SARANYASREE SAKTHI KUMAR B.E., M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of the CSE Department, **Dr. S.MURUGAVALLI ,M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mr. K. KAJENDRAN, M.C.A.,M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

MOHAN VAMSI .A
NITEESH .B
SAI ASHWIN .D

ABSTRACT

The project aims to provide a safe and user-friendly environment to take exams online to various locations for those who don't have access to campus. The project employs various machine learning models to reduce the burden on the admin to detect malpractice by the students on a large scale. The model verifies and authenticates the candidates taking the exam and continuously checks for any malpractices. This is done by analyzing video and audio that the model processes to verify the integrity of the student. Continuous processing of the inputs allows the system to check the integrity of the candidate so that academic integrity in e-learning is maintained. The system is affordable and convenient to use from the test taker's perspective since it only requires having inexpensive web cameras and a microphone. The system includes five basic components that continuously estimate the key behavior cues: user verification, audio processing, gaze detection, number of person detection, object and phone detection. By combining the continuous estimation components, and applying a temporal sliding window, we design higher-level features to classify whether the test taker is cheating at any moment during the exam.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

S.No	ABBREVIATION	EXPANSION
1	**HCI**	Human Computer Interaction
2	**AAM**	Active Appearance Model
3	**AAAM**	Adaptive Active Appearance Model
4	**SSE**	Streaming SIMD Extension
5	**STL**	Standard Template Library
6	**CUDA**	Compute Unified Device Architecture
7	**NLP**	Natural Language Processing
8	**NLTK**	Natural Language ToolKit
9	**UI**	User Interface
10	**AWS**	Amazon Web Service
11	**CPU**	Central Processing Unit
12	**CRM**	Customer Relationship Management
13	**OES**	Online Examination System
14	**Amazon S3**	Simple Storage Service
15	**VLE**	Virtual Learning Environment
16	**OpenCV**	Open Source Computer Vision Library
17	**YOLOv3**	You Only Look Once Version 3
18	**GPU**	Graphics Processing Unit

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

The use of e-learning or other remote education continues to increase due to its ability to reach people who don't have access to campus. Exams are important components of educational programs as well as on an online learning program. Conducting exams gives the institution a very brief perspective about the delivery of knowledge to their students irrespective of the medium of interaction, be it offline or online mode. In an exam, a proctoring method is employed to detect and reduce the malpractices involved. It turns out to be very important so that the examiner can ensure that the students have learned the material given and try to grasp the knowledge delivered to them to their potential. Various methods had been proposed to provide an efficient, effective and comfortable online exam proctoring. A visual verification for the whole exam session is needed in an online exam, therefore a face verification is needed. This is to ensure that the examinee is a legit candidate who has enrolled himself/herself for the process of examination with the organization. The visual verification part will have to deal with detecting the person who has enrolled for taking the examination. For implementing this part we utilize image capturing techniques to capture the image of the candidate. Secondly, the system employs fragments of alternative algorithms which work on the audio which

is being captured using the microphone of the connected device(s). The audio feed will serve as an input for voice processing algorithms which constantly monitor the examination environment. This constant monitoring allows us to use the input data to process accordingly with the help of various algorithms to detect and notify if any kind of verbal malpractice methods are being employed. The system uses specific input references to cross check within its permissible limits to classify any act as indictable. Moreover, there are some other methods which are employed within the system framework that can detect any anomalies in the examination environment. The visual verification module constantly looks out for multiple persons' presence and/or electronic gadgets within the candidates' vicinity. The system is also designed to notify the examiner about the working nodes in the environment which may or may not be a part of a suspicious act. Unlike generic systems which require the presence of an examiner for the whole process, this system employs automated algorithms which process input information and produce a detailed analytical output which can be inspected even after the commencement of the examination. This allows the organisation to be in full control of the entire operation with the help of the automated processing taking place within modules of the system to run a smooth, effective and efficient examination process in which the candidates can be examined thoroughly.

CHAPTER 2
LITERATURE SURVEY

2. LITERATURE SURVEY

TITLE: Automated Online Exam Proctoring[1]

DESCRIPTION: Massive open online courses (MOOCs) and other forms of remote education continue to increase in popularity and reach. The ability to efficiently proctor remote online examinations is an important limiting factor to the scalability of this next stage in education. Presently, human proctoring is the most common approach of evaluation, by either requiring the test taker to visit an examination center, or by monitoring them visually and acoustically during exams via a webcam. However, such methods are labor-intensive and costly. In this paper, we present a multimedia analytic system that performs automatic online exam proctoring. The system hardware includes one webcam, one wearcam, and a microphone, for the purpose of monitoring the visual and acoustic environment of the testing location. The system includes six basic components that continuously estimate the key behavior cues: user verification, text detection, voice detection, active window detection, gaze estimation and phone detection. By combining the continuous estimation components, and applying a temporal sliding window, we design higher level features to classify whether the test taker is cheating at any moment during the exam. To evaluate our proposed system, we collect multimedia (audio and visual) data from 24 subjects performing various types of cheating while taking online exams. Extensive experimental results demonstrate the accuracy, robustness, and efficiency of our online exam proctoring system.

TITLE: A Design of Continuous User Verification for Online Exam Proctoring on M-Learning[2]

DESCRIPTION: The use of m-learning or other remote education continues to increase due to its ability to reach people who don't have access to campus. Exams are important components of educational programs as well as on an online learning program. In an exam, a proctoring method to detect and reduce the cheating possibility is very important to ensure that the students have learned the material given. Various methods had been proposed to provide an efficient, comfortable online exam proctoring. Start with implementing an exam design with hard constraints in a no proctoring exam, a remote proctoring using a webcam, a machine based proctoring and finally research on automated online proctoring. A visual verification for the whole exam session is needed in an online exam, Therefore a face verification is needed. A remaining problem in the face recognition area is the system robustness for pose and lighting variations. In this paper, we proposed a method to enhance the robustness for pose and lighting variations by doing an incremental training process using the training data set obtained from m-learning online lecture sessions. As a result, the design of the proposed method is presented in this paper.

TITLE: An Intelligent System for Online Exam Monitoring[3]

DESCRIPTION: For the past few years, e-learning has become popular across countries because of its flexibility, availability and user friendliness. As far as

online examinations are concerned; the major challenge faced by the research community is the proctoring techniques used. In this paper, we present a method to avoid the physical presence of a proctor throughout the exam by creating a comprehensive multi modal system. We have used hardware such as web-cams to capture audio and video along with active window capture. This combination forms the inputs to an intelligent rule based inference system which has the capability to decide whether any malpractices have happened. Examinee's face is detected and is used to extract feature points thereby estimating a head pose. Misconduct is detected based on yaw angle variations, audio presence and active window capture. Our system has been tested in an e-learning scenario and we were able to make exam monitoring easy. Experiment results proved that our system performed better than the existing systems.

TITLE: An Autonomous Articulating Desktop Robot for Proctoring Remote Online Examinations[4]

DESCRIPTION: In this paper we describe a new low-cost, autonomous desktop robot for proctoring examinations in online/distance learning courses. The robot is attached to the student's computer via a USB port and monitors the examination environment using a webcam that articulates in both altitude and azimuth together with an array of acoustic sensors that provides audio directionality. The examination may be monitored in real time by a live proctor via the Internet or the data may be recorded for future review. Authentication of the identity of the test taker is accomplished using the webcam and simple, reliable ear recognition techniques. This eliminates the need for expensive digital fingerprint hardware.

TITLE:A Multi-Gesture Interaction System Using a 3-D Iris Disk Model for Gaze Estimation and an Active Appearance Model for 3-D Hand Pointing[5]

DESCRIPTION: In this paper, we present a vision-based human–computer interaction system, which integrates control components using multiple gestures, including eye gaze, head pose, hand pointing, and mouth motions. To track head, eye, and mouth movements, we present a two-camera system that detects the face from a fixed, wide-angle camera, estimates a rough location for the eye region using an eye detector based on topographic features, and directs another active pan-tilt-zoom camera to focus in on this eye region. We also propose a novel eye gaze estimation approach for point-of-regard (POR) tracking on a viewing screen. To allow for greater head pose freedom, we developed a new calibration approach to find the 3-D eyeball location, eyeball radius, and fovea position. Moreover, in order to get the optical axis, we create a 3-D iris disk by mapping both the iris center and iris contour points to the eyeball sphere. We then rotate the fovea accordingly and compute the final, visual axis gaze direction. This part of the system permits natural, non-intrusive, pose-invariant POR estimation from a distance without resorting to infrared or complex hardware setups. We also propose and integrate a two-camera hand pointing estimation algorithm for hand gesture tracking in 3-D from a distance. The algorithms of gaze pointing and hand finger pointing are evaluated individually, and the feasibility of the entire system is validated through two interactive information visualization applications.

TITLE: E-cheating : Incidences and Trends Among College Students[6]

DESCRIPTION: There has been substantial research in the area of student cheating in traditional face-to-face classes but much less on cheating using technology often termed “e-cheating.” With the increased offering of online courses to provide students with greater flexibility, concerns regarding academic integrity arise. How many opportunities exist for students to cheat when taking an online course? The authors have utilized a biannual survey to gather information on student cheating in both traditional and online courses. This paper summarizes the results of the authors’ surveys for the past five years. Findings show that the most common type of cheating activity reported by students is the downloading of Internet papers and representing them as the student’s own work. In addition, students report that they believe that other students cheat more on homework problems as compared to exams, term papers, and Internet projects. The most troubling finding for educators, however, is that e-cheating appears to be on the rise.

TITLE: Detection and Controlling of Suspicious Behaviour in the Examination Hall[7]

DESCRIPTION: Within the examination hall, in general the eye gaze and the orientation of the head of the examinee provide us clues to the suspicious behaviors. Nowadays, most of the examinations are held within smart classrooms so that cheating activities can be monitored centrally and remotely. But most of the suspicious behaviors' monitoring procedures are manual. In this study, we proposed a smart system which can autonomously detect and track an examinee's eye gaze

and head orientation to robustly detect their cheating activities. With the help of the webcam installed within each of the smart devices placed on each of the examination desks, initially, real time video is captured during the examination period. Then the video is analyzed to study the examinee's psychology regarding the illegal approaches to build a knowledgebase of our system. Recently, we conducted experiments using our smart system within the examination hall under a controlled environment to validate its effectiveness.

TITLE: E-Invigilator: A biometric-based supervision system for eAssessments[8]

DESCRIPTION: The creation of Virtual Learning Environments (VLEs) have revolutionized the online delivery of learning materials, from traditional lectures slides through to podcasts, blogs and wikis. However, such advances in how we assess such learning have not evolved – with physical attendance at proctored exams still a necessity for formal assessments. This paper presents a novel model to enable remote and electronic invigilation of students during formal assessment. The approach utilizes transparent authentication to provide for a non-intrusive and continuous verification of the candidates identity throughout the examination time frame. A prototype is developed and a technology evaluation of the platform demonstrates the feasibility of the approach.

TITLE: Enhanced Security for Online Exams Using Group Cryptography[9]

DESCRIPTION: While development of the Internet has contributed to the spread of online education, online exams have not been widely adopted. An online exam is

defined here as one that takes place over the insecure Internet, and where no proctor is in the same location as the examinees. This paper proposes an enhanced secure online exam management environment mediated by group cryptography using remote monitoring and control of ports and input. The target domain of this paper is that of online exams for math or English contests in middle or high school, as well as exams in online university courses with students in remote locations.

TITLE: Estimation of behavioral user state based on eye gaze and head pose-application in an e-learning environment[10]

DESCRIPTION: Most e-learning environments which utilize user feedback or profiles, collect such information based on questionnaires, resulting very often in incomplete answers, and sometimes deliberate misleading input. In this work, we present a mechanism which compiles feedback related to the behavioral state of the user (e.g. level of interest) in the context of reading an electronic document; this is achieved using a non-intrusive scheme, which uses a simple web camera to detect and track the head, eye and hand movements and provides an estimation of the level of interest and engagement with the use of a neuro-fuzzy network initialized from evidence from the idea of Theory of Mind and trained from expert annotated data. The user does not need to interact with the proposed system, and can act as if she was not monitored at all. The proposed scheme is tested in an e-learning environment, in order to adapt the presentation of the content to the user profile and current behavioral state. Experiments show that the proposed system detects reading- and attention-related user states very effectively, in a testbed where children's reading performance is tracked.

TITLE: Eye Gaze Tracking With a Web Camera in a Desktop Environment[11]

DESCRIPTION: This paper addresses the eye gaze tracking problem using a low cost and more convenient web camera in a desktop environment, as opposed to gaze tracking techniques requiring specific hardware, e.g., infrared high-resolution camera and infrared light sources, as well as a cumbersome calibration process. In the proposed method, we first track the human face in a real-time video sequence to extract the eye regions. Then, we combine intensity energy and edge strength to obtain the iris center and utilize the piecewise eye corner detector to detect the eye corner. We adopt a sinusoidal head model to simulate the 3-D head shape, and propose adaptive weighted facial features embedded in the pose from the orthography and scaling with iterations algorithm, whereby the head pose can be estimated. Finally, the eye gaze tracking is accomplished by integration of the eye vector and the head movement information. Experiments are performed to estimate the eye movement and head pose on the BioID dataset and pose dataset, respectively. In addition, experiments for gaze tracking are performed in real-time video sequences under a desktop environment. The proposed method is not sensitive to the light conditions. Experimental results show that our method achieves an average accuracy of around 1.28◦ without head movement and 2.27◦ with minor movement of the head.

TITLE: Face Detection, Pose Estimation, and Landmark Localization in the Wild[12]

DESCRIPTION: We present a unified model for face detection, pose estimation, and landmark estimation in real-world, cluttered images. Our model is based on a mixture of trees with a shared pool of parts; we model every facial landmark as a part and use global mixtures to capture topological changes due to viewpoint. We show that tree-structured models are surprisingly effective at capturing global elastic deformation, while being easy to optimize unlike dense graph structures. We present extensive results on standard face benchmarks, as well as a new "in the wild" annotated dataset, that suggests our system advances the state-of-theart, sometimes considerably, for all three tasks. Though our model is modestly trained with hundreds of faces, it compares favorably to commercial systems trained with billions of examples (such as Google Picasa and face.com).

TITLE: Face Verification using Correlation Filters[13]

DESCRIPTION: Face verification is an important tool for authentication of an individual and it can be of significant value in security and e-commerce applications. This paper deals with the application of correlation filters [1] for face verification. The performance of a specific type of correlation filter called the minimum average correlation energy (MACE) filter [2] is evaluated using a facial

expression database collected at the Advanced Multimedia Processing Lab at Carnegie Mellon University (CMU). A comparison of verification performance between the correlation filter method and Individual Eigenface Subspace Method (IESM) is also presented. It is seen that these correlation filters offer significant potential for face verification.

TITLE: Heuristic-Based Automatic Online Proctoring System[14]

DESCRIPTION: This paper proposes a novel multi-modal method for automatic online proctoring using a combination of image processing, audio processing and PC monitoring techniques. In the proposed system the remote proctor only has to inspect the examination room. The proposed system is evaluated with a dataset mimicking the malpractice scenarios, the accuracy of the system is demonstrated with false positive rate of 0.08 and true negative rate of 0.13.

TITLE: Human–computer interaction using vision-based hand gesture recognition systems: a survey[15]

DESCRIPTION: Considerable effort has been put toward the development of intelligent and natural interfaces between users and computer systems. In line with this endeavor, several modes of information (e.g., visual, audio, and pen) that are used either individually or in combination have been proposed. The use of gestures to convey information is an important part of human communication. Hand gesture recognition is widely used in many applications, such as in computer games, machinery control (e.g., crane), and thorough mouse replacement. Computer

recognition of hand gestures may provide a natural computer interface that allows people to point at or to rotate a computer-aided design model by rotating their hands. Hand gestures can be classified into two categories: static and dynamic. The use of hand gestures as a natural interface serves as a motivating force for research on gesture taxonomy, its representations, and recognition techniques. This paper summarizes the surveys carried out in human–computer interaction (HCI) studies and focuses on different application domains that use hand gestures for efficient interaction. This exploratory survey aims to provide a progress report on static and dynamic hand gesture recognition (i.e., gesture taxonomies, representations, and recognition techniques) in HCI and to identify future directions on this topic.

TITLE: Illumination-Free Gaze Estimation Method for First-Person Vision Wearable Device[16]

DESCRIPTION: Gaze estimation is a key technology to understand a person's interests and intents, and it is becoming more popular in daily situations such as driving scenarios. Wearable gaze estimation devices are used for long periods of time, therefore non-active sources are not desirable from a safety point of view. Gaze estimation that does not rely on active source, is performed by locating iris position. To estimate the iris position accurately, most studies use ellipse fitting in which the ellipse is defined by 5 parameters(position (x,y) , rotation angle, semi-major axis and semi-minor axis). We claim that, for iris position estimation, 5 parameters are redundant because they might be influenced by non-iris edges. Therefore, we propose to use 2 parameters(position) introducing a 3D eye model(the transformation between eye and camera coordinate and eyeball/iris size).

Given a 3D eye model, projected ellipses that represent iris shape can be specified only by position under weak-perspective approximation. We quantitatively evaluate our method on both iris position and gaze estimation. Our results show that our method outperforms other state-of-the-art iris estimation and is competitive to commercial products that use infrared ray with respect to both accuracy and robustness.

TITLE: Implementation of E-Proctoring in Online Teaching: A Study about Motivational Factors[17]

DESCRIPTION: Most online teaching institutions still do not offer complete remote teaching, requiring the physical presence of the student in the evaluation process (for supervisory reasons), which could aggravate the evaluation and certification in massive open online teaching. Although, there are already e-proctoring tools (electronic proctoring) that allow this process to be carried out remotely, without requiring that physical presence. For this reason, and in order for this complete remote teaching to be extended to institutions that do not yet believe in the success of its implementation, this study, through a bibliographic study and a causal study carried out by experts in online teaching, focuses on locating the determining motivational factors when accepting and implementing this evaluation system as a method of remote supervision and tries to encourage its use through them. The list obtained consists of the following motivational factors: Quality management, available information, external conditioning, trust, perceived compatibility, perceived usefulness, attitude and intention, and the most decisive

factor in this whole process is trust (which would be the extent of security and privacy that institutions have in the use of this tool).

TITLE: Video-based face model fitting using Adaptive Active Appearance Model[18]

DESCRIPTION: Active Appearance Model (AAM) represents the shape and appearance of an object via two low-dimensional subspaces, one for shape and one for appearance. AAM for facial images is currently receiving considerable attention from the computer vision community. However, most existing work focuses on fitting an AAM to a single image. For many applications, effectively fitting an AAM to video sequences is of critical importance and challenging, especially considering the varying quality of real-world video content. This paper proposes an Adaptive Active Appearance Model (AAAM) to address this problem, where both a generic AAM component and a subject-specific appearance model component are employed simultaneously in the proposed fitting scheme. While the generic AAM component is held fixed, the subject-specific model component is updated during the fitting process by selecting the frames that can be best explained by the generic model. Experimental results from both indoor and outdoor representative video sequences demonstrate the faster fitting convergence and improved fitting accuracy.

TITLE: Toward Constructing a Secure Online Examination System[19]

DESCRIPTION: Nowadays the rapid development of technology can be observed in every aspect of human life, with no exception to the educational world. In particular, implementation of a secure online examination system is a hot topic.

Issues that should be addressed in the secure online examination system are computer and network security issues of the system and prevention of cheatings by examinees. In our paper, we provide a website application and a secure network design which can prevent cheatings by examinees, and examine both of them in aspects of security and cheating prevention.

TITLE: Thwarting online exam cheating without proctor supervision[20]

DESCRIPTION: To demonstrate and maintain academic integrity, some institutions require proctor supervision of online exams. However, proctoring can be very expensive. Costs to students can include fees at testing centers, costs to purchase the Remote Proctor, time to find an approved proctor, and effort required to coordinate a time for the exam. Costs to the institution include salaries of staff to administer a proctoring process, approval of proctors, maintaining testing centers, and potential loss of enrollments and revenue since not all institutions require proctors for online exams. This paper examines the control issues related to online exams and asserts that the total cost of proctors for online exams (time and money of both students and the institution) exceed potential benefits. The authors propose a less costly, non-proctor alternative to promote academic honesty, using eight control procedures that enable faculty to increase the difficulty and thus reduce the likelihood of cheating by students.

TITLE: Online Exam Proctoring System[21]

DESCRIPTION: Online education is helping students and institutions worldwide to access a knowledge base of wide variety. This form of learning and education is increasing rapidly, Evaluation and proctoring for the online courses has become a major bottleneck for scalability of such learning systems. Manual human supervision is a common approach for exam proctoring and evaluation where the examiner needs to be present in the testing environment or needs to monitor the testing environment of a test taker visually and acoustically through a webcam. In our proposed system, we present a completely automated, exam proctoring solution that requires no human involvement. The system integrates all the inputs to process and estimate the variety of events, behaviors and patterns typically associated with cheating. By combining continuous identity verification and automatic detection of malpractice or suspicious activities by a student, this system provides a scalable, online, completely automated, human interaction free proctoring system that can be accessed by test takers and administrators to a truly efficient solution to conventional problem of online exam proctoring.

TITLE: Online Examination System with Cheating Prevention Using Question Bank Randomization and Tab Locking[22]

DESCRIPTION: Online examination system is used by educational institutions to improve the quality of instruction by having a supervised measure of outcomes for self-paced learning environments of their students. The reason E-learning became so popular is because of its fast feedback in assessing the examiners or candidates.

An online examination system that has the ability to address academic malpractice should be the main concern to be able to trim down those acts to some degree. Saving time is one of the perks of having an Online examination system, but it also has limitations on dependency to the quality of Internet service leaving both the proctor and the examiners not being able to use the system. The research looked into interviewing through a focus group the proctors of online exams to identify root causes of academic malpractice at the same time interview exam content creators on possible approaches on exam question generators that allow a validity of measure of outcomes. Generally, a final validation done by the focus group respondents and end users for effectivity and usability.

TITLE: The Research and Application of Online Examination and Monitoring System[23]

DESCRIPTION: The development of modern education technology has promoted the changes of teaching pattern and examination pattern, the appearance of online examination system (OES) is the best embodiment of these reforms, and the monitoring system is designed to ensure the fairness and impartiality of the OES. The paper provides the structure and function of OES and Monitoring System, discusses the key problems about communication and security and gives the solutions. The system is satisfied with the requirement of network examination well.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Online remote proctoring is the act of invigilating an online exam from any location to eliminate aberrant behavior. It is administered by experienced human proctors, by an AI Algorithm, or both, to maintain the examination process's integrity. The ability to efficiently proctor remote online examinations is an important limiting factor to the scalability of this next stage in education. Presently, human proctoring is the most common approach of evaluation, by either requiring the test taker to visit an examination center, or by monitoring them visually and acoustically during exams via a webcam. However, such methods are labor-intensive and costly. So, a multimedia analytics system that performs automatic online exam proctoring was presented. The system comprises variousmodules which may include hardware components such as a webcam, and a microphone, for the purpose of monitoring the visual and acoustic environment of the testing location. The system also employs software modules which continuously estimate the key behavior cues. These modules employ user verification, active window detection, gaze estimation and phone detection to process the examination process and prevent malpractices. By combining the continuous estimation components, and applying a temporal sliding window, and higher level features to classify whether the test taker is cheating at any moment during the exam can be easily determined.

3.1.1 USER VERIFICATION

An OEP system continuously verifies whether the test taker is who he claims to be throughout the entire exam session. The test taker is also expected to take the exam alone without the aid of another person in the room. While there are various

options for continuous user authentication, such as keystroke dynamics, commonly

Face verification is used due to its robustness.

3.1.2 ACTIVE WINDOW DETECTION

The Internet and computers are an open gateway to valuable information answering exam questions. Commonly Black boards Respondus Lockdown Browser (RLB) is used to access the online exam. RLB is a special browser where the test taker is locked into the exam and has no way to exit/return, cut/paste, or electronically manipulate the system. However, some exams might require Internet access to some specific websites, or perhaps the use of e-mail or chat functions.

Moreover, some test takers might have saved files and documents on the computer containing answers to the exam. Therefore, it is critical to keep track of how many windows the test taker is opening. So, the user is granted full Internet and computer access during the exam. Periodically estimation of the number of active windows running in the system is obtained from the operational system API. Most of the time, there should be only one active window, which is the online exam itself. If there are more than one window open at a specific time t during the exam, the system assumes that the test taker is cheating, and a warning will be displayed on the monitor requesting an immediate shutdown of the opened window. The probability of cheating increases as the test taker keeps the unexpected window opened longer. Since this component relies on the operational system API, the accuracy of active window detection is 100%.

3.1.3 GAZE ESTIMATION

In traditional classroom-based proctoring, the abnormal head gaze direction and its dynamics over time can be a strong indicator of potential cheating. An abnormal gaze is when the test taker’s eyes are off the screen for an extended period

of time, or if the head quickly gazes around a few times. Although abnormal gaze does not directly constitute a cheating behavior, it is an important cue to suggest the potential subsequent cheating actions. As a classic computer vision problem, head gaze estimation is a particularly challenging problem in any application due to the spontaneous head motion of the test taker as well as the partial occlusion by the cameras. To address this issue, algorithms that use various visual sensors to enhance head gaze estimation are implemented. From the webcam, we may estimate the gaze from the face in the video frame. By combining the information from the cameras, accurate estimates of the head gaze of the test taker in a wide range of yaw and pitch angles are obtained.

3.1.4 PHONE DETECTION

Most online exam rules prohibit the use of any type of mobile phones. Therefore, the presence of a mobile phone in the examination environment can be an indication of potential cheating. With advancements in mobile phone technology, there are many ways to cheat from them, such as reading saved notes, text messaging friends, browsing the Internet, and taking a snapshot of the exam to share with other test takers. Phone detection is challenging due to the various sizes, models and shapes of phones (a tablet could also be considered a type of phone). Some test takers might have large touch screens while others might use button-based flip phones. Moreover, cheating from a phone is usually accompanied with various occlusions, such as holding the phone under the desk, or covering part of the phone with their hand. To enable this capability, the video captured from the Webcam is utilized, since it sees what the test taker is seeing. The phone detection is based on a simple approach, i.e., searching for pixels that are brighter than the background pixels. The motivation of using the screen’s brightness over detecting the phone object, is that the system shouldn't claim there is a phone-based cheating

behavior unless the phone is switched on. So, the estimated phone screen is represented by using the area of the local region.

3.2 PROPOSED SYSTEM

This system presents a multimedia analytics system for remote online exam proctoring, which aims to maintain academic integrity in e-learning. The system is affordable and convenient to use from the test taker's perspective, since it only requires having inexpensive web cameras and a microphone. These features are then processed in a temporal window to acquire high-level features, and then are used for cheat detection. The main contribution of this work is to present a comprehensive framework for remote online exam proctoring. While we have achieved good performance in our evaluation, our framework can certainly be improved in a number of ways. For the basic components, we can either apply more advanced algorithms for each component, such as the deep learning-based feature representation, typing-based continuous authentication, face alignment-based pose estimation, upper body alignment, and model personalization. We may also expand the array of basic components, to include additional components such as pen detection. For cheat classification, we can explore temporal-spatial dynamic features, similar to the work in video-based activity recognition. Moreover, the system efficiency can also be improved while maintaining a high accuracy in recognizing cheat events, by selecting more suitable features and classifiers, as well as selecting a smaller number of frames instead of utilizing all frames. Apart from the basic components and the inherited components from different use cases this system implements novel modules such as: user verification, audio processing, gaze detection, number of person detection, object and phone detection.

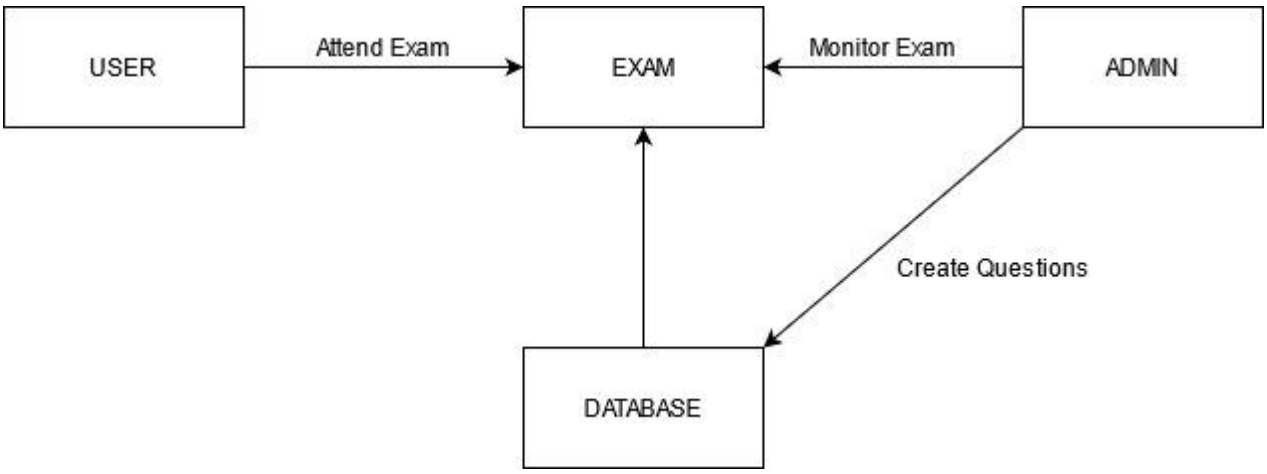


Fig 3.1 Basic block diagram of E-Proctoring System

3.2.1 USER VERIFICATION

In this module, the system continuously verifies whether the test taker is who he claims to be throughout the entire exam session. It is implemented by the use of face_recognition algorithm from OpenCV which uses feed from the web camera to process the data and determine whether the candidate enrolled is the same as the one taking the examination at any given point of time. The test taker is also expected to take the exam alone without the aid of another person in the room.

3.2.2 NUMBER OF PERSON DETECTION

In any kind of examination, the test taker is expected to take the exam alone without the aid of another person in the room. So, the system uses the input video feed from the web camera to verify whether there is any other persons' presence in

the examination environment. Most probably, the candidates' environment will be any confined spot in a room. So, the frame for reference of verification of any other persons' presence will have to be set within a specific spot. The module makes sure that this processing of input data follows the specified algorithms. YoloV3 from Keras TensorFlow is an algorithm implemented in this module.

3.2.3 OBJECT AND PHONE DETECTION

The presence of any kind of mobile phones in the examination environment can be an indication of potential cheating. With advancements in mobile phone technology, there are many ways to cheat from them, such as reading saved notes, text messaging friends, browsing the Internet, and taking a snapshot of the exam to share with other test takers. This is to be prevented effectively because most of the Malpractice techniques involve mobile phones. So, by implementing YoloV3 from Keras TensorFlow algorithm the system achieves at most robustness for this use case.

3.2.4 GAZE DETECTION

Any abnormal head gaze in any inappropriate direction with context to its dynamics over time can be a strong indicator of potential cheating. An abnormal gaze is when the test taker's eyes are off the screen for an extended period of time, or if the head quickly gazes around a few times. Although abnormal gaze does not directly constitute a cheating behavior, it is an important cue to suggest the potential

subsequent cheating actions. Head gaze estimation is a particularly challenging problem in any application involving classic computer vision due to the spontaneity in the head motion of the test taker as well as the partial occlusion by the cameras. This issue is addressed by implementing algorithms which will be imported from Keras TensorFlow and Computer vision. From the webcam, we may estimate the gaze from the face in the video frame by combining the information from the cameras, accurate estimates of the head gaze of the test taker in a wide range of yaw and pitch angles are obtained.

3.2.5 AUDIO PROCESSING

One of the most likely fraudulent behaviors in online exams is to seek verbal assistance from another person in the same room, or remotely via a phone call. This is the most frequent cheating behavior. By requiring the test taker to take the exam in a quiet room with no one around, any human speech being detected could be considered a potential cheating instance. Therefore, the module uses Natural Language Processing Toolkits in this component to detect speech from acoustic signals. This module predominantly employs keywords based identification to detect potential instances of fraudulent behaviour.

3.3 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

3.3.1 TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

3.3.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn't incur any kind of drastic increase in the expenses. Python is open source and readily available for all users. Since the project is run in python and jupyter notebook hence is cost efficient.

3.4 SYSTEM CONFIGURATION

3.4.1 HARDWARE CONFIGURATION

- Processor - Intel Core i5 and above, AMD Ryzen 3 and above
- RAM - 4 GB
- Hard Disk - 500 GB

3.4.2 SOFTWARE CONFIGURATION

- Operating System - Windows 7 and above
- Programming Language : Python 3.8
- Python Modules : TensorFlow 2.2

3.5 SOFTWARE SPECIFICATION

3.5.1 OPENCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate

the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people in the user community and an estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow

Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interface is being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.5.2NATURAL LANGUAGE PROCESSING TOOLKIT

Natural Language Processing (NLP) is a process of manipulating or understanding the text or speech by any software or machine. An analogy is that humans interact and understand each other's views and respond with the appropriate answer. In NLP, this interaction, understanding, and response are made by a computer instead of a human.correct answers. For example, a classification algorithm will learn to identify animals after being trained on a dataset of images that are properly labeled with the species of the animal and some identifying

characteristics. Supervised learning problems can be further grouped into Regression and Classification problems. Both problems have as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification.

NLTK (Natural Language Toolkit) is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. There are 32 universities in the US and 25 countries using NLTK in their courses. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

3.5.3 TENSORFLOW

TensorFlow is a free and open-sourcesoftware library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.Tensorflow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs. TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of

computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

TensorFlow computations are expressed as stateful dataflowgraphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as *tensors*. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.

3.5.4 REACT

React (React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

3.5.5 AWS

Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demandcloud computingplatforms and APIs to individuals, companies, and

governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtualcluster of computers, available all the time, through the Internet. AWS's version of virtual computers emulates most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement, Amazon provides security for subscribers' systems. AWS operates from many global geographical regions including 6 in North America.

Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns a dominant 33% of all cloud (IaaS, PaaS) while the next two competitors Microsoft and Google have 18%, 9% respectively according to Synergy Group.

CHAPTER 4

ARCHITECTURE

4. ARCHITECTURE

4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

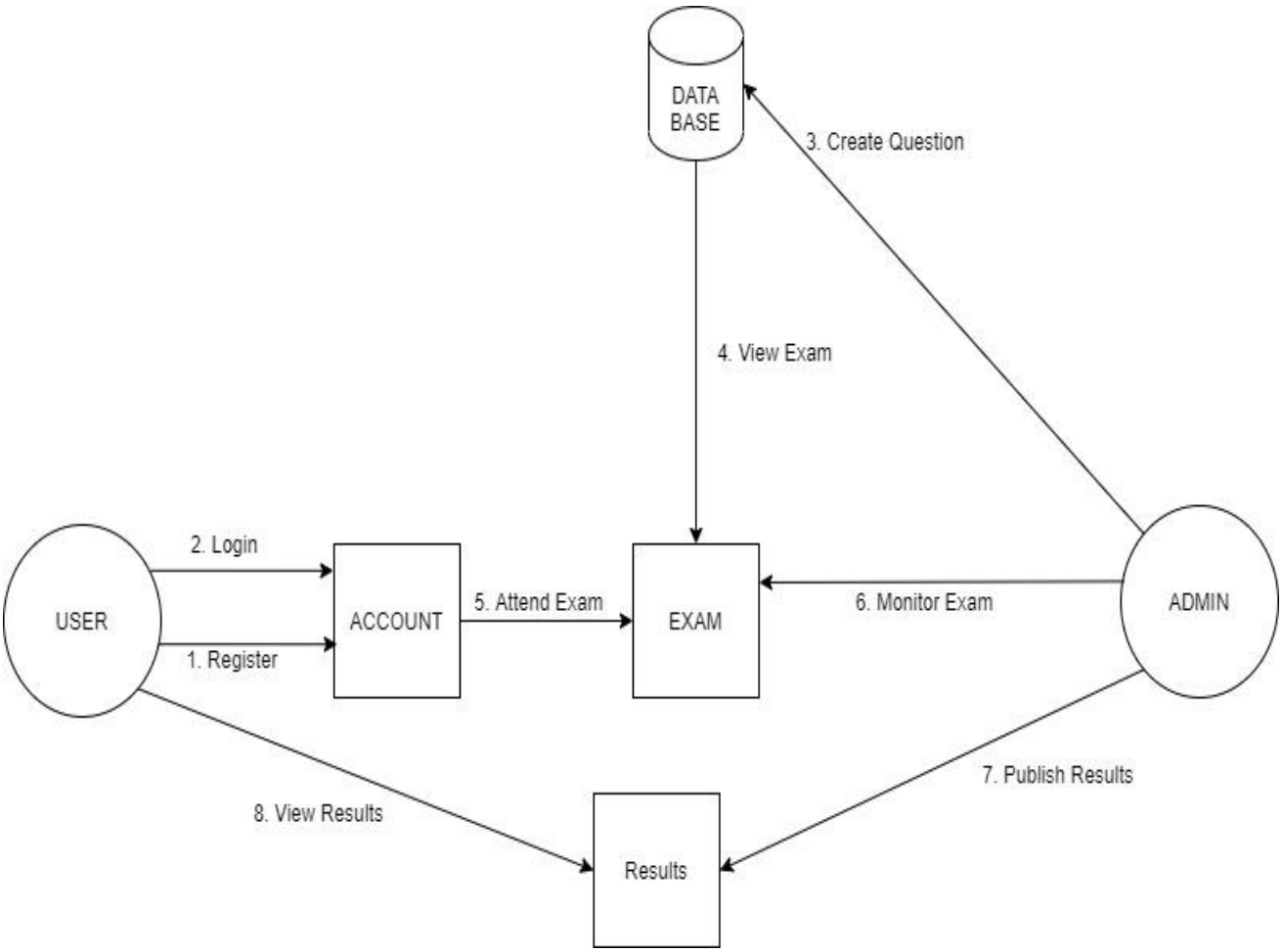


Fig 4.1.1 Architecture Diagram of Remote E-Proctoring System

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM

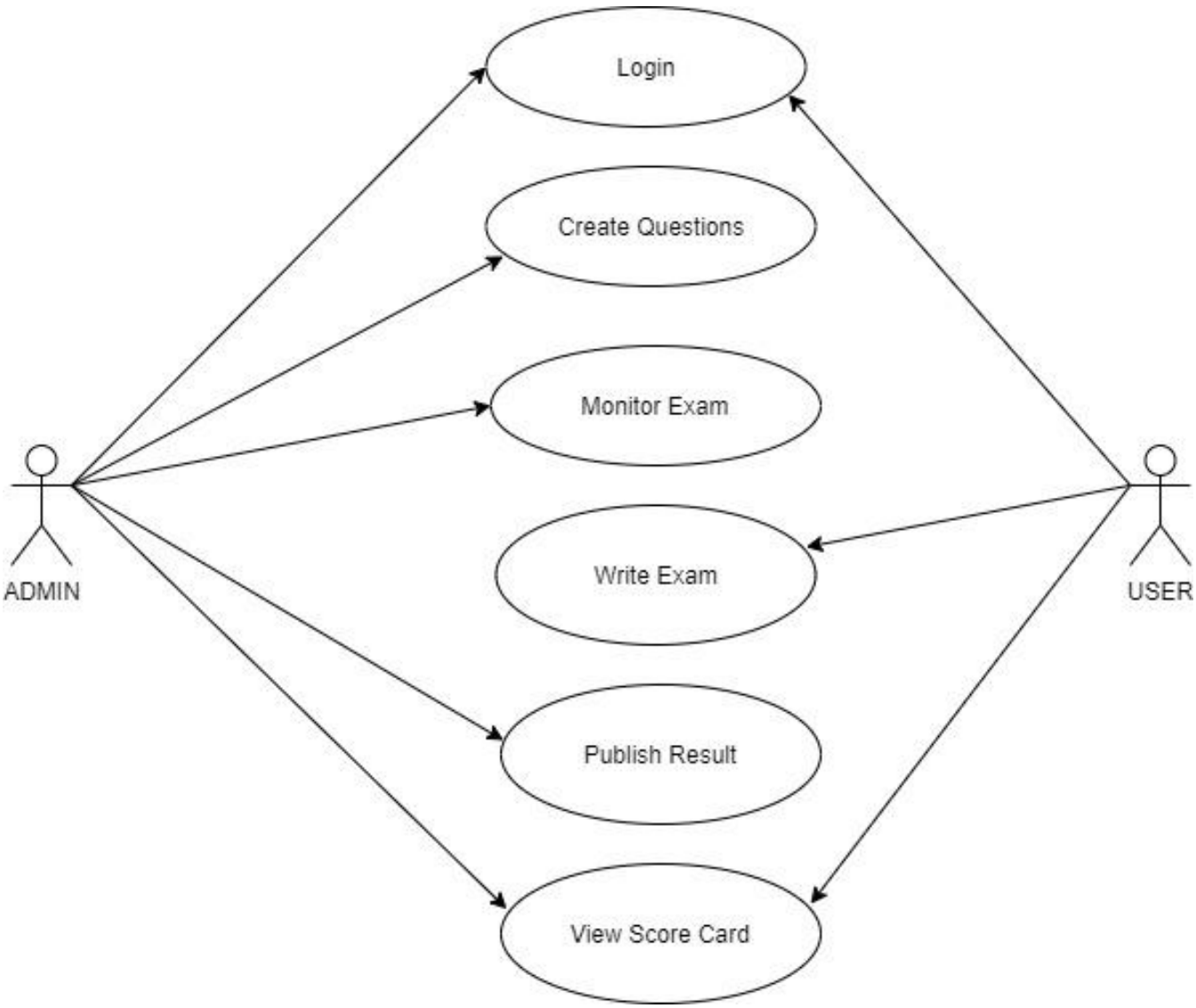


Fig 4.2.1 Use Case Diagram of Remote E-Proctoring System

4.2.2 ER DIAGRAM

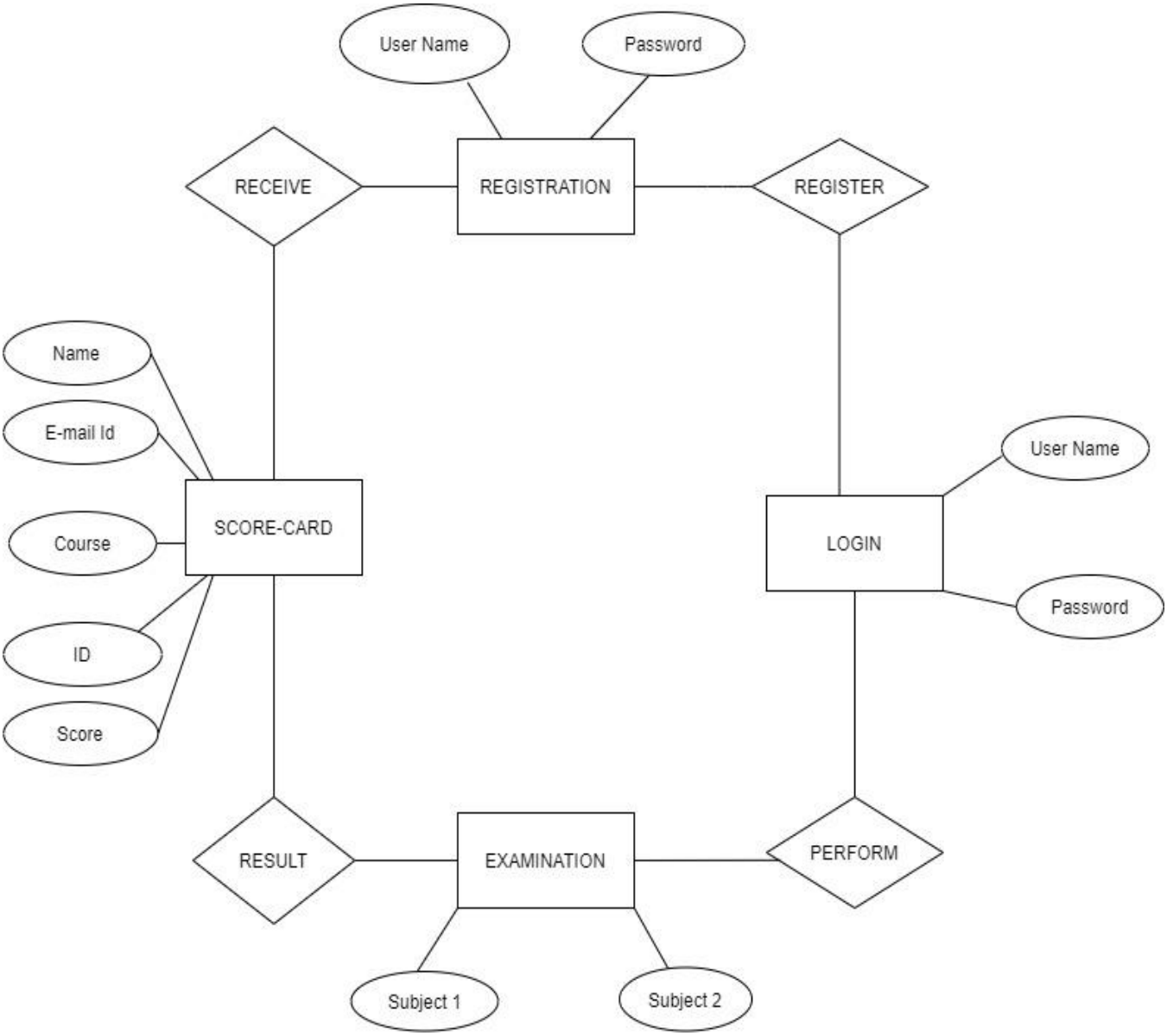


Fig 4.2.2ER Diagram of Remote E-Proctoring System

4.2.3 SEQUENCE DIAGRAM

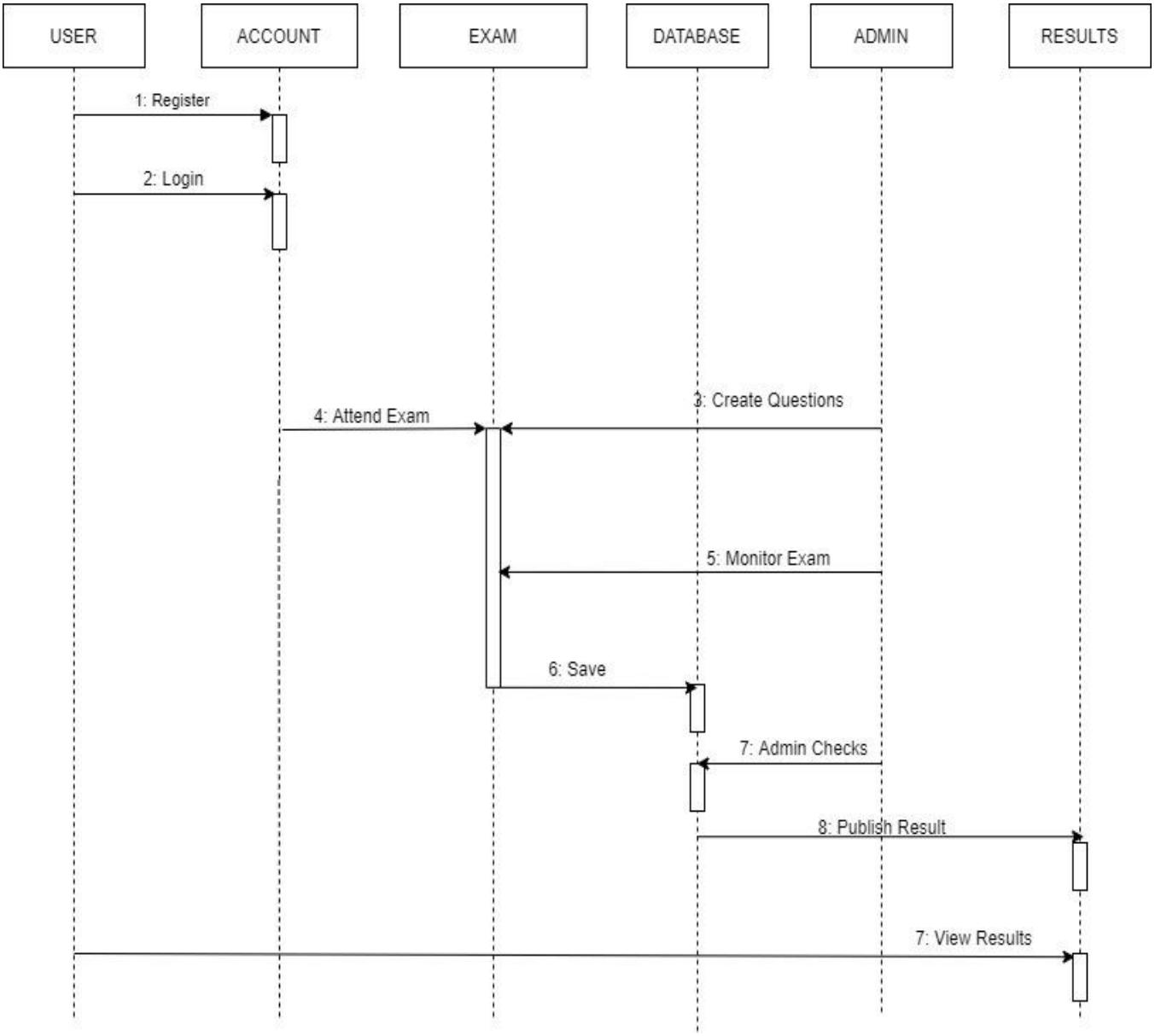


Fig 4.2.3 Sequence Diagram of Remote E-Proctoring System

4.2.4 DATA FLOW DIAGRAM

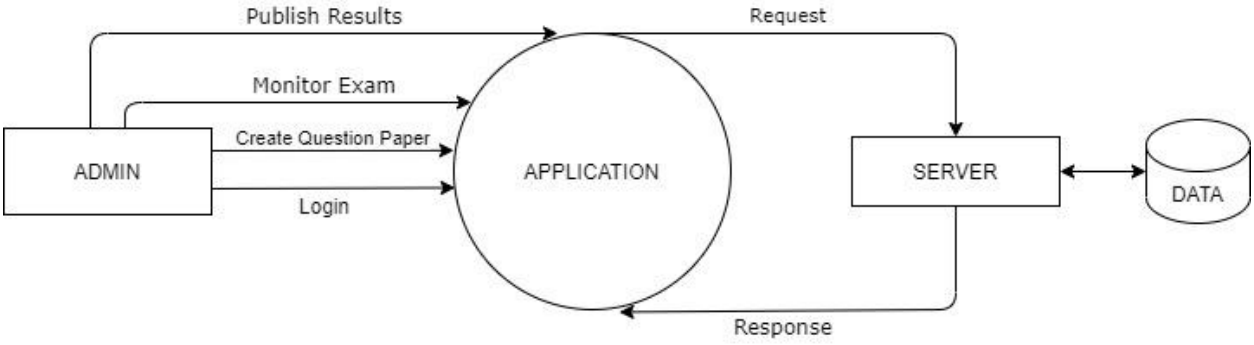


Fig 4.2.4.1 Admin Data Flow Diagram of Remote E-Proctoring System

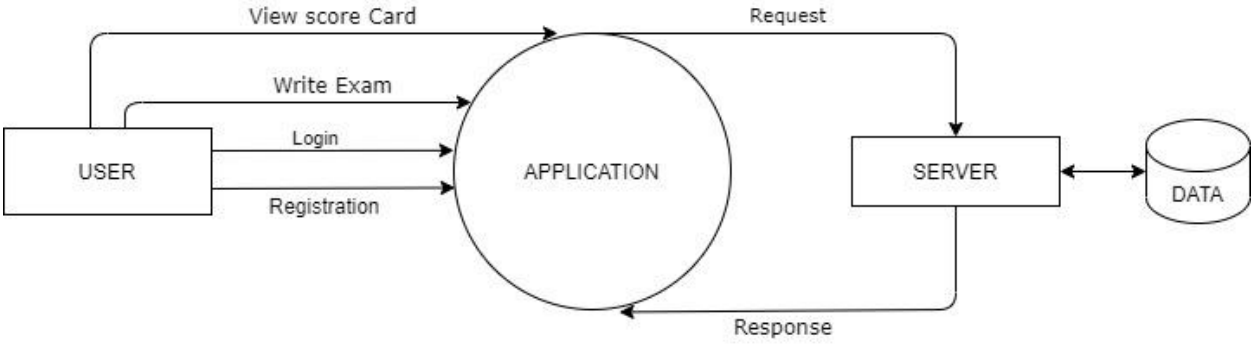


Fig 4.2.4.2 User Data Flow Diagram of Remote E-Proctoring System

CHAPTER 5

SYSTEM MODULE

5. SYSTEM MODULE

5.1 MODULE

- User Verification
- Number of person detection
- Object and Phone detection
- Gaze detection
- Audio processing
- Yolo algorithm.

5.2 MODULE DESCRIPTION

5.2.1 USER VERIFICATION

In this module, the system continuously verifies whether the test taker is who he claims to be throughout the entire exam session. It is implemented by the use of face_recognition algorithm from Opencv which uses feed from the web camera to process the data and determine whether the candidate enrolled is the same as the one taking the examination at any given point of time. The test taker is also expected to take the exam alone without the aid of another person in the room. Implementation for the same, is explained in the next module.

5.2.2 NUMBER OF PERSON DETECTION

In any kind of examination, the test taker is expected to take the exam alone without the aid of another person in the room. So, the system uses the input video feed from the web camera to verify whether there is any other persons' presence in the examination environment. Most probably, the candidates' environment will be any confined spot in a room. So, the frame for reference of verification of any other persons' presence will have to be set within a specific spot. The module makes sure that this processing of input data follows the specified algorithms. YoloV3 from Keras TensorFlow is an algorithm implemented in this module.

5.2.3 OBJECT AND PHONE DETECTION

The presence of any kind of mobile phones in the examination environment can be an indication of potential cheating. With advancements in mobile phone technology, there are many ways to cheat from them, such as reading saved notes, text messaging friends, browsing the Internet, and taking a snapshot of the exam to share with other test takers. This is to be prevented effectively because most of the Malpractice techniques involve mobile phones. So, by implementing YoloV3 from Keras TensorFlow algorithm the system achieves at most robustness for this use case.

5.2.4 GAZE DETECTION

Any abnormal head gaze in any inappropriate direction with context to its dynamics over time can be a strong indicator of potential cheating. An abnormal gaze is when the test taker's eyes are off the screen for an extended period of time, or if the head quickly gazes around a few times. Although abnormal gaze does not directly constitute a cheating behavior, it is an important cue to suggest the potential subsequent cheating actions. Head gaze estimation is a particularly challenging problem in any application involving classic computer vision due to the spontaneity in the head motion of the test taker as well as the partial occlusion by the cameras. This issue is addressed by implementing algorithms which will be imported from Keras TensorFlow and Computer vision.

From the webcam, we may estimate the gaze from the face in the video frame by combining the information from the cameras, accurate estimates of the head gaze of the test taker in a wide range of yaw and pitch angles are obtained.

5.2.5 AUDIO PROCESSING

One of the most likely fraudulent behaviors in online exams is to seek verbal assistance from another person in the same room, or remotely via a phone call. This is the most frequent cheating behavior. By requiring the test taker to take the exam in a quiet room with no one around, any human speech being detected

could be considered a potential cheating instance. Therefore, the module uses Natural Language Processing Toolkits in this component to detect speech from acoustic signals. This module predominantly employs keywords based identification to detect potential instances of fraudulent behaviour.

5.2.6 YOLO ALGORITHM

Yolo is an algorithm that uses convolutional neural networks for object detection. In comparison to recognition algorithms, a detection algorithm does not only predict class labels, but detects locations of objects as well.The algorithm divides the image into grids and runs the image classification and localization algorithm (discussed under object localization) on each of the grid cells. For example, we have an input image of size 256 × 256. We place a 3 × 3 grid on the image.

CHAPTER 6

SYSTEM DESIGN

6. SYSTEM DESIGN

6.1 USER VERIFICATION

```
import face_recognition

import cv2


video_capture = cv2.VideoCapture(0)

Manish_image =
face_recognition.load_image_file("/Users/msml/Vamsi/Face/images/Manish.jpeg")

Manish_face_encoding = face_recognition.face_encodings(Manish_image)[0]

Maddy_image =
face_recognition.load_image_file("/Users/msml/Vamsi/Face/images/Maddy.jpeg")

Maddy_face_encoding = face_recognition.face_encodings(Maddy_image)[0]

Niteesh_image = face_recognition.load_image_file("/Users/msml/E-
Proctoring/images/Niteesh.jpeg")

Niteesh_face_encoding = face_recognition.face_encodings(Niteesh_image)[0]

Rahul_image =
face_recognition.load_image_file("/Users/msml/Vamsi/Face/images/Rahul.jpeg")

Rahul_face_encoding = face_recognition.face_encodings(Rahul_image)[0]

Vamsi_image = face_recognition.load_image_file("/Users/msml/E-
Proctoring/images/Vamsi.jpeg")

Vamsi_face_encoding = face_recognition.face_encodings(Vamsi_image)[0]

Ashwin_image = face_recognition.load_image_file("/Users/msml/E-
Proctoring/images/Ashwin.jpeg")
```

```
Ashwin_face_encoding = face_recognition.face_encodings(Ashwin_image)[0]
known_face_encodings = [
        Niteesh_face_encoding,
        Vamsi_face_encoding,
        Ashwin_face_encoding,
]
known_face_names = [
        " Niteesh",
        " Vamsi",
        "Ashwin",
]
face_locations = []
face_encodings = []
face_names = []
all_time_faces_names = []
Unknown_array = []
process_this_frame = True

while True:
        ret, frame = video_capture.read()
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, ::-1]
```

```
        if process_this_frame:
        face_locations =
face_recognition.face_locations(rgb_small_frame,number_of_times_to_upsample=
2)
        face_encodings = face_recognition.face_encodings(rgb_small_frame,
face_locations)


        face_names = []
        for face_encoding in face_encodings:
        matches = face_recognition.compare_faces(known_face_encodings,
face_encoding)
        name = "Unknown"
        if True in matches:
        first_match_index = matches.index(True)
        name = known_face_names[first_match_index]
        face_names.append(name)
        if name == "Unknown":
        Unknown_array.append(name)
        else:
        if name not in all_time_faces_names:
        all_time_faces_names.append(name)
        process_this_frame = not process_this_frame
```

```
        #for (top, right, bottom, left), name in zip(face_locations, face_names):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255),
cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
        cv2.imshow('Video', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video_capture.release()
cv2.destroyAllWindows()
```

6.2 GAZE ESTIMATION

```
import cv2
from face_detector import get_face_detector, find_faces
from face_landmarks import get_landmark_model, detect_marks, draw_marks

face_model = get_face_detector()
landmark_model = get_landmark_model()
outer_points = [[49, 59], [50, 58], [51, 57], [52, 56], [53, 55]]
d_outer = [0]*5
inner_points = [[61, 67], [62, 66], [63, 65]]
d_inner = [0]*3
font = cv2.FONT_HERSHEY_SIMPLEX
cap = cv2.VideoCapture(0)

while(True):
        ret, img = cap.read()
        rects = find_faces(img, face_model)
        for rect in rects:
        shape = detect_marks(img, landmark_model, rect)
        draw_marks(img, shape)
        cv2.putText(img, 'Press r to record Mouth distances', (30, 30), font,
```

```
        1, (0, 255, 255), 2)
        cv2.imshow("Output", img)
        if cv2.waitKey(1) & 0xFF == ord('r'):
        for i in range(100):
        for j, (p1, p2) in enumerate(outer_points):
        d_outer[j] += shape[p2][1] - shape[p1][1]
        for k, (p1, p2) in enumerate(inner_points):
        d_inner[k] += shape[p2][1] - shape[p1][1]
        break
cv2.destroyAllWindows()
d_outer[:] = [x / 100 for x in d_outer]
d_inner[:] = [x / 100 for x in d_inner]

while(True):
        ret, img = cap.read()
        rects = find_faces(img, face_model)
        for rect in rects:
        shape = detect_marks(img, landmark_model, rect)
        cnt_outer = 0
        cnt_inner = 0
        draw_marks(img, shape[48:])
        for i, (p1, p2) in enumerate(outer_points):
```

```
        if d_outer[i] + 3 < shape[p2][1] - shape[p1][1]:
        cnt_outer += 1
        for i, (p1, p2) in enumerate(inner_points):
        if d_inner[i] + 2 <  shape[p2][1] - shape[p1][1]:
        cnt_inner += 1
        if cnt_outer > 3 and cnt_inner > 2:
        print('Mouth open')
        cv2.putText(img, 'Mouth open', (30, 30), font,
        1, (0, 255, 255), 2)
        # show the output image with the face detections + facial landmarks
        cv2.imshow("Output", img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

6.3 NUMBER OF PERSON DETECTION AND PHONE DETECTION

```
import tensorflow as tf
import numpy as np
import cv2

from tensorflow.keras import Model
from tensorflow.keras.layers import (
        Add,
        Concatenate,
        Conv2D,
        Input,
        Lambda,
        LeakyReLU,
        UpSampling2D,
        ZeroPadding2D,
        BatchNormalization
)
from tensorflow.keras.regularizers import l2
import wget
def load_darknet_weights(model, weights_file):
        '''
```

```
Helper function used to load darknet weights.
:param model: Object of the Yolo v3 model
:param weights_file: Path to the file with Yolo V3 weights
'''

#Open the weights file
wf = open(weights_file, 'rb')
major, minor, revision, seen, _ = np.fromfile(wf, dtype=np.int32, count=5)

#Define names of the Yolo layers (just for a reference)
layers = ['yolo_darknet',
'yolo_conv_0',
'yolo_output_0',
'yolo_conv_1',
'yolo_output_1',
'yolo_conv_2',
'yolo_output_2']

for layer_name in layers:
sub_model = model.get_layer(layer_name)
for i, layer in enumerate(sub_model.layers):
```

```
if not layer.name.startswith('conv2d'):
continue

#Handles the special, custom Batch normalization layer
batch_norm = None
if i + 1 < len(sub_model.layers) and \
sub_model.layers[i + 1].name.startswith('batch_norm'):
batch_norm = sub_model.layers[i + 1]

filters = layer.filters
size = layer.kernel_size[0]
in_dim = layer.input_shape[-1]

if batch_norm is None:
conv_bias = np.fromfile(wf, dtype=np.float32, count=filters)
else:
# darknet [beta, gamma, mean, variance]
bn_weights = np.fromfile(
wf, dtype=np.float32, count=4 * filters)
# tf [gamma, beta, mean, variance]
bn_weights = bn_weights.reshape((4, filters))[[1, 0, 2, 3]]
```

```
        # darknet shape (out_dim, in_dim, height, width)
        conv_shape = (filters, in_dim, size, size)
        conv_weights = np.fromfile(
        wf, dtype=np.float32, count=np.product(conv_shape))
        # tf shape (height, width, in_dim, out_dim)
        conv_weights = conv_weights.reshape(
        conv_shape).transpose([2, 3, 1, 0])

        if batch_norm is None:
        layer.set_weights([conv_weights, conv_bias])
        else:
        layer.set_weights([conv_weights])
        batch_norm.set_weights(bn_weights)

        assert len(wf.read()) == 0, 'failed to read all data'
        wf.close()

def draw_outputs(img, outputs, class_names):
        '''
        Helper, util, function that draws predictons on the image.

        :param img: Loaded image
```

```
    :param outputs: YoloV3 predictions
    :param class_names: list of all class names found in the dataset
    '''
    boxes, objectness, classes, nums = outputs
    boxes, objectness, classes, nums = boxes[0], objectness[0], classes[0],
nums[0]
    wh = np.flip(img.shape[0:2])
    for i in range(nums):
    x1y1 = tuple((np.array(boxes[i][0:2]) * wh).astype(np.int32))
    x2y2 = tuple((np.array(boxes[i][2:4]) * wh).astype(np.int32))
    img = cv2.rectangle(img, x1y1, x2y2, (255, 0, 0), 2)
    img = cv2.putText(img, '{} {:.4f}'.format(
    class_names[int(classes[i])], objectness[i]),
    x1y1, cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0, 255), 2)
    return img

yolo_anchors = np.array([(10, 13), (16, 30), (33, 23), (30, 61), (62, 45),
    (59, 119), (116, 90), (156, 198), (373, 326)],
    np.float32) / 416

yolo_anchor_masks = np.array([[6, 7, 8], [3, 4, 5], [0, 1, 2]])
```

```
def DarknetConv(x, filters, kernel_size, strides=1, batch_norm=True):
    '''
    Call this function to define a single Darknet convolutional layer

    :param x: inputs
    :param filters: number of filters in the convolutional layer
    :param kernel_size: Size of kernel in the Conv layer
    :param strides: Conv layer strides
    :param batch_norm: Whether or not to use the custom batch norm layer.
    '''
    #Image padding
    if strides == 1:
    padding = 'same'
    else:
    x = ZeroPadding2D(((1, 0), (1, 0)))(x)  # top left half-padding
    padding = 'valid'

    #Defining the Conv layer
    x = Conv2D(filters=filters, kernel_size=kernel_size,
    strides=strides, padding=padding,
    use_bias=not batch_norm, kernel_regularizer=l2(0.0005))(x)
```

```
        if batch_norm:
        x = BatchNormalization()(x)
        x = LeakyReLU(alpha=0.1)(x)
        return x

def DarknetResidual(x, filters):
        '''
        Call this function to define a single DarkNet Residual layer

        :param x: inputs
        :param filters: number of filters in each Conv layer.
        '''
        prev = x
        x = DarknetConv(x, filters // 2, 1)
        x = DarknetConv(x, filters, 3)
        x = Add()([prev, x])
        return x


def DarknetBlock(x, filters, blocks):
        '''
```

```
    Call this function to define a single DarkNet Block (made of multiple
Residual layers)

    :param x: inputs
    :param filters: number of filters in each Residual layer
    :param blocks: number of Residual layers in the block
    '''
    x = DarknetConv(x, filters, 3, strides=2)
    for _ in range(blocks):
    x = DarknetResidual(x, filters)
    return x


def Darknet(name=None):
    '''
    The main function that creates the whole DarkNet.
    '''
    x = inputs = Input([None, None, 3])
    x = DarknetConv(x, 32, 3)
    x = DarknetBlock(x, 64, 1)
    x = DarknetBlock(x, 128, 2)  # skip connection
    x = x_36 = DarknetBlock(x, 256, 8)  # skip connection
    x = x_61 = DarknetBlock(x, 512, 8)
```

```
    x = DarknetBlock(x, 1024, 4)
    return tf.keras.Model(inputs, (x_36, x_61, x), name=name)

def YoloConv(filters, name=None):
    '''
    Call this function to define the Yolo Conv layer.

    :param flters: number of filters for the conv layer
    :param name: name of the layer
    '''
    def yolo_conv(x_in):
    if isinstance(x_in, tuple):
    inputs = Input(x_in[0].shape[1:]), Input(x_in[1].shape[1:])
    x, x_skip = inputs

    # concat with skip connection
    x = DarknetConv(x, filters, 1)
    x = UpSampling2D(2)(x)
    x = Concatenate()([x, x_skip])
    else:
    x = inputs = Input(x_in.shape[1:])
```

```
        x = DarknetConv(x, filters, 1)
        x = DarknetConv(x, filters * 2, 3)
        x = DarknetConv(x, filters, 1)
        x = DarknetConv(x, filters * 2, 3)
        x = DarknetConv(x, filters, 1)
        return Model(inputs, x, name=name)(x_in)
        return yolo_conv

def YoloOutput(filters, anchors, classes, name=None):
        '''
        This function defines outputs for the Yolo V3. (Creates output projections)

        :param filters: number of filters for the conv layer
        :param anchors: anchors
        :param classes: list of classes in a dataset
        :param name: name of the layer
        '''
        def yolo_output(x_in):
        x = inputs = Input(x_in.shape[1:])
        x = DarknetConv(x, filters * 2, 3)
        x = DarknetConv(x, anchors * (classes + 5), 1, batch_norm=False)
        x = Lambda(lambda x: tf.reshape(x, (-1, tf.shape(x)[1], tf.shape(x)[2],
```

```
        anchors, classes + 5)))(x)
        return tf.keras.Model(inputs, x, name=name)(x_in)
        return yolo_output

def yolo_boxes(pred, anchors, classes):
        '''
        Call this function to get bounding boxes from network predictions

        :param pred: Yolo predictions
        :param anchors: anchors
        :param classes: List of classes from the dataset
        '''

        # pred: (batch_size, grid, grid, anchors, (x, y, w, h, obj, ...classes))
        grid_size = tf.shape(pred)[1]
        #Extract box coortinates from prediction vectors
        box_xy, box_wh, objectness, class_probs = tf.split(
        pred, (2, 2, 1, classes), axis=-1)

        #Normalize coortinates
        box_xy = tf.sigmoid(box_xy)
        objectness = tf.sigmoid(objectness)
```

```
    class_probs = tf.sigmoid(class_probs)
    pred_box = tf.concat((box_xy, box_wh), axis=-1)  # original xywh for loss

    # !!! grid[x][y] == (y, x)
    grid = tf.meshgrid(tf.range(grid_size), tf.range(grid_size))
    grid = tf.expand_dims(tf.stack(grid, axis=-1), axis=2)  # [gx, gy, 1, 2]

    box_xy = (box_xy + tf.cast(grid, tf.float32)) / \
    tf.cast(grid_size, tf.float32)
    box_wh = tf.exp(box_wh) * anchors

    box_x1y1 = box_xy - box_wh / 2
    box_x2y2 = box_xy + box_wh / 2
    bbox = tf.concat([box_x1y1, box_x2y2], axis=-1)

    return bbox, objectness, class_probs, pred_box

def yolo_nms(outputs, anchors, masks, classes):
    # boxes, conf, type
    b, c, t = [], [], []

    for o in outputs:
```

```
        b.append(tf.reshape(o[0], (tf.shape(o[0])[0], -1, tf.shape(o[0])[-1])))
        c.append(tf.reshape(o[1], (tf.shape(o[1])[0], -1, tf.shape(o[1])[-1])))
        t.append(tf.reshape(o[2], (tf.shape(o[2])[0], -1, tf.shape(o[2])[-1])))

        bbox = tf.concat(b, axis=1)
        confidence = tf.concat(c, axis=1)
        class_probs = tf.concat(t, axis=1)

        scores = confidence * class_probs
        boxes, scores, classes, valid_detections =
tf.image.combined_non_max_suppression(
        boxes=tf.reshape(bbox, (tf.shape(bbox)[0], -1, 1, 4)),
        scores=tf.reshape(
        scores, (tf.shape(scores)[0], -1, tf.shape(scores)[-1])),
        max_output_size_per_class=100,
        max_total_size=100,
        iou_threshold=0.5,
        score_threshold=0.6
        )

        return boxes, scores, classes, valid_detections
```

```
def YoloV3(size=None, channels=3, anchors=yolo_anchors,
        masks=yolo_anchor_masks, classes=80):

        x = inputs = Input([size, size, channels], name='input')

        x_36, x_61, x = Darknet(name='yolo_darknet')(x)

        x = YoloConv(512, name='yolo_conv_0')(x)
        output_0 = YoloOutput(512, len(masks[0]), classes, name='yolo_output_0')(x)

        x = YoloConv(256, name='yolo_conv_1')((x, x_61))
        output_1 = YoloOutput(256, len(masks[1]), classes, name='yolo_output_1')(x)

        x = YoloConv(128, name='yolo_conv_2')((x, x_36))
        output_2 = YoloOutput(128, len(masks[2]), classes, name='yolo_output_2')(x)

        boxes_0 = Lambda(lambda x: yolo_boxes(x, anchors[masks[0]], classes),
        name='yolo_boxes_0')(output_0)
        boxes_1 = Lambda(lambda x: yolo_boxes(x, anchors[masks[1]], classes),
        name='yolo_boxes_1')(output_1)
        boxes_2 = Lambda(lambda x: yolo_boxes(x, anchors[masks[2]], classes),
```

```
        name='yolo_boxes_2')(output_2)

        outputs = Lambda(lambda x: yolo_nms(x, anchors, masks, classes),
        name='yolo_nms')((boxes_0[:3], boxes_1[:3], boxes_2[:3]))

        return Model(inputs, outputs, name='yolov3')

def weights_download(out='models/yolov3.weights'):
        _ = wget.download('https://pjreddie.com/media/files/yolov3.weights',
out='models/yolov3.weights')

# weights_download() # to download weights
yolo = YoloV3()
load_darknet_weights(yolo, 'models/yolov3.weights')

cap = cv2.VideoCapture(0)


while(True):
        ret, image = cap.read()
        if ret == False:
        break
```

```
img = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (320, 320))
img = img.astype(np.float32)
img = np.expand_dims(img, 0)
img = img / 255
class_names = [c.strip() for c in open("models/classes.TXT").readlines()]
boxes, scores, classes, nums = yolo(img)
count=0
for i in range(nums[0]):
if int(classes[0][i] == 0):
count +=1
if int(classes[0][i] == 67):
print('Mobile Phone detected')
if count == 0:
print('No person detected')
elif count > 1:
print('More than one person detected')

image = draw_outputs(image, (boxes, scores, classes, nums), class_names)

cv2.imshow('Prediction', image)
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
        break
cap.release()

cv2.destroyAllWindows()
```

6.4 AUDIO PROCESSING

```
import speech_recognition as sr
import pyaudio
import wave
import time
import threading
import os
import nltk
def read_audio(stream, filename):
        chunk = 1024  # Record in chunks of 1024 samples
        sample_format = pyaudio.paInt16  # 16 bits per sample
        channels = 1
        fs = 44100  # Record at 44100 samples per second
        seconds = 10 # Number of seconds to record at once
        filename = filename
        frames = []  # Initialize array to store frames
```

```
        for i in range(0, int(fs / chunk * seconds)):
        data = stream.read(chunk)
        frames.append(data)

        # Save the recorded data as a WAV file
        wf = wave.open(filename, 'wb')
        wf.setnchannels(channels)
        wf.setsampwidth(p.get_sample_size(sample_format))
        wf.setframerate(fs)
        wf.writeframes(b''.join(frames))
        wf.close()
        # Stop and close the stream
        #stream.stop_stream()
        stream.close()

def convert(i):
        if i >= 0:
        sound = 'record' + str(i) +'.wav'
        r = sr.Recognizer()

        with sr.AudioFile(sound) as source:
```

```
r.adjust_for_ambient_noise(source)

print("Converting Audio To Text and saving to file..... ")

audio = r.listen(source)

try:

value = r.recognize_google(audio) ##### API call to google for speech recognition

os.remove(sound)

if str is bytes:

result = u"{}".format(value).encode("utf-8")

else:

result = "{}".format(value)


with open("test.txt","a") as f:

f.write(result)

f.write("")

f.close()


except sr.UnknownValueError:

print("")

except sr.RequestError as e:

print("{0}".format(e))

except KeyboardInterrupt:
```

```
        pass


p = pyaudio.PyAudio()  # Create an interface to PortAudio
chunk = 1024  # Record in chunks of 1024 samples
sample_format = pyaudio.paInt16  # 16 bits per sample
channels = 1
fs = 44100


def save_audios(i):
        stream = p.open(format=sample_format,channels=channels,rate=fs,
        frames_per_buffer=chunk,input=True)
        filename = 'record'+str(i)+'.wav'
        read_audio(stream, filename)



'''for i in range(30//10): # Number of total seconds to record/ Number of seconds per recording
        t1 = threading.Thread(target=save_audios, args=[i])
        x = i-1
        t2 = threading.Thread(target=convert, args=[x]) # send one earlier than being recorded
        t1.start()
        t1.join()
```

```
        t2.start()
        t2.join()
        save_audios(i)
        convert(x)
        if i==2:
        flag = True
if flag:
        convert(i)
        p.terminate()'''
for i in range(30//10):
    save_audios(i)
    x = i-1
    convert(x) # send one earlier than being recorded


    if i==2:
         flag = True
if flag:
        convert(i)
        #p.terminate()



from nltk.corpus import stopwords
```

```
from nltk.tokenize import word_tokenize
nltk.download('stopwords')
nltk.download('punkt')

file = open("test.txt") ## Student speech file
data = file.read()
file.close()
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(data) ######### tokenizing sentence
filtered_sentence = [w for w in word_tokens if not w in stop_words]
filtered_sentence = []

for w in word_tokens:   ######## Removing stop words
        if w not in stop_words:
        filtered_sentence.append(w)

######## creating a final file
f=open('final.txt','w')
for ele in filtered_sentence:
        f.write(ele+' ')
f.close()
```

```
##### checking whether proctor needs to be alerted or not
file = open("Paper.txt") ## Question file
data = file.read()
file.close()
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(data) ######### tokenizing sentence
filtered_questions = [w for w in word_tokens if not w in stop_words]
filtered_questions = []

for w in word_tokens:   ######## Removing stop words
        if w not in stop_words:
        filtered_questions.append(w)

def common_member(a, b):
        a_set = set(a)
        b_set = set(b)

        # check length
        if len(a_set.intersection(b_set)) > 0:
        return(a_set.intersection(b_set))
        else:
        return([])
```

```
comm = common_member(filtered_questions, filtered_sentence)
print('Number of common elements:', len(comm))
print(comm)
```

6.5 USER INTERFACE

```
import React from 'react';
import './App.css';
import ExamState from './components/context/ExamState';
import { BrowserRouter as Router, Switch, Route } from "react-router-dom";
import Dashboard from './components/Dashboard';
import Exams from './components/Exams';
import Profile from './components/Profile';
import CreateQues from './components/CreateQues';
import ScheduleExam from './components/ScheduleExam';
import BoardingPage from './components/BoardingPage';
import LoginPage from './components/LoginPage';
import SignupPage from './components/SignupPage';
import About from './components/About';
import Contact from './components/Contact';
import AttendExam from './components/AttendExam';
```

```
import AttendExam2 from './components/AttendExam2';
import NotFound from './components/NotFound';
import Result from './components/Result';

const App = () => {
  return (
        <ExamState>
        <Router>
        <div className="App">
        <Switch>
        <Route exact path="/" component={BoardingPage} />
        <Route exact path="/about" component={About}/>
        <Route exact path="/contact" component={Contact}/>
        <Route exact path="/login" component={LoginPage} />
        <Route exact path="/signup" component={SignupPage} />
        <Route exact path="/dashboard" component={Dashboard}/>
        <Route exact path="/exams" component={Exams} />
        <Route exact path="/profile" component={Profile} />
        <Route exact path="/createExam" component={ScheduleExam}/>
        <Route exact path="/QuesBank" component={CreateQues}/>
        <Route exact path="/attendExam" component={AttendExam}/>
        <Route exact path="/attendExam2" component={AttendExam2}/>
```

```
        <Route exact path="/result" component={Result}/>
        <Route component={NotFound}/>
        </Switch>
        </div>
        </Router>
        </ExamState>
  );
}
export default App;
```

CHAPTER 7

TESTING

7. TESTING

7.1 INTRODUCTION

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consisting of several key activities and steps for running a program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

Test Case No.	Action	Expected Output	Actual Output	Result
1	SignUp using G-mail Credentials	Validating the G-Mail Credentials and Proceeding to Login Page	Validating the G-Mail Credentials and Proceeding to Login Page	Pass
2	Login using G-Mail Credentials	Proceeding to Dash Board	Proceeding to Dash Board	Pass
3	Dash Board	Displaying Scheduled Exams	Displaying Scheduled Exams	Pass
4	Attend Exam	Proceed Attending the Exam	Proceed Attending the Exam	Pass

5	Exam Schedule (For Faculty)	After entering with faculty credential create question paper	After entering with faculty credential create question paper	Pass
6	View Exam Result	Display Student Exam Result	Display Student Exam Result	Pass
Test Case No.	**Action**	**Expected Output**	**Actual Output**	**Result**
7	User Verification	Authenticate the User via Face Recognition	Authenticate the User via Face Recognition	Pass
8	Person Detection	Monitoring the Candidate for Presence and Updating to the faculty if not present	Monitoring the Candidate for Presence and Updating to the faculty if not present	Pass
9	Multiple Person Detection	Checking for the presence of Multiple Persons and altering the Faculty	Checking for the presence of Multiple Persons and altering the Faculty	Pass
10	Object Detection	Altering the Faculty for the presence of any other objects via text alter	Altering the Faculty for the presence of any other objects via text alter	Pass
11	Gaze Detection	Monitoring the Persons Gaze	Monitoring the Persons Gaze	Pass
12	Audio Processing	Altering the Faculty with the Keywords present in the Question Paper via Text	Altering the Faculty with the Keywords present in the Question Paper via Text	Pass

7.2 TYPES OF TESTS

7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying. Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.2.5 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 WHITE BOX TESTING

This testing is also called Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

7.4 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as a black box .You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

CHAPTER 8

CONCLUSION & FUTURE ENHANCEMENT

8. CONCLUSION & FUTURE ENHANCEMENT

8.1 CONCLUSION

Thus, the aim is to provide a safe and user-friendly environment to take exams from a remote location with a better machine learning model to proctor the examination and to prevent students from malpractices. By combining the continuous estimation components, and applying a temporal sliding window, we design higher-level features to classify whether the test taker is cheating at any moment during the exam.

8.2 FUTURE ENHANCEMENT:

- Working in non-ideal conditions:
  Any non-ideal conditions such as low quality web cameras or microphones being used can be negated by using advanced processing algorithms.

- Bad Network Connections:
  If the candidates' network quality is poor, any method to employ offline assessments has to be automatically run so that the transition is smooth so that the assessee does not face any hindrances.

- Ui/Ux Enhancements:
  Enhancements can be made to the front-end UI so that the candidates can have a more pleasant experience.

- Scalability:
  Further advancements to accommodate even more numbers of candidates to assess simultaneously can be implemented.

8.3 APPLICATIONS:

1. E-Learning:

   eLearning, or electronic learning, is the delivery of learning and training through digital resources. Although eLearning is based on formalized learning, it is provided through electronic devices such as computers, tablets and even cellular phones that are connected to the internet. This makes it easy for users to learn anytime, anywhere, with few, if any, restrictions. Basically, eLearning is training, learning, or education delivered online through a computer or any other digital device.

2. Distance Education:

   Distance education, also called distance learning, is the education of students who may not always be physically present at a school. Traditionally, this usually involved correspondence courses wherein the student corresponded with the school via mail. Today, it involves online education.

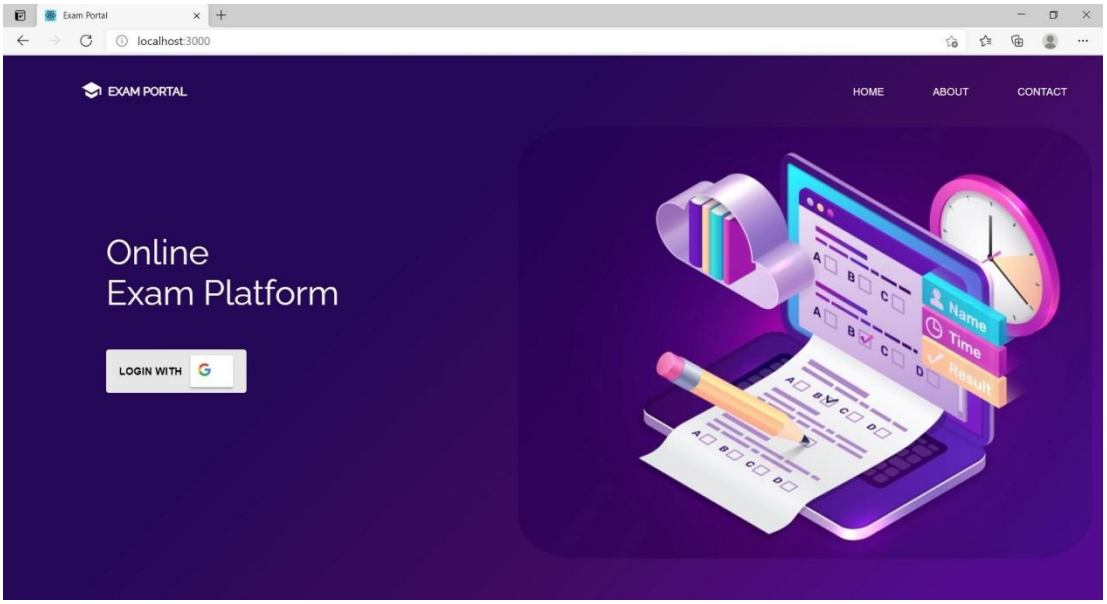
3. Assessment:

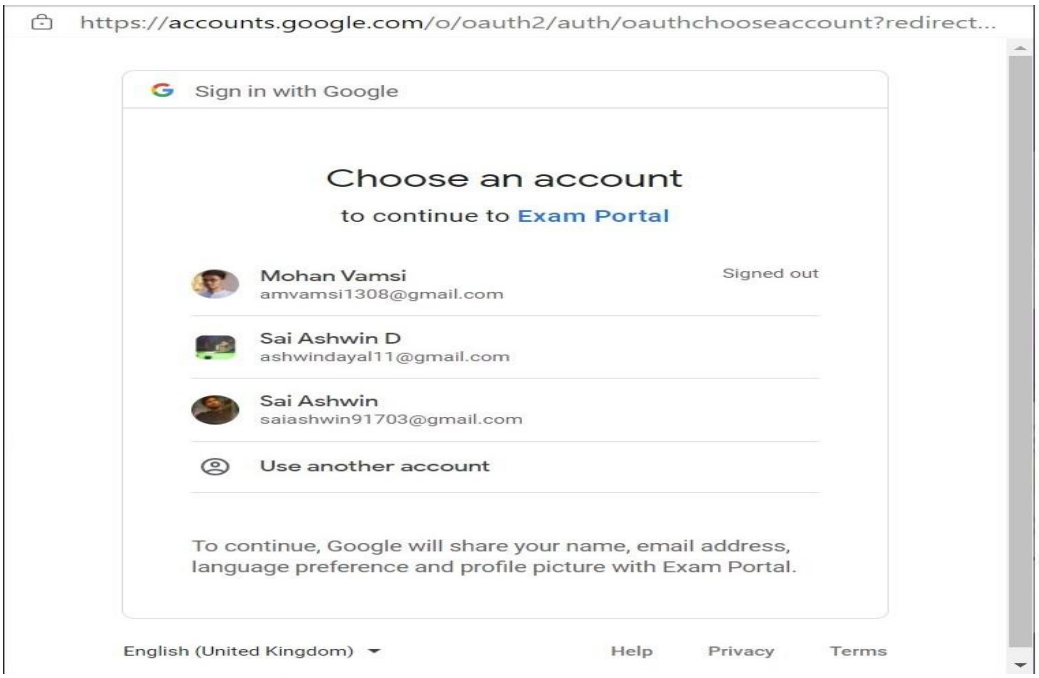
   Educational assessment is the systematic process of documenting and using empirical data on the knowledge, skill, attitudes, and beliefs to refine programs and improve student learning. Assessment data can be obtained from directly examining student work to assess the achievement of learning outcomes or can be based on data from which one can make inferences about learning. Assessment can focus on the individual learner, the learning community, a course, an academic program, the institution, or the educational system as a whole.

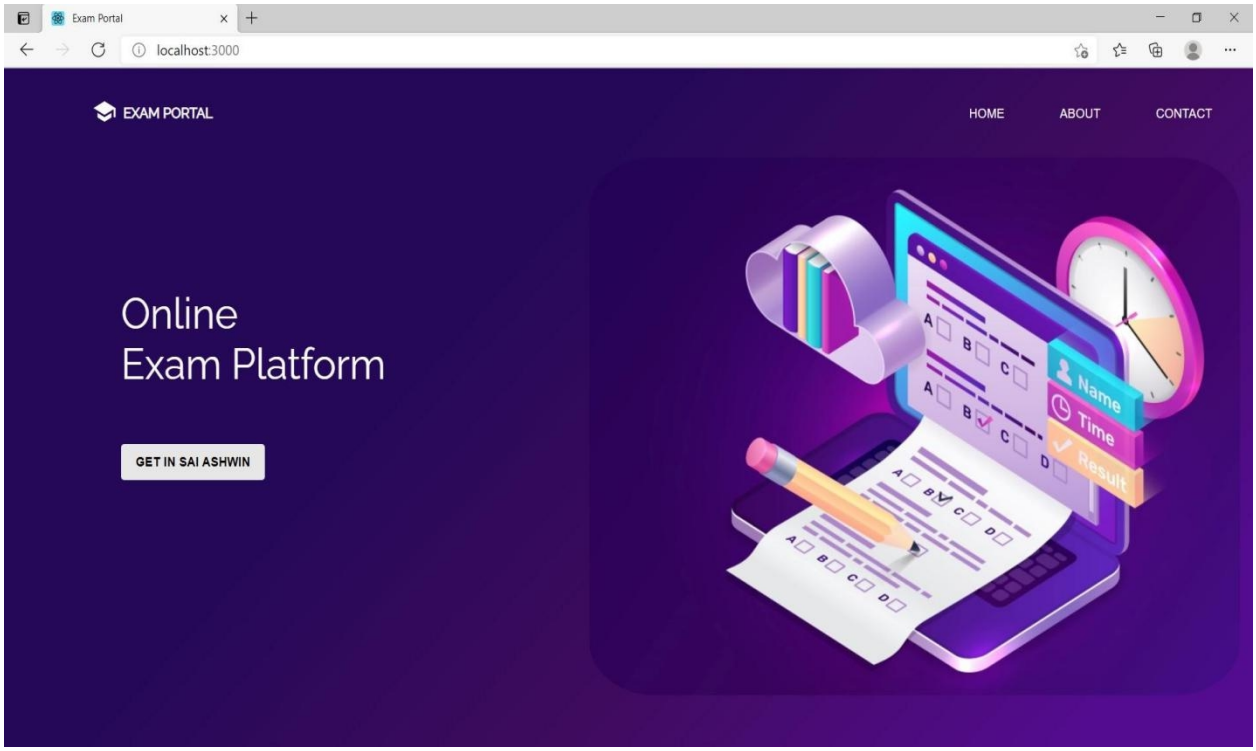
4. It uses real time remote proctoringmethods to effectively conduct examinations for candidates.

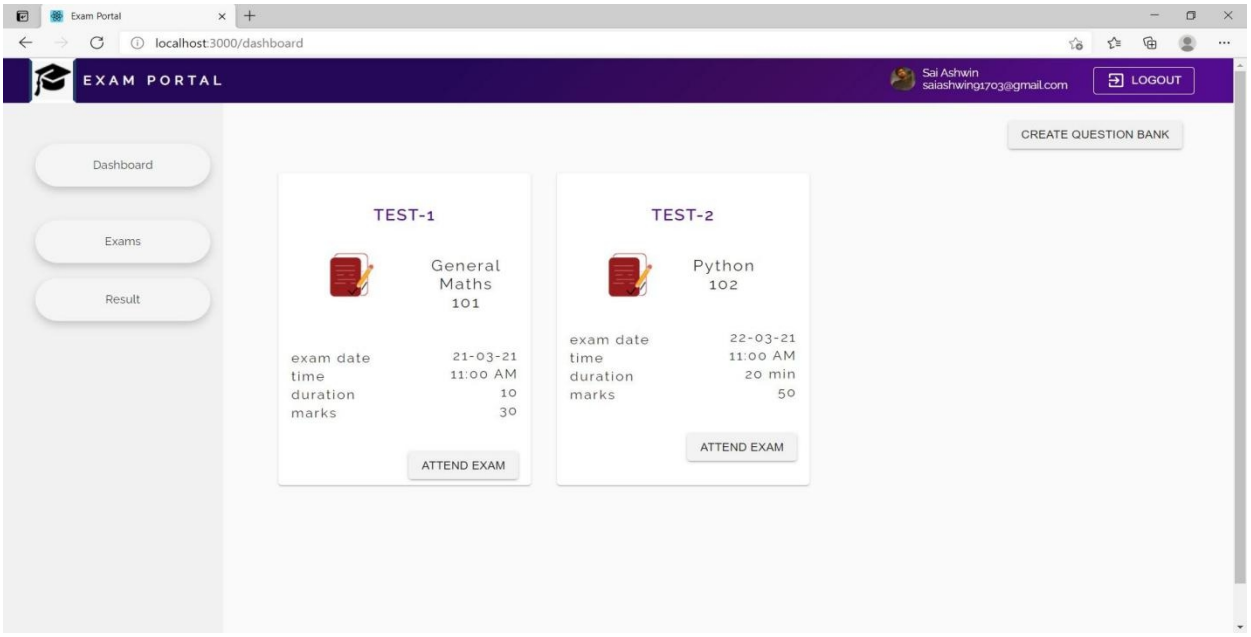
APPENDICES

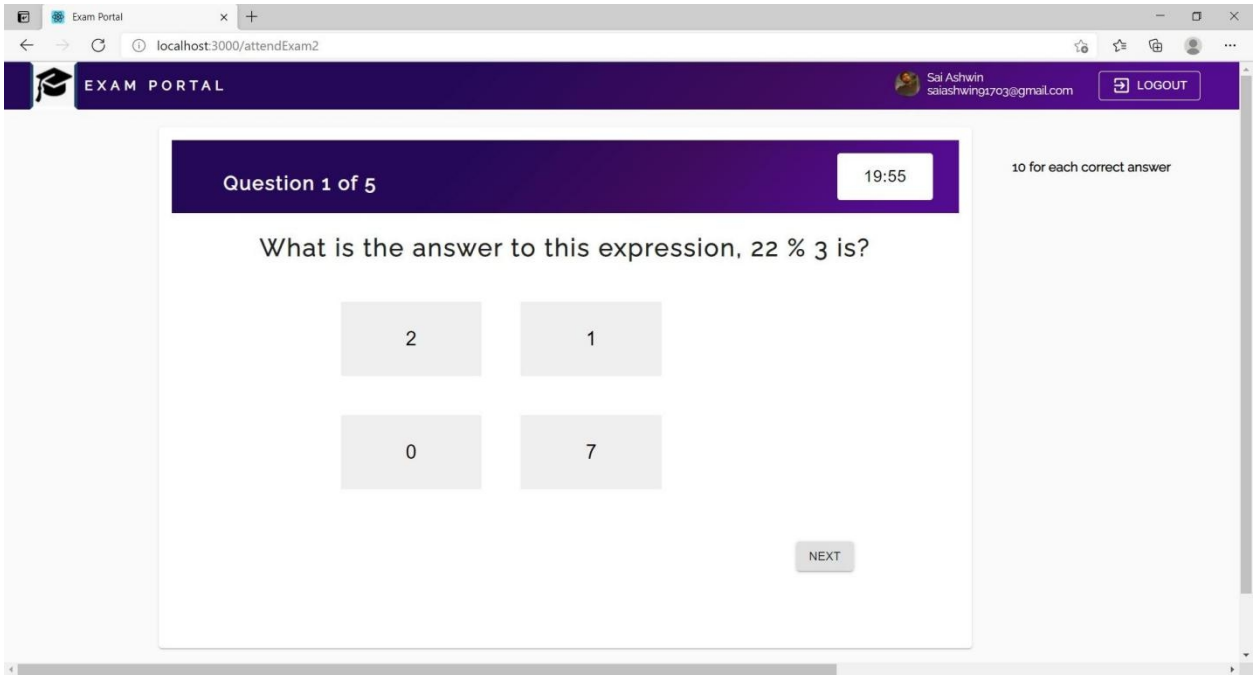
A.1 SAMPLE SCREENS

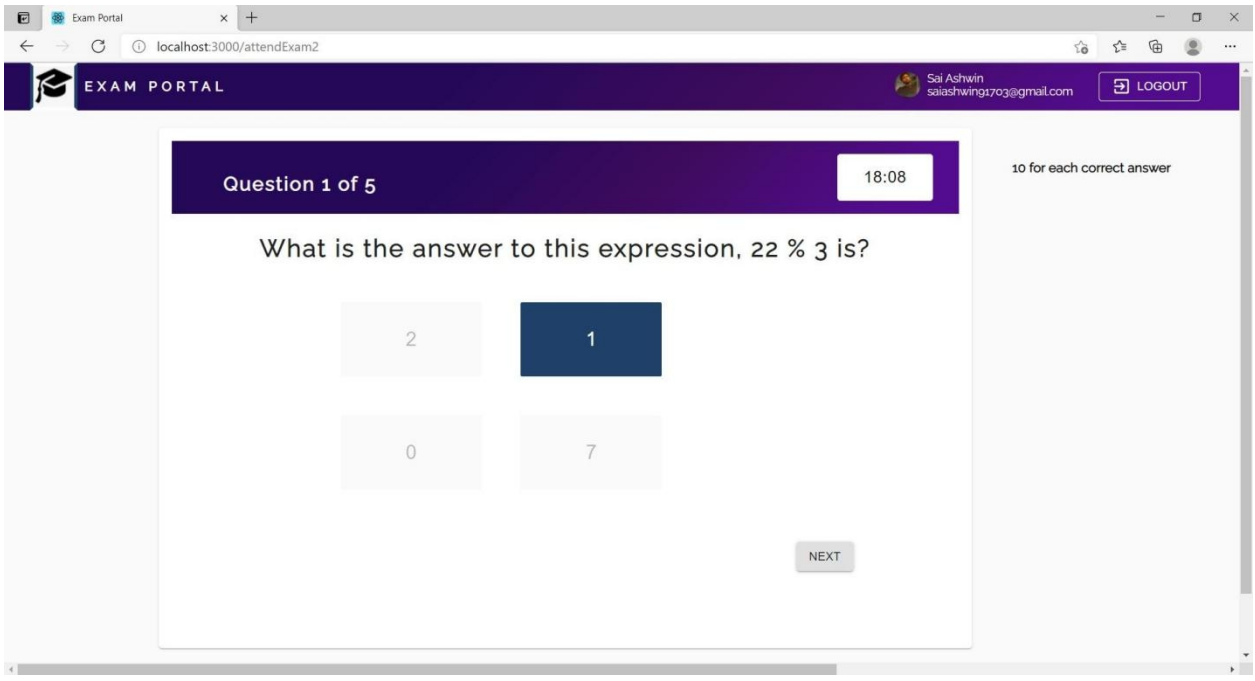


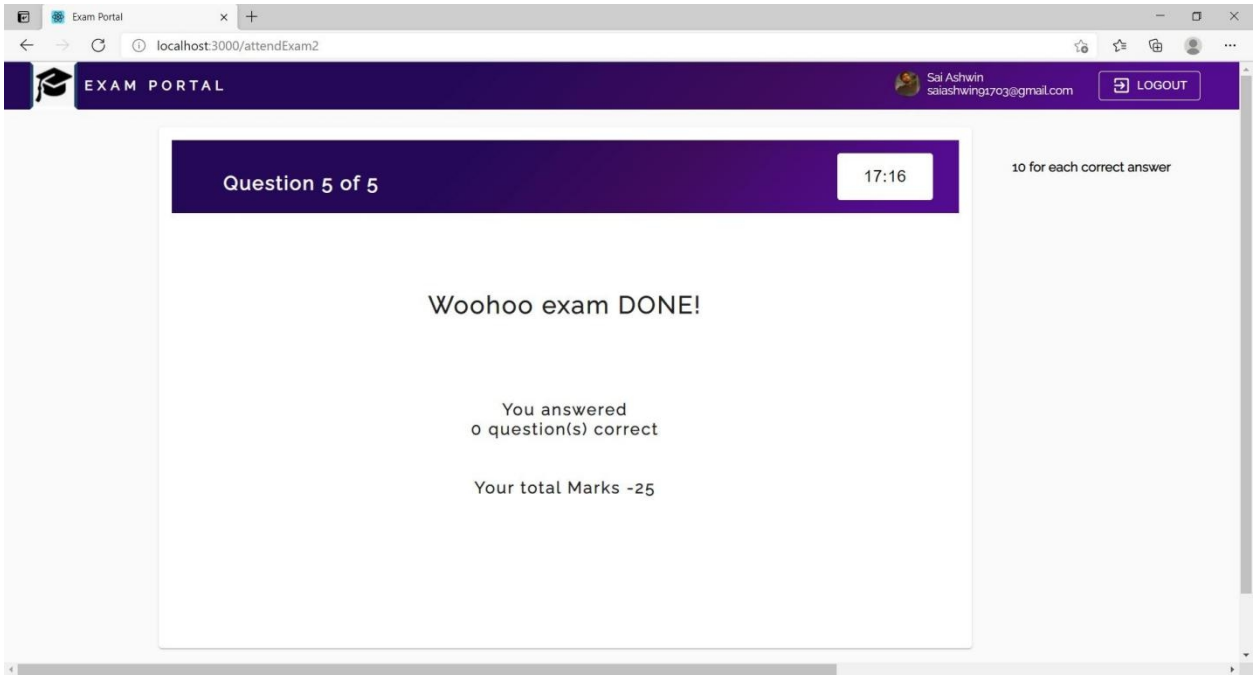


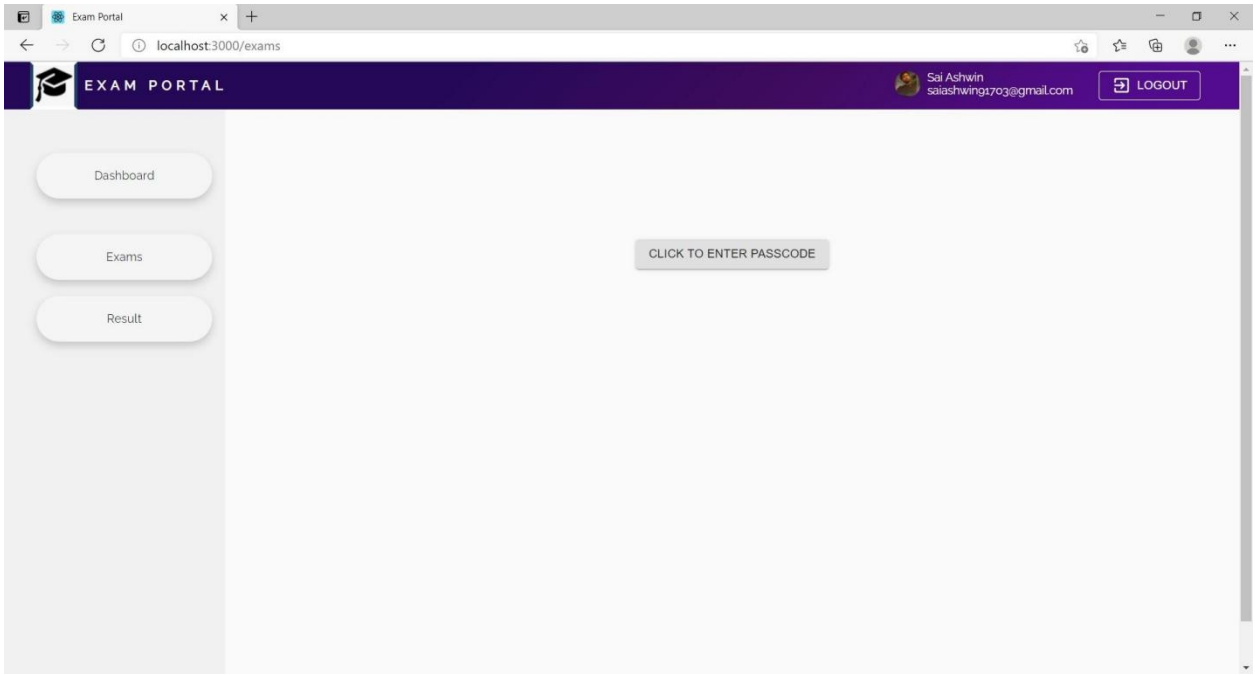

Exam Portal
localhost:3000
EXAM PORTAL
HOME
ABOUT
CONTACT
Online
Exam Platform
GET IN SAI ASHWIN
Name
Time
Result

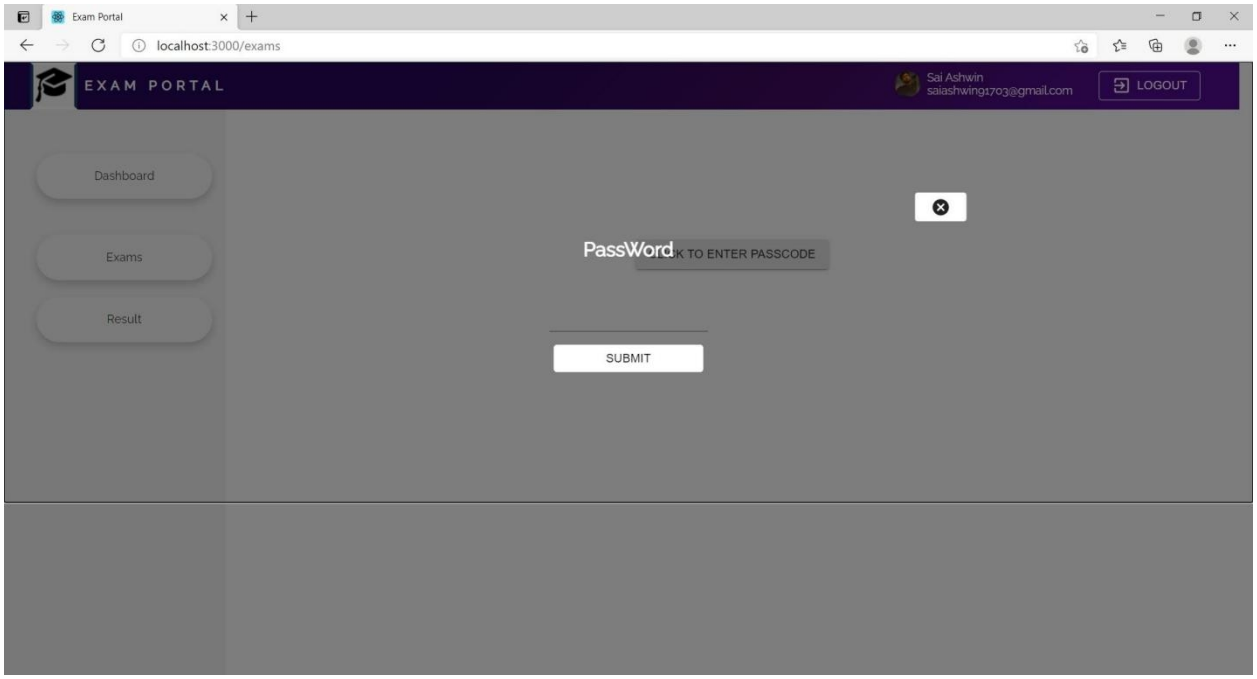

Exam Portal
localhost:3000/dashboard
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
CREATE QUESTION BANK
Dashboard
Exams
Result
TEST-1
General Maths 101
exam date 21-03-21
time 11:00 AM
duration 10
marks 30
ATTEND EXAM
TEST-2
Python 102
exam date 22-03-21
time 11:00 AM
duration 20 min
marks 50
ATTEND EXAM

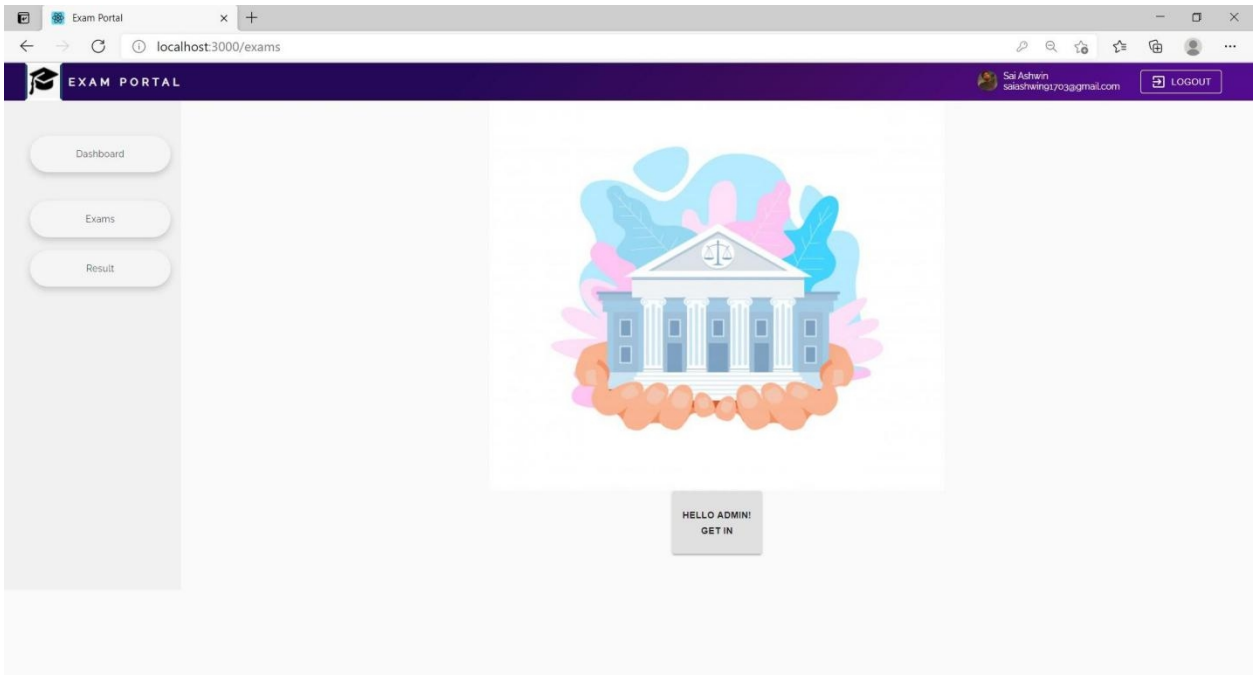

Exam Portal
localhost:3000/attendExam2
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Question 1 of 5
19:55
10 for each correct answer
What is the answer to this expression, 22 % 3 is?
2
1
0
7
NEXT

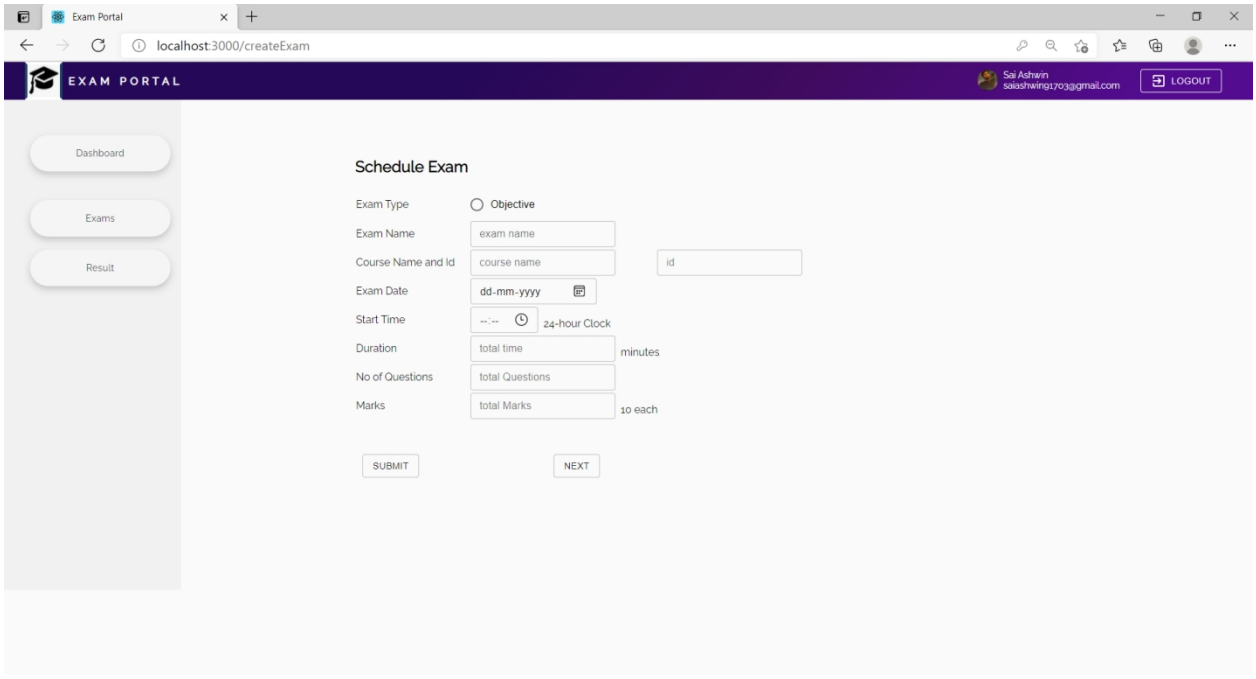

Exam Portal
localhost:3000/attendExam2
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Question 1 of 5
18:08
10 for each correct answer
What is the answer to this expression, 22 % 3 is?
2
1
0
7
NEXT

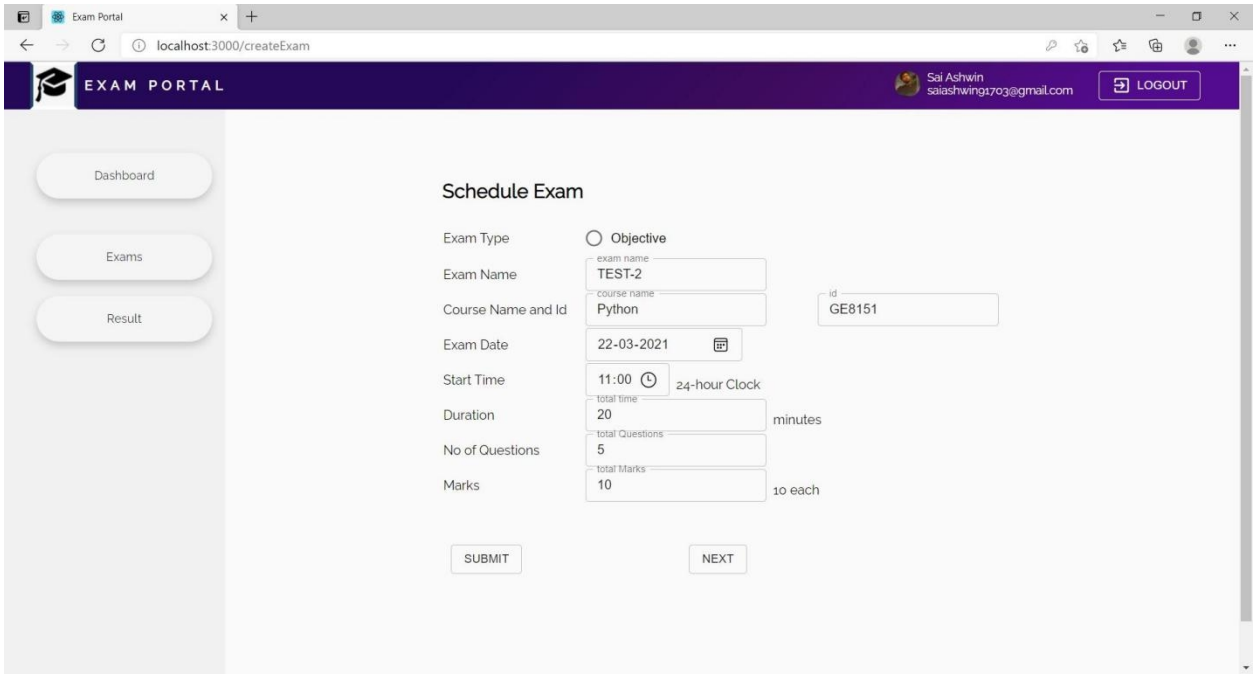

Exam Portal
localhost:3000/attendExam2
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Question 5 of 5
17:16
10 for each correct answer
Woohoo exam DONE!
You answered
0 question(s) correct
Your total Marks -25

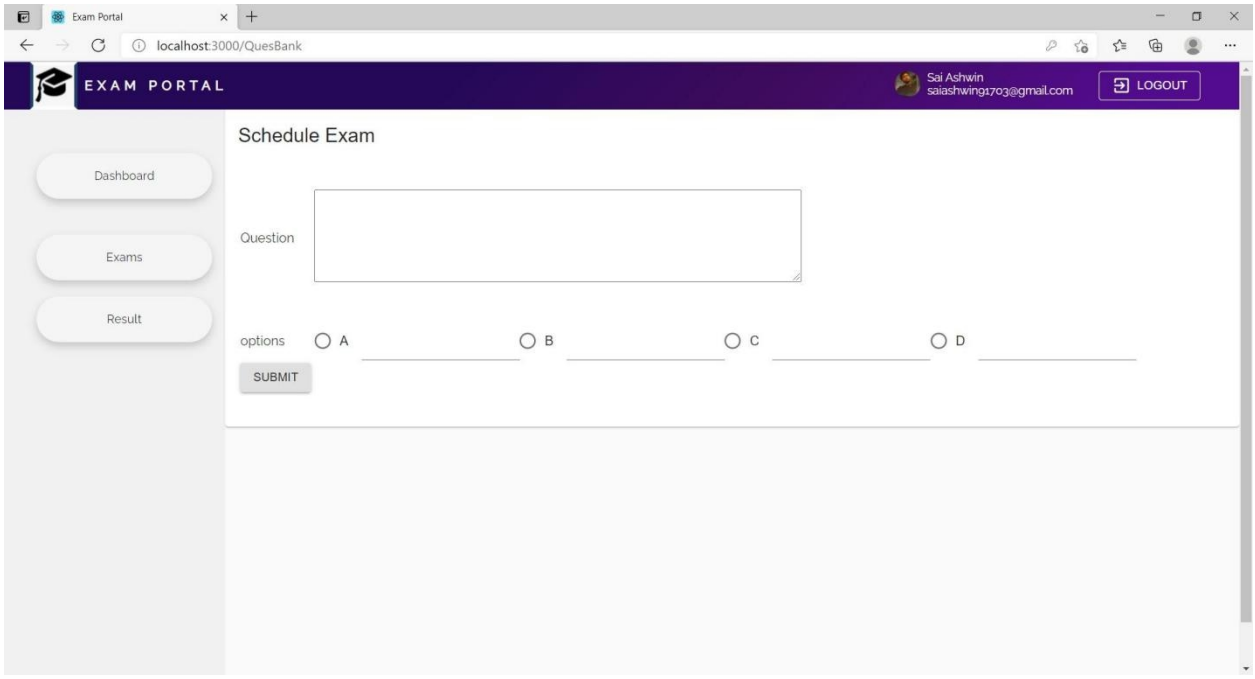

Exam Portal
localhost:3000/exams
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Dashboard
Exams
Result
CLICK TO ENTER PASSCODE

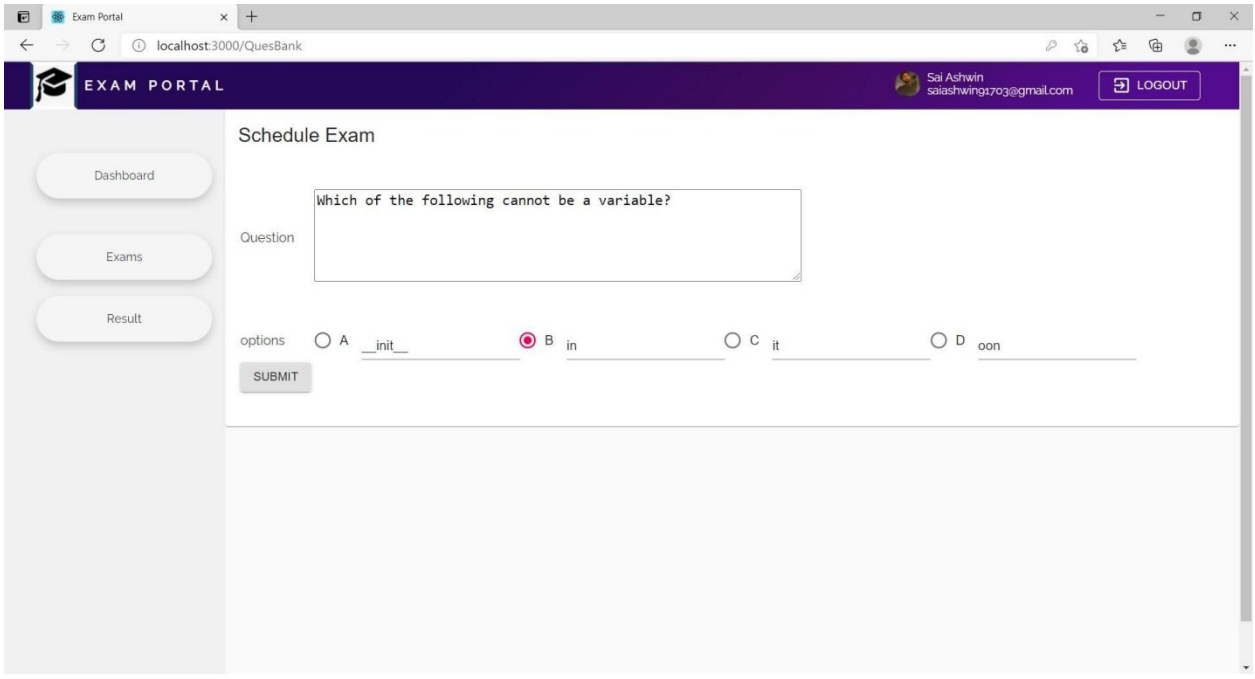

Exam Portal
localhost:3000/exams
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Dashboard
Exams
Result
PassWord
K TO ENTER PASSCODE
SUBMIT

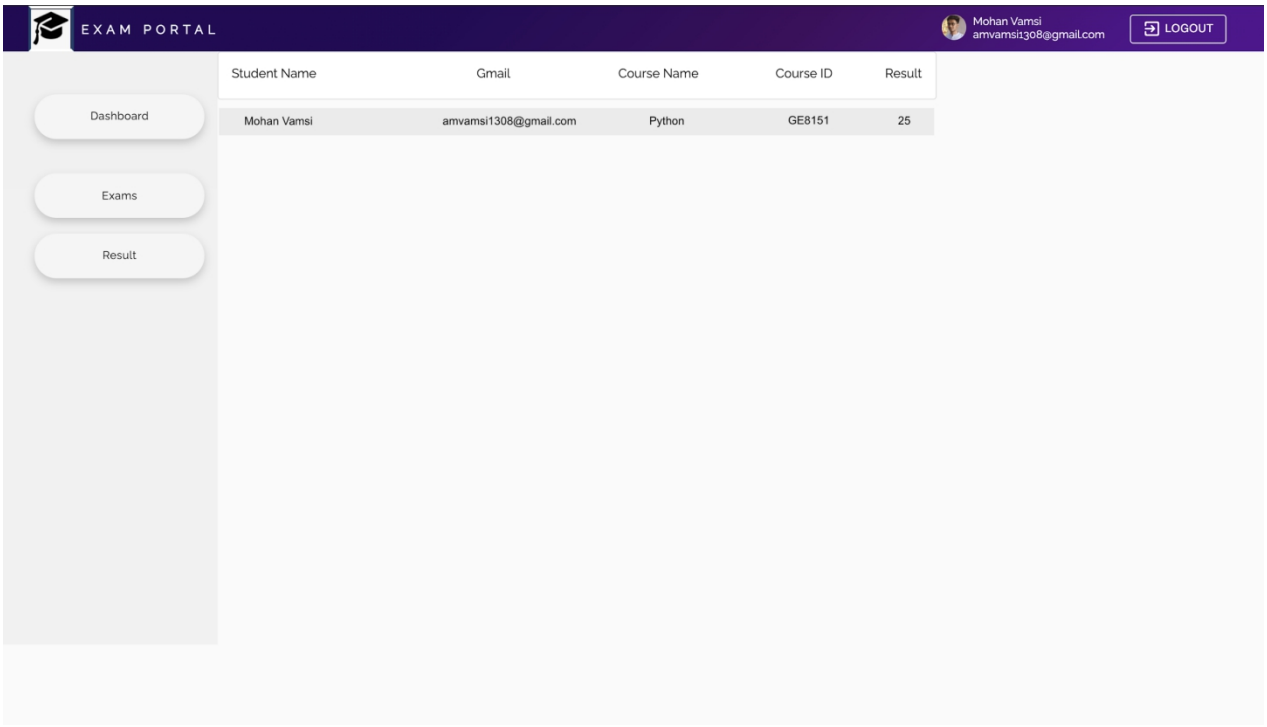

Exam Portal
localhost:3000/exams
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Dashboard
Exams
Result
HELLO ADMIN!
GET IN


Exam Portal
localhost:3000/createExam
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Dashboard
Exams
Result
Schedule Exam
Exam Type
Objective
Exam Name
exam name
Course Name and Id
course name
id
Exam Date
dd-mm-yyyy
Start Time
24-hour Clock
Duration
total time
minutes
No of Questions
total Questions
Marks
total Marks
10 each
SUBMIT
NEXT

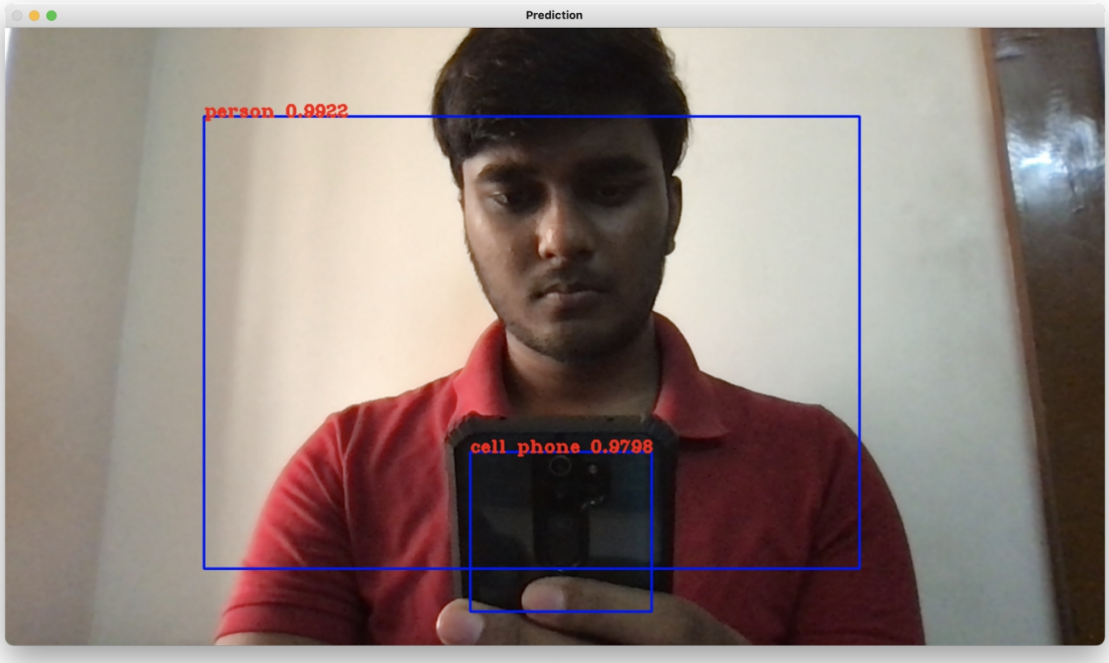

Exam Portal
localhost:3000/createExam
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Dashboard
Exams
Result
Schedule Exam
Exam Type
Objective
Exam Name
exam name
TEST-2
Course Name and Id
course name
Python
id
GE8151
Exam Date
22-03-2021
Start Time
11:00
24-hour Clock
Duration
total time
20
minutes
No of Questions
total Questions
5
Marks
total Marks
10
10 each
SUBMIT
NEXT

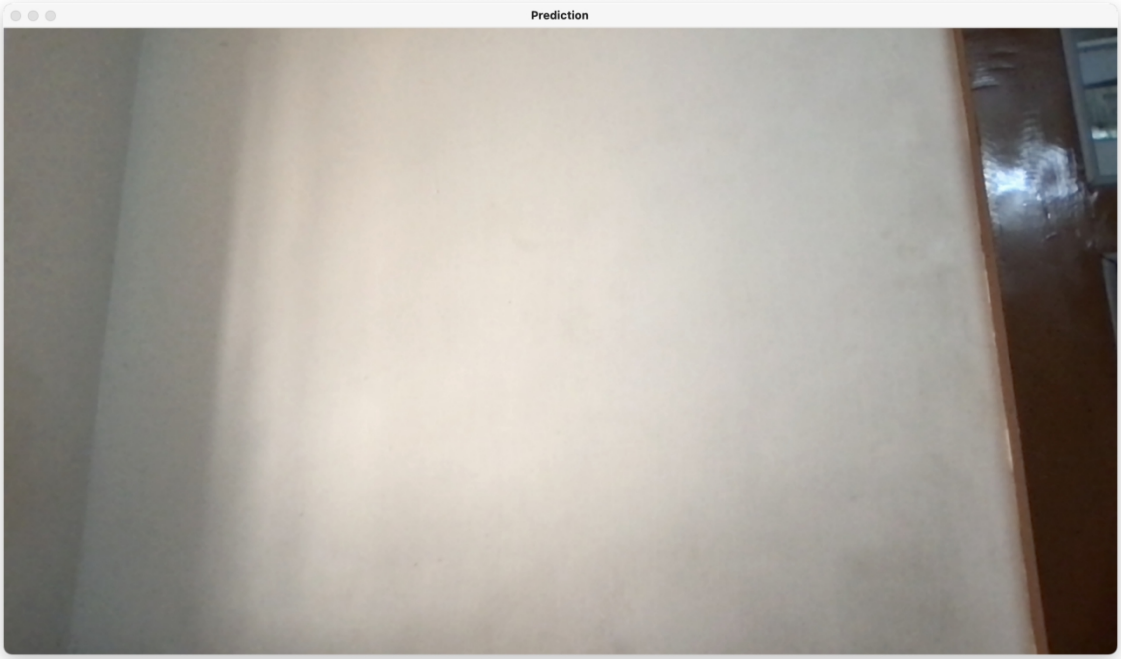

Exam Portal
localhost:3000/QuesBank
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Schedule Exam
Dashboard
Exams
Result
Question
options
A
B
C
D
SUBMIT

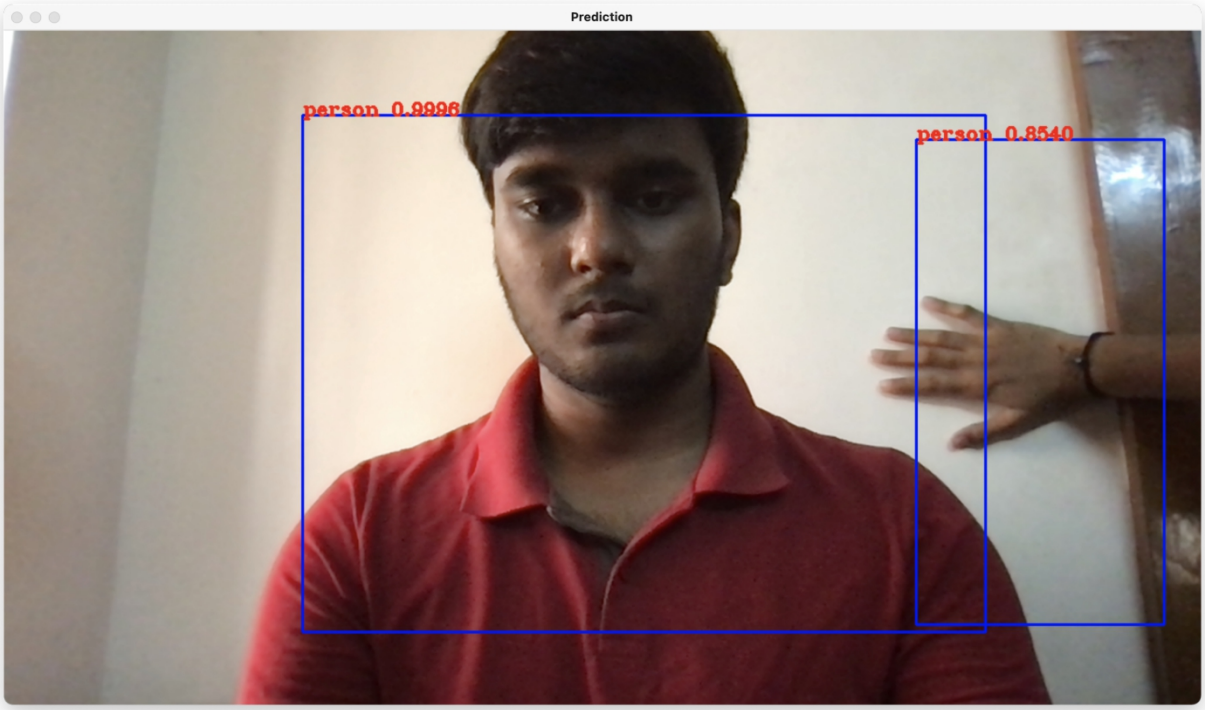

Exam Portal
localhost:3000/QuesBank
EXAM PORTAL
Sai Ashwin
saiashwing1703@gmail.com
LOGOUT
Schedule Exam
Dashboard
Exams
Result
Question
Which of the following cannot be a variable?
options
A __init__
B in
C it
D oon
SUBMIT


EXAM PORTAL
Mohan Vamsi
amvamsit308@gmail.com
LOGOUT
Dashboard
Exams
Result
Student Name
Gmail
Course Name
Course ID
Result
Mohan Vamsi
amvamsi1308@gmail.com
Python
GE8151
25

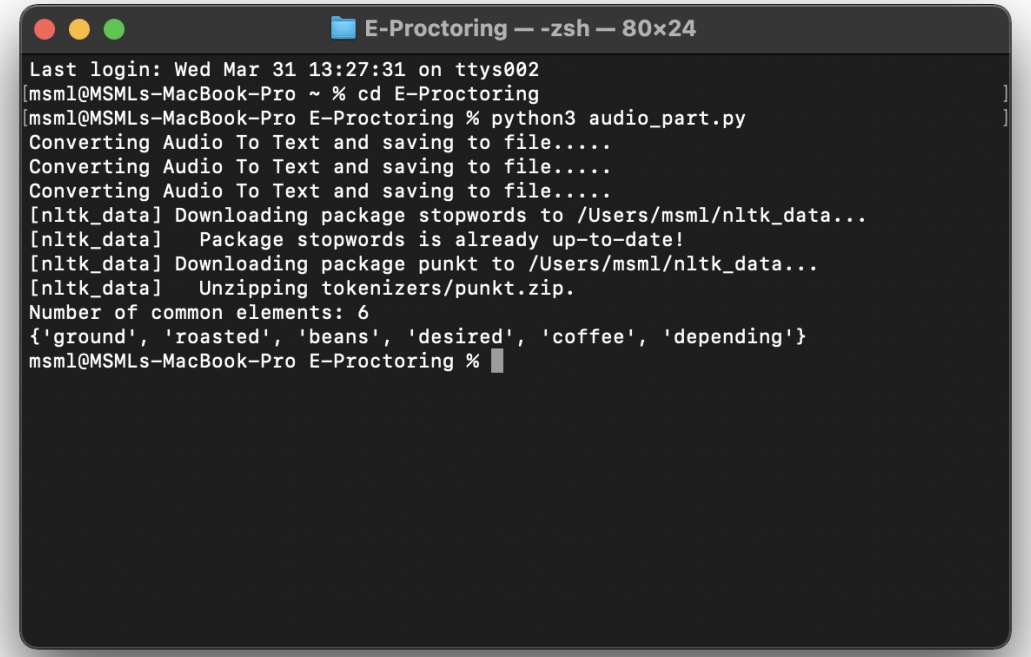

Video
Mohan Vamsi


Prediction
person 0.9922
cell phone 0.9798


Prediction




```
E-Proctoring — Python person_and_phone.py — 80×24
creating XLA devices, tf_xla_enable_xla_devices not set
2021-03-31 13:23:43.214330: I tensorflow/core/platform/cpu_feature_guard.cc:142]
 This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (on
eDNN) to use the following CPU instructions in performance-critical operations:
 AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate comp
iler flags.
Mobile Phone detected
Mobile Phone detected
Mobile Phone detected
Mobile Phone detected
Mobile Phone detected
No person detected
No person detected
No person detected
No person detected
No person detected
No person detected
More than one person detected
More than one person detected
More than one person detected
More than one person detected
More than one person detected
```


E-Proctoring — -zsh — 80×24
Last login: Wed Mar 31 13:27:31 on ttys002
msml@MSMLs-MacBook-Pro ~ % cd E-Proctoring
msml@MSMLs-MacBook-Pro E-Proctoring % python3 audio_part.py
Converting Audio To Text and saving to file.....
Converting Audio To Text and saving to file.....
Converting Audio To Text and saving to file.....
[nltk_data] Downloading package stopwords to /Users/msml/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /Users/msml/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
Number of common elements: 6
{'ground', 'roasted', 'beans', 'desired', 'coffee', 'depending'}
msml@MSMLs-MacBook-Pro E-Proctoring %

A.2 PAPER PUBLICATION

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ref No : IJCRT/Vol 9 / Issue 5/ 041

To,
Mohan Vamsi A

Subject: Publication of paper at International Journal of Creative Research Thoughts.

Dear Author,

With Greetings we are informing you that your paper has been successfully published in the International Journal of Creative Research Thoughts - IJCRT (ISSN: 2320-2882). Thank you very much for your patience and cooperation during the submission of paper to final publication Process. It gives me immense pleasure to send the certificate of publication in our Journal. Following are the details regarding the published paper.

About IJCRT	:	Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly, Indexing in all major database & Metadata, Citation Generator, Digital Object Identifier(DOI)
Registration ID	:	IJCRT_208001
Paper ID	:	IJCRT2106041
Title of Paper	:	REMOTE ONLINE PROCTORING SYSTEM
Impact Factor	:	7.97 (Calculate by Google Scholar) | License by Creative Common 3.0
Publication Date	:	2021-05-29 10:23:56
DOI	:	
Published in	:	Volume 9 | Issue 5 | May 2021
Page No	:	j559-j565
Published URL	:	http://www.ijcrt.org/viewfull.php?&p_id=IJCRT2106041
Authors	:	Mohan Vamsi A, Niteesh B, Sai Ashwin D, K Kajendran
Notification	:	

Thank you very much for publishing your article in IJCRT.

Editor In Chief
International Journal of Creative Research Thoughts - IJCRT
(ISSN: 2320-2882)

Indexing Google scholar ISSN Academic ResearchGate Academia.edu RESEARCHERID MENDELEY Semantic Scholar
CiteSeer SSRN docstoc Google Scribd. OPEN ACCESS publons

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all major database & Metadata, Citation Generator

Website: www.ijcrt.org | Email: editor@ijcrt.org

REFERENCES

1. Atoum, Y., Chen, L., Liu, . X., Hsu, S. D. H., & Liu, X. (2017). Automated Online Exam Proctoring. IEEE Transactions on Multimedia, 19(7), 1609–1624. doi:10.1109/tmm.2017.2656064

2. Asep, H. S. G., & Bandung, Y. (2019). A Design of Continuous User Verification for Online Exam Proctoring on M-Learning. 2019 International Conference on Electrical Engineering and Informatics (ICEEI). doi:10.1109/iceei47359.2019.8988786

3. Prathish, S., Athi Narayanan S, & Bijlani, K. (2016). An intelligent system for online exam monitoring. 2016 International Conference on Information Science (ICIS). doi:10.1109/infosci.2016.7845315

4. Rosen, W. A., & Carr, M. E. (2013). An autonomous articulating desktop robot for proctoring remote online examinations. 2013 IEEE Frontiers in Education Conference (FIE). doi:10.1109/fie.2013.6685172

5. Reale, M. J., Canavan, S., Yin, L., Hu, K., & Hung, T. (2011). A Multi-Gesture Interaction System Using a 3-D Iris Disk Model for Gaze Estimation and an Active Appearance Model for 3-D Hand Pointing. IEEE Transactions on Multimedia, 13(3), 474–486. doi:10.1109/tmm.2011.2120600.

6. D. L. King and C. J. Case. E-cheating: Incidence and trends among college students. Issues in Information Systems, 15(1), 2014.

7. Debnath, Partha & Rashed, Md. Golam & Das, Dipankar. (2018). Detection and Controlling of Suspicious Behaviour in the Examination Hall.

8. N. L. Clarke, P. Dowland and S. M. Furnell, "e-Invigilator: A biometric-based supervision system for e-Assessments," International Conference on Information Society (i-Society 2013), Toronto, ON, Canada, 2013, pp. 238-242.

9. I. Y. Jung and H. Y. Yeom, "Enhanced Security for Online Exams Using Group Cryptography," in IEEE Transactions on Education, vol. 52, no. 3, pp. 340-349, Aug. 2009, doi: 10.1109/TE.2008.928909.

10. Asteriadis, S., Tzouveli, P., Karpouzis, K., & Kollias, S. (2008). Estimation of behavioral user state based on eye gaze and head pose—application in an e-learning environment. Multimedia Tools and Applications, 41(3), 469–493. doi:10.1007/s11042-008-0240-1

11. Y. Cheung and Q. Peng, "Eye Gaze Tracking With a Web Camera in a Desktop Environment," in IEEE Transactions on Human-Machine Systems, vol. 45, no. 4, pp. 419-430, Aug. 2015, doi: 10.1109/THMS.2015.2400442.

12. X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2879-2886, doi: 10.1109/CVPR.2012.6248014.
13. Savvides, Marios & Kumar, B. & Khosla, Pradeep. (2002). Face verification using correlation filters. 3rd IEEE Automatic Identification Advanced Technologies.

14. R. S. V. Raj, S. A. Narayanan and K. Bijlani, "Heuristic-Based Automatic Online Proctoring System," 2015 IEEE 15th International Conference on Advanced Learning Technologies, Hualien, Taiwan, 2015, pp. 458-459, doi: 10.1109/ICALT.2015.127.

15. Hasan, H., & Abdul-Kareem, S. (2013). RETRACTED ARTICLE: Human–computer interaction using vision-based hand gesture recognition systems: a survey. Neural Computing and Applications, 25(2), 251–261. doi:10.1007/s00521-013-1481-0

16. Tsukada, A., Shino, M., Devyver, M., & Kanade, T. (2011). Illumination-free gaze estimation method for first-person vision wearable device. 2011 IEEE

International Conference on Computer Vision Workshops (ICCV Workshops). doi:10.1109/iccvw.2011.6130505.

17. González-González, C. S., Infante-Moro, A., & Infante-Moro, J. C. (2020). Implementation of E-proctoring in Online Teaching: A Study About Motivational Factors. Sustainability, 12(8), 3488. doi:10.3390/su12083488

18. Liu, X. (2010). Video-based face model fitting using Adaptive Active Appearance Model. Image and Vision Computing, 28(7), 1162–1172. doi:10.1016/j.imavis.2009.09.016

19. Wahid, A., Sengoku, Y., & Mambo, M. (2015). Toward constructing a secure online examination system. Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication - IMCOM '15. doi:10.1145/2701126.2701203

20. Jr, Cluskey, & Ehlen, Craig & Raiborn, Mitchell. (2011). Thwarting online exam cheating without proctor supervision. Journal of Academic and Business Ethics. 4.

21. NerkarM., P., A. T. Awaghade, D. A. Bombe and T. R. Deshmukh. "Online Exam Proctoring System-IJAERD." (2017).

22. Chua, S. S., Bondad, J. B., Lumapas, Z. R., & Garcia, J. D. (2019). Online Examination System with Cheating Prevention Using Question Bank Randomization and Tab Locking. 2019 4th International Conference on Information Technology (InCIT). doi:10.1109/incit.2019.8912065

23. Ping Guo, Hai-feng yu, & qian yao. (2008). The research and application of online examination and monitoring system. 2008 IEEE International Symposiumon IT in Medicine and Education. doi:10.1109/itme.2008.4743914.