**A PROJECT REPORT ON**

**PROCTO AI: An** **Online Proctoring System**.

**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**DHRUV SHARMA**  **2016732**

**SARTHAK SAHAI** **2017000**

**SATTRAJIT BASU** **2017002**

**UJJWAL KUMAR** **2017102**

***Under the Guidance of***

**Mr. RISHI KUMAR**

**(Assistant Professor)**

**Project Team ID: MP23CSE067**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**June-2024**



**CANDIDATE’S DECLARATION**

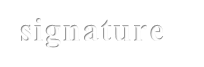
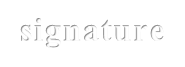
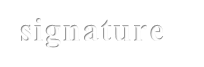
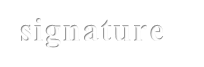
We hereby certify that the work is being presented in the Project Report entitled **“PROCTO AI”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun is an authentic record of my work carried out during a period from **August-2023 to May-2024** under the supervision of **Mr. Rishi Kumar, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University).

The matter presented in this dissertation has not been submitted by me/us for the award of any other degree of this or any other Institute/University.

Dhruv Sharma 2016732 **signature**

Sarthak Sahai 2017000 **signature**

Sattrajit Basu 2017002 **signature**



Ujjwal Kumar 2017102 **signature**

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

**Signature** **Signature**

**Supervisor** **Head of the Department**

**External Viva**

**Name of the Examiners: Signature with Date**

1.

2.

**Abstract**

The project aims to provide a safe and user-friendly environment to take exams online in 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 eLearning is maintained. The system is affordable and convenient to use from the test taker’s perspective since it only requires 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.

**Keywords:** Remote Monitoring, Identity Verification, Cheating Prevention, Automated Proctoring

**Acknowledgement**

Any achievement, be it scholastic or otherwise does not depend solely on individual effort but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. Several personalities in their capacity have helped me in carrying out this project work.

Our sincere thanks to project guide **Mr. RISHI KUMAR**

**, Assistant Professor,** Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable guidance and support throughout the course of project work and for being a constant source of inspiration.

We extend our thanks to **Prof. (Dr.) Guru Prasad M.S.**, Project Coordinator, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), for his valuable suggestions throughout all the phases of the Project Work.

We are extremely grateful to **Prof. (Dr.) D. P. Singh**, HOD of the Computer Science and Engineering Department, Graphic Era (Deemed to be University), for his moral support and encouragement.

We thank the **management of Graphic Era (Deemed to be University)** for the support throughout the course of our bachelor’s degree and for all the facilities they have provided.

Last, but certainly not least we thank all teaching and non-teaching staff of Graphic Era (Deemed to be University) for guiding us on the right path. Most importantly we wish to thank our parents for their support and encouragement.

Dhruv Sharma 2016732

Sarthak Sahai 2017000

Sattrajit Basu 2017002

Ujjwal Kumar 2017102

**Table of Contents**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **Contents** | **Page No.** | | Abstract | i | | Acknowledgement | ii | | Table of Contents | iii | | List of Tables | vi | | List of Figures | ix | | **Chapter 1 Introduction** | **1-2** | | 1.1 Project Introduction | 1 | | 1.2 Problem Statement | 2 | | 1.3 Objectives | 2 | | **Chapter 2**  **Literature Survey/ Background** | **3-6** | | **Chapter 3 Software Design** | **7-11** | | **Chapter 4**  **Requirements and Methodology** | **12-16** | | 4.1 Requirements |  | | 4.1.1 Hardware Requirements |  | | 4.1.2 Software Requirements |  | | 4.2 Methodology |  | | **Chapter 5**  **Coding /Code Templates** | **17-19** | | **Chapter 6**  **Testing** | **20-23** | | **Chapter 7** **Results and Discussion** | **24-29** | | **Chapter 8 Conclusion and Future Work** | **30** | | **Details of Research Publication** | **31** | | **References** | **32-34** | |  |
|  | **List of Tables**   |  |  |  | | --- | --- | --- | | **TABLE No.** | **TITLE** | **PAGE No.** | | 4.1 | Hardware Requirements | 13 | | 4.2 | Software Requirements | 13 | | 6.1 | Testing Test Cases | 20 | | 8.1 | Research Paper Details | 31 |   **List of Figures**   |  |  |  | | --- | --- | --- | | **FIGURE No.** | **TITLE** | **PAGE No.** | | 3.1 | Architecture Diagram of Remote E-Proctoring System | 7 | | 3.2.1 | Use Case Diagram of Remote E-Proctoring System | 8 | | 3.2.2 | ER Diagram of Remote E-Proctoring System | 9 | | 3.2.3 | Sequence Diagram of Remote E-Proctoring System | 10 | | 3.2.4.1 | Admin Data Flow Diagram of Remote E-Proctoring System | 11 | | 3.2.4.2 | User Data Flow Diagram of Remote E-Proctoring System | 11 | | 4.1 | Basic block diagram of E-Proctoring System | 12 | |  |

**Chapter 1**

**Introduction**

* 1. **Project Introduction**

The world of learning and evaluation has experienced a significant upheaval in an era characterized by quick technological breakthroughs and the ongoing global trend towards digital education. The rise in popularity of online testing and remote learning is a significant effect of this transition. The appropriate proctoring and monitoring of online exams to maintain academic integrity presents a crucial difficulty despite the fact that these digital modalities provide unparalleled accessibility and convenience. Our project intends to pioneer a ground-breaking solution - the creation of an automated proctoring program - to address these issues head-on. This application uses cutting-edge technology to automatically proctor and oversee online exams, such as artificial intelligence (AI) and machine learning (ML). It promises to revolutionize how we guarantee the integrity of remote assessments by seamlessly integrating with current online examination platforms. The main goal of this project is to automate the proctoring procedure, do away with manual involvement, and offer a scalable, effective, and private solution. We will examine the current problems with conventional proctoring techniques as we delve into the specific difficulties our project seeks to solve and clarify the potential advantages that this automated proctoring application offers to educational institutions and students alike. Our goal is to redefine proctoring standards for the modern era to improve the quality and fairness of online education.

Key features and objectives of our automated proctoring application include:

* Face Detection: The project aims to implement robust face detection technology, enabling the system to detect the identity of the test-taker and ensure that the person taking the exam as a registered student.
* Face Cover/Visibility Detection: It will incorporate sophisticated algorithms to detect face coverings or visibility issues, ensuring that test-takers' faces remain visible throughout the examination.
* Multiple People Detection with Face Verification: The project will extend its capabilities to detect and verify the presence of multiple individuals in the test environment, further enhancing security and integrity.
* Multiple People Detection: Beyond face verification, the system will employ advanced techniques to detect the presence of multiple people, alerting proctors to potential breaches of exam protocols.
  1. **Problem Statement**

Remote learning has increasingly been the standard in recent years, and online exams have also gained popularity. The successful proctoring of these online exams has become a chronic difficulty, though, as cases of cheating and unfair practices have been on the rise. The solutions currently in place frequently use a two-platform strategy. The actual testing is conducted on the first platform, which makes use of applications like Google Forms to make exam creation and administration easier. The second platform consists of video conferencing tools like Google Meet or Zoom that are used for manual proctoring, which involves having a proctor or teacher present to oversee the exam procedure in person. By creating a cutting-edge program capable of automated proctoring and student monitoring, our initiative intends to take on this obstacle head-on and eliminate the need for manual proctoring support. Our main goal is to develop a system that integrates easily into the online exam environment, improving its efficiency and security.

* 1. **Objectives**

**1. To verify identity via face recognition:**

Implement a robust face verification system to authenticate the identity of the test-taker, ensuring that the person taking the exam is indeed the registered student.

**2. To ensure the visibility of the test-takers face throughout the examination:**

Incorporate facial visibility checks to monitor and ensure that students maintain a clear and unobstructed view of their face throughout the examination.

**3. To enhance face recognition to detect multiple people and prevent impersonation:**

Enhance face verification capabilities to detect and prevent any attempts at impersonation or multiple individuals attempting to take the exam simultaneously.

**4. To accurately identify the presence of multiple individuals in the test-taker's environment:**

Implement a system that can accurately detect the presence of multiple individuals in the test-taker's environment, preventing unauthorized collaboration.

**5. To monitor the use of multiple browser tabs or windows during the exam:**

Implement mechanisms to detect and restrict the opening of multiple browser tabs or windows during the examination to maintain a single, controlled exam environment.

**6. To maintain a full-screen view of the examination interface throughout the test duration:**

Ensure that the examination interface remains in full-screen mode throughout the test duration, preventing distractions or unauthorized access to external resources.

These objectives collectively aim to create a secure, technologically advanced remote exam monitoring system that ensures academic integrity, privacy, and fairness in remote assessments.

**Chapter 2**

**Literature Survey/ Background**

**TITLE: Automated Online Exam Proctoring**

**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 webcam, 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: An Intelligent System for Online Exam Monitoring**

**DESCRIPTION**: For the past few years, e-learning has become popular across countries because of its flexibility, availability, and user friendliness. As far as 7 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 webcams 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. The 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: Eye Gaze Tracking with a Web Camera in a Desktop Environment**

**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 Bio-ID 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 Verification using Correlation Filters**

**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 14 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**

**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 must only 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: Implementation of E-Proctoring in Online Teaching: A Study about Motivational Factors**

**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 17 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: Toward Constructing a Secure Online Examination System**

**DESCRIPTION**: Nowadays the rapid development of technology can be observed in every aspect of human life, with no exception to the educational world. Implementation of a secure online examination system is a hot topic. 18 Issues that should be addressed in the secure online examination system are computer and network security issues of the system and prevention of cheating by examinees. In our paper, we provide a website application and a secure network design which can prevent cheating by examinees and examine both in aspects of security and cheating prevention.

**TITLE: Online Examination System with Cheating Prevention Using Question Bank Randomization and Tab Locking**

**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. 20 An online examination system that can 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 investigated 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, final validation is done by the focus group respondents and end users for effectiveness and usability.

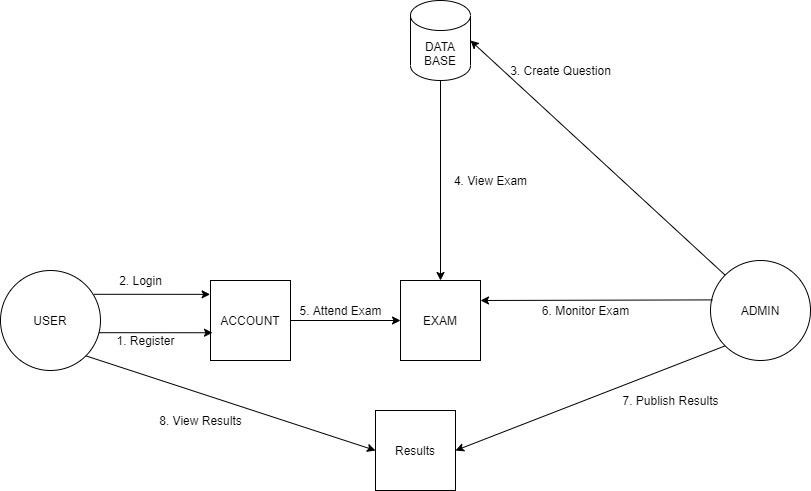
**TITLE: The Research and Application of Online Examination and Monitoring System**

**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**

**Software Design**

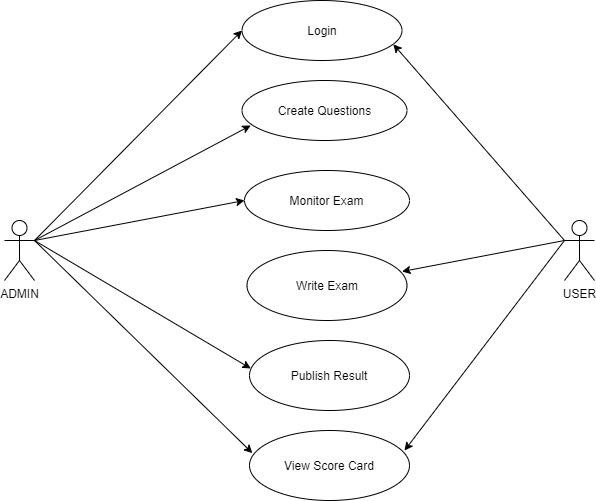
#### **3.1 SYSTEM ARCHITECTURE**

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 3.1.1** Architecture Diagram of Remote E-Proctoring System

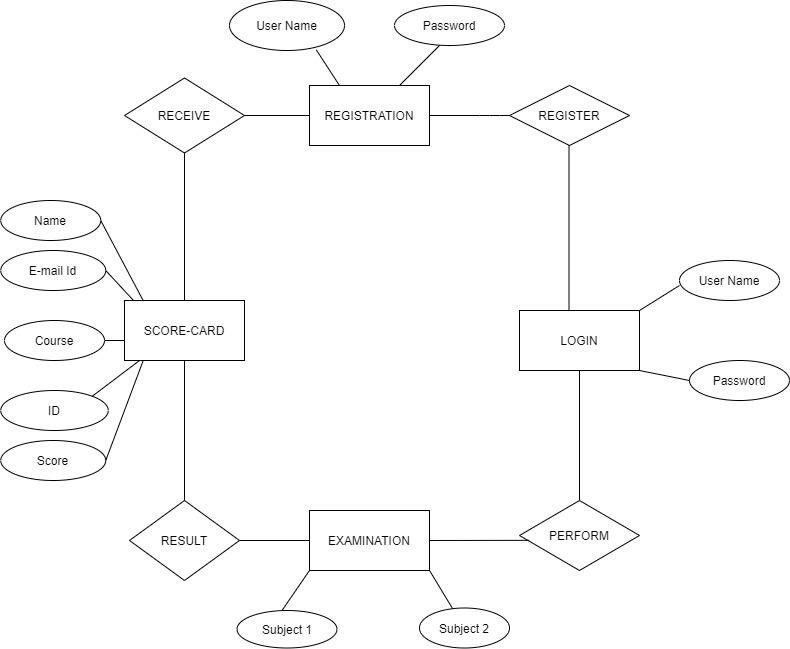
#### **3.2 UML DIAGRAMS**

##### **3.2.1 USE CASE DIAGRAM**



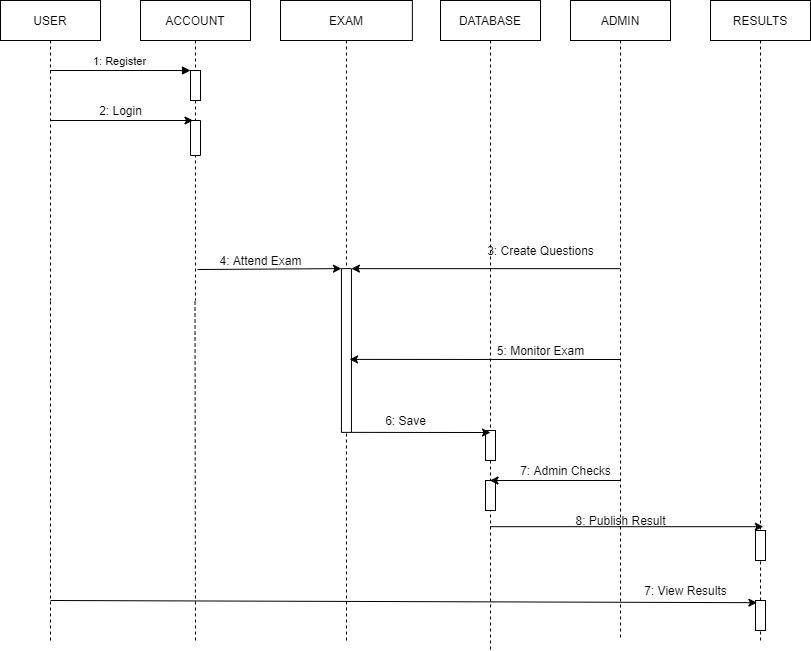
**Fig 3.2.1** Use Case Diagram of Remote E-Proctoring System

##### **3.2.2 ER DIAGRAM**



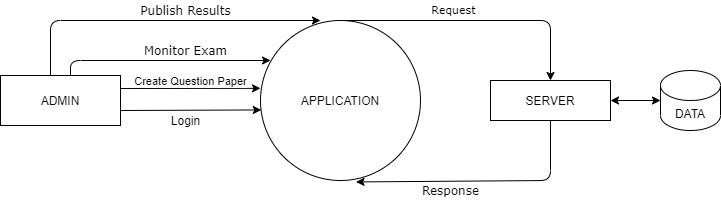
**Fig 3.2.2** ER Diagram of Remote E-Proctoring System

##### **3.2.3 SEQUENCE DIAGRAM**

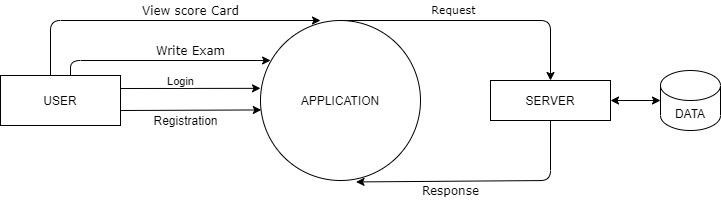


**Fig 3.2.3** Sequence Diagram of Remote E-Proctoring System

##### **3.2.4 DATA FLOW DIAGRAM**



**Fig 3.2.4.1**Admin Data Flow Diagram of Remote E-Proctoring System

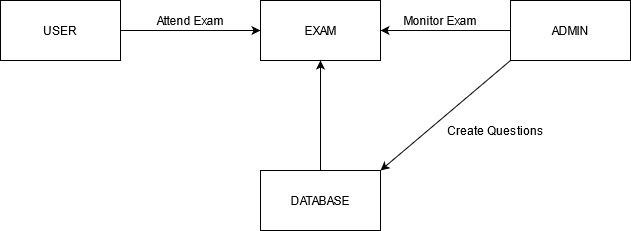


**Fig 3.2.4.2**User Data Flow Diagram of Remote E-Proctoring System

**Chapter 4**

**Requirements and Methodology**

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 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 several 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, like 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, gaze detection, number of person detection, object, and phone detection.



**Fig 4.1** Basic block diagram of E-Proctoring System

#### **4.1 SYSTEM CONFIGURATION**

##### **4.1.1 HARDWARE CONFIGURATION**

|  |  |  |
| --- | --- | --- |
| **S1.No** | **Name of the Hardware** | **Specification** |
| 1 | Processor | Intel Core i5 or equivalent |
| 2 | RAM | 8GB or higher |
| 3 | Storage | SSD with at least 256GB of free space |
| 4 | Internet Connection | High-speed Broadband Connection |
| 5 | Input/Output Devices | Camera, keyboard, and mouse |

**Table 4.1 Hardware requirements**

##### **4.1.2 SOFTWARE CONFIGURATION**

|  |  |  |
| --- | --- | --- |
| Sl. No | Name of the Software | Specification |
| 1. | Operating System | Windows 7 or later (64-bit) |
| 2. | Language | Python |
| 3. | Tools | Microsoft Visual Studio |
| 4. | Library | Tensorflow 2.2 |

**Table 4.2 Software Requirements**

#### **4.2 SOFTWARE SPECIFICATION**

##### **4.2.1 NODE.JS**

Node.js is a powerful, open-source, cross-platform runtime environment that allows developers to execute JavaScript code server-side. Built on Chrome's V8 JavaScript engine, Node.js offers an event-driven, non-blocking I/O model, which makes it highly efficient and suitable for real-time applications. This architecture enables Node.js to handle multiple simultaneous connections with high throughput, making it ideal for applications such as web servers, chat applications, and API services. Its NPM (Node Package Manager) provides access to a vast ecosystem of libraries and modules, facilitating rapid development. Node.js's ability to unify web application development around a single programming language, JavaScript, for both client and server sides, enhances productivity and code maintainability. Popular for its performance and scalability, Node.js is used by companies like Netflix, LinkedIn, and PayPal.

#### **4.2.2 EXPRESS**

Express.js is a minimal and flexible web application framework for Node.js, designed for building robust and scalable web applications and APIs. It simplifies the development process by providing a thin layer of fundamental web application features, such as routing, middleware, and HTTP utilities. Express's straightforward and unopinionated design allows developers to structure their applications in a way that suits their needs, whether for simple web servers or complex RESTful APIs.

With its vast ecosystem of plugins and middleware, Express extends Node.js's capabilities, making tasks like handling requests, responses, cookies, and sessions easier. Its compatibility with various templating engines and support for dynamic content rendering enhances its utility in web development. Express is known for its performance, ease of use, and extensive documentation, making it a popular choice for both beginners and experienced developers. Widely adopted by companies such as Uber, IBM, and Accenture, Express.js continues to be a fundamental tool in modern web development.

#### **4.2.3 TENSORFLOW.JS**

TensorFlow.js is an open-source library that brings the power of TensorFlow, Google's popular machine learning framework, to JavaScript. It enables developers to create, train, and deploy machine learning models directly in the browser or in Node.js environments. This allows for real-time, interactive applications and eliminates the need for server-side processing, making machine learning more accessible and responsive.

Key features of TensorFlow.js include its ability to run pre-trained models or develop new models from scratch using high-level APIs. It leverages WebGL for GPU-accelerated computation, enhancing performance for complex mathematical operations required in machine learning.

TensorFlow.js supports various use cases, from browser-based image recognition and natural language processing to interactive art and education tools. By enabling machine learning in JavaScript, TensorFlow.js opens new possibilities for web and mobile developers, providing an engaging way to incorporate advanced AI features into applications. Its versatility and ease of use have made it a valuable tool for developers exploring the intersection of machine learning and web technologies.

#### **4.2.4 MONGODB**

MongoDB is a leading NoSQL database known for its flexibility, scalability, and performance. Unlike traditional relational databases, MongoDB stores data in JSON-like documents, allowing for a more dynamic schema. This document-oriented approach enables developers to store and query complex data structures more efficiently.

Key features of MongoDB include its powerful query language, support for indexing, and rich aggregation framework, which facilitates sophisticated data analysis and manipulation. MongoDB's horizontal scalability, achieved through sharding, allows it to handle large volumes of data and high-traffic applications seamlessly.

The database is designed to support modern application development needs, offering high availability with replica sets and distributed architecture. MongoDB integrates well with various programming languages and platforms, making it a versatile choice for diverse use cases, from content management systems to real-time analytics and IoT applications. Its robustness and flexibility have led to its adoption by major companies like Adobe, eBay, and Google.

**4.3 Methodology**

The project employs a robust system for user authentication and role management, ensuring secure access and role-specific functionalities for students and teachers. Users log in through a secure authentication process, where each account is assigned a distinct role, either student or teacher, with corresponding permissions. This secure user authentication and role management is implemented using JSON Web Tokens (JWT) to ensure that each session is securely maintained, while bcrypt.js is used to hash passwords, adding an extra layer of security to user credentials.

Teachers are provided with comprehensive capabilities to manage exams. They can create exams, define questions, and configure various exam settings. This includes specifying question types, setting time limits, and organizing the overall structure of the exam. The exam management interface for teachers, built with React and styled using Material-UI, is designed to be intuitive, allowing them to efficiently manage and modify exams as needed.

Students, on the other hand, have a streamlined experience focused on participating in exams. They can view a list of available exams and select which ones to take. During the exam, the test page, developed with React and Redux Toolkit, displays each question along with a timer that ensures the exam is completed within the allocated time. The timer is equipped with an auto-submit feature that submits the student's answers automatically when the time expires, thus ensuring fairness and adherence to exam rules.

A significant feature of the system is the implementation of real-time AI proctoring. TensorFlow.js is employed to bring machine learning models directly to the browser, enabling real-time monitoring during exams. This advanced functionality enhances the integrity of the exam process by detecting cheating behaviours, such as the presence of mobile phones, multiple faces within the camera's view, and the absence of a face, which could indicate the student has left the exam environment. The exam interface utilizes React Webcam to capture real-time video feeds for this purpose. Any incidents of suspicious behaviour are logged and made accessible to teachers through their dashboard, built with Material-UI components for a consistent and professional look. This allows teachers to review potential cheating incidents and take appropriate actions to maintain exam integrity.

The backend of the system is built using Node.js, which provides a powerful runtime environment for executing server-side code. Express.js is used to create a robust API that handles routing, middleware, and HTTP utilities, facilitating smooth communication between the server and client. The express-async-handler middleware simplifies error handling in asynchronous routes, improving code readability and maintainability. Data persistence is managed with MongoDB, a NoSQL database chosen for its flexibility and scalability, while Mongoose provides a straightforward schema-based solution to model application data, ensuring consistent and organized data management.

On the frontend, React is employed to build a dynamic and responsive user interface. The Redux Toolkit manages the application's state efficiently, allowing seamless data flow and state management across components. React Router is used to handle client-side routing, enabling a smooth navigation experience. Material-UI provides a set of customizable components that adhere to modern design principles, enhancing the user interface's aesthetics and usability. For form validation, Yup and Formik are integrated to create robust and user-friendly forms, ensuring data integrity, and improving user experience. React Toastify offers a way to display notifications and alerts, while SweetAlert provides attractive and responsive alert boxes, improving user interaction feedback.

Overall, the system's architecture ensures that user roles and permissions are securely managed, exam functionalities are robust and user-friendly, and the AI proctoring adds a critical layer of security to uphold the standards of the examination process. By leveraging these modern technologies, the project delivers a cohesive and efficient platform for educational assessments.

**Chapter 5**

**Coding /Code Templates**

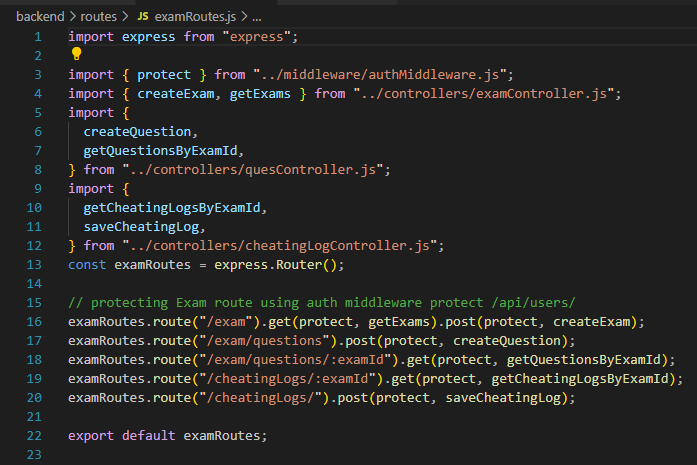
A screen shot of a computer program

Description automatically generated

**Fig 5.1** Database Connection with the backend



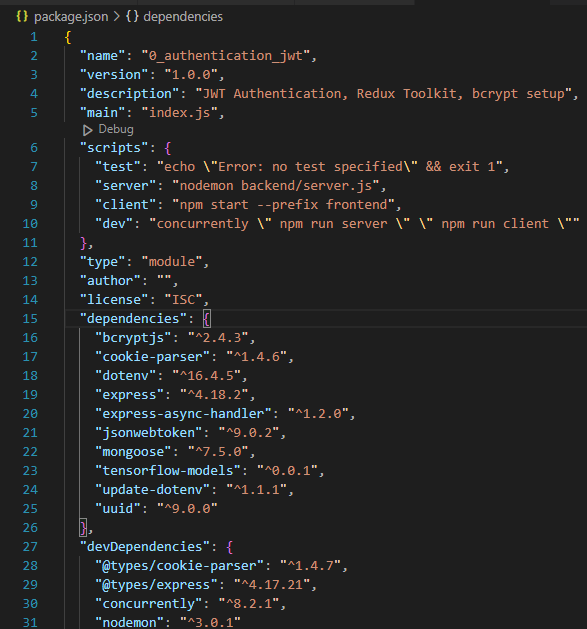
**Fig 5.2** Web Token Generation for the authentication

 **Fig 5.3** Defining Exam Routes

A screen shot of a computer screen

Description automatically generated

**Fig 5.4** Defining User Routes



**Fig 5.5** Package. Json File of the Project

**Chapter 6**

**Testing**

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 | Sign Up using Credentials | Validating the Credentials and Proceeding to  Login Page | Validating the Credentials and  Proceeding to Login Page | Pass |
| 2 | Login using 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 | Instructions Page | Display the rules and instructions for the exam | Display the rules and instructions for the exam | Pass |
| 7 | Cheating log Results | Display the cheating log of each student for every exam created | Display the cheating log of each student for every exam created | 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 | Mobile phone Detection | Check the presence of the mobile phones and show the warning alert | Check the presence of the mobile phones and show the warning alert | Pass |

#### **Table 6.1** Testing Test Cases

#### **6.1 TYPES OF TESTS**

##### **6.1.1 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.

##### **6.1.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they 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.

##### **6.1.3 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.

##### **6.1.4 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.

##### **6.1.5 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.

#### **6.1.6 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.

#### **6.1.7 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, like 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 7**

**Results and Discussion**

**APPENDICES**

**A screenshot of a computer

Description automatically generated**

**REGISTERATION PAGE**

**A screenshot of a computer

Description automatically generated**

**LOGIN PAGE**

**A screenshot of a computer

Description automatically generated**

**TEACHER DASHBOARD**

**A screenshot of a computer

Description automatically generated**

**STUDENT DASHBOARD**

**A screenshot of a computer

Description automatically generated**

**ADDING QUESTIONS IN EXAM**

**A computer screen shot of a computer screen

Description automatically generated**

**TEST INSTRUCTIONS**

**A screenshot of a computer

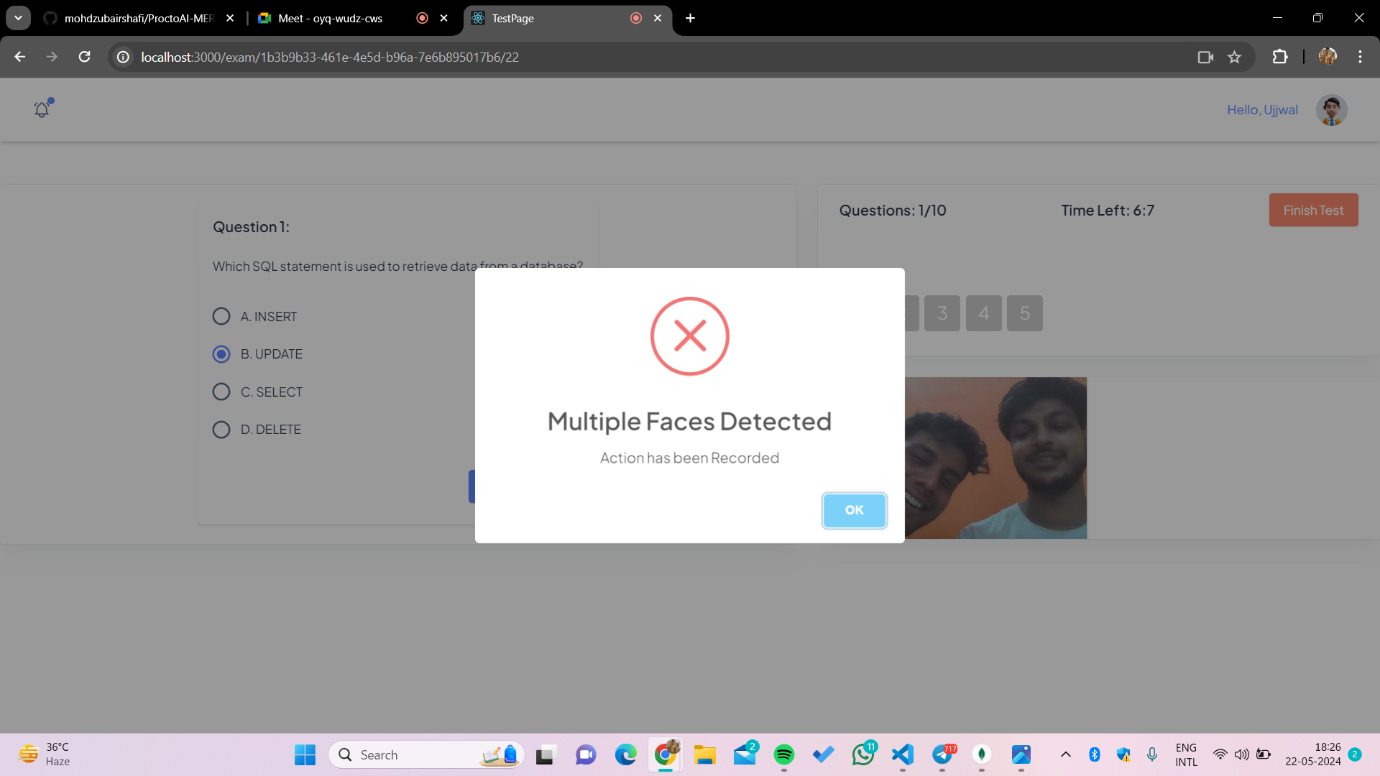
Description automatically generated**

**EXAM PAGE**

**A screenshot of a computer

Description automatically generated**

**PROHIBITED OBJECT DETECTED ALERT**

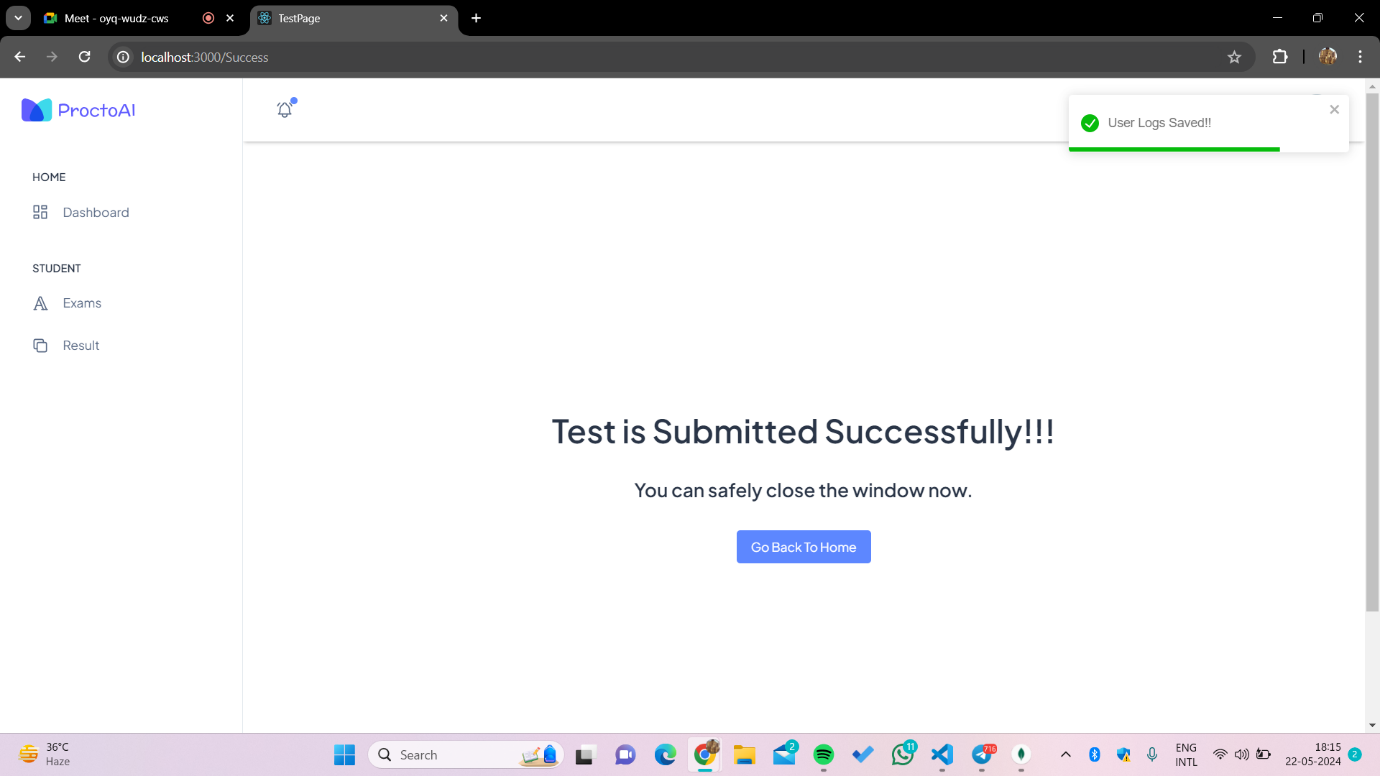
****

**MULTIPLE FACE DETECTED ALERT**

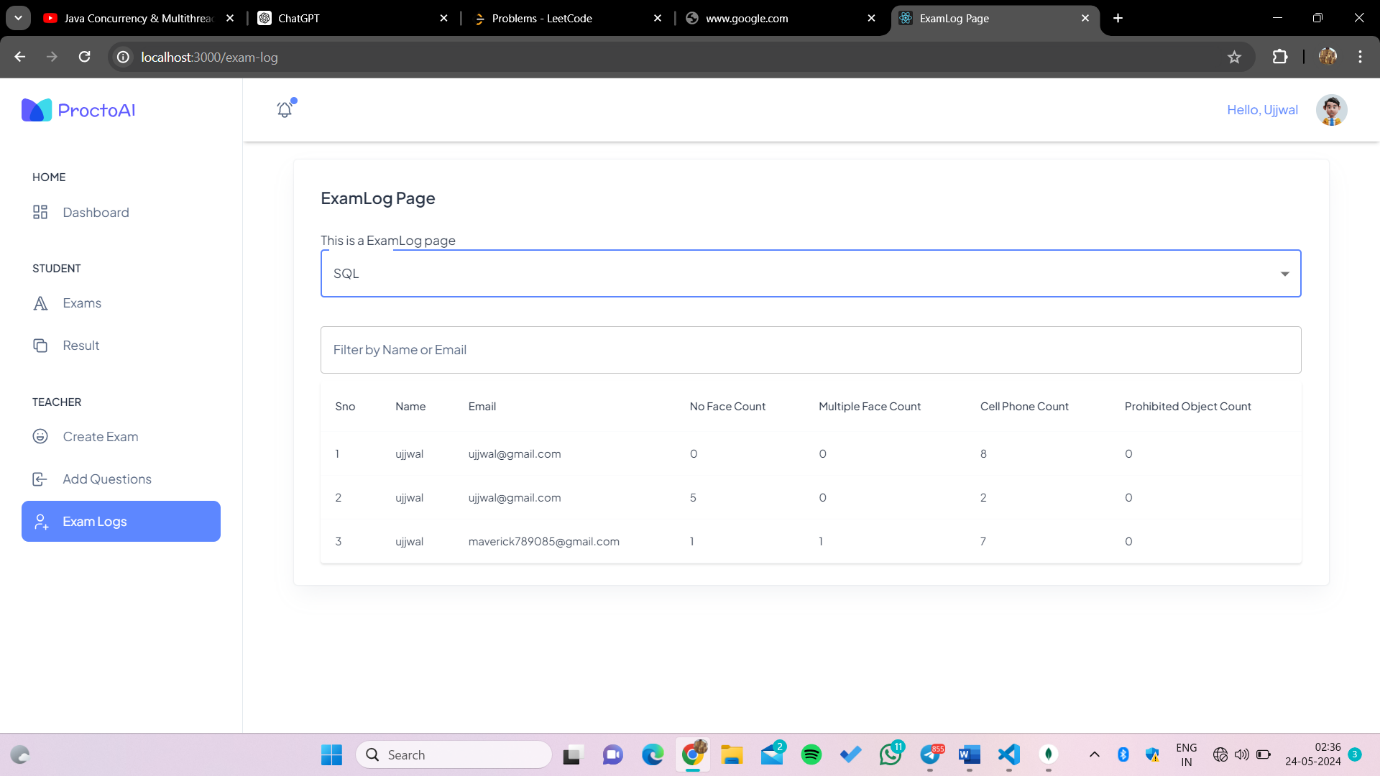
**A screen shot of a computer

Description automatically generated**

**FACE NOT VISIBLE ALERT**

****

**EXAM COMPLETION SUCCESS**

****

**CHEATING LOG OF AN EXAM**

**Chapter 8**

**Conclusion and Future Work**

#### **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 must be automatically run so that the transition is smooth so that the assesses 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.

**Details of Major Project Research Paper**

|  |  |  |
| --- | --- | --- |
|  | **Project Team ID** | MP23CSE067 |
|  | **Name of Supervisor** | Mr. Rishi Kumar |
|  | **Title of the paper** | PROCTO AN ONLINE PROCTORING SYSTYEM |
|  | **Authors Name** | 1. Dhruv Sharma 2. Ujjwal Kumar 3. Sattrajit Basu 4. Sarthak Sahai |
|  | **Percentage of plagiarism**  **(Check in Turnitin,**  **10 words)** |  |
|  | **Status of the research paper *(Project supervisor has to mark the appropriate one)*** | * Published/ Presented * Registered for conference * Accepted * Communicated * Not-Communicated |
|  | **Scopus Indexed**  **(mark the appropriate one)** | * Journal * Conference |
| Signature of students Signature of Supervisor | | |

**Table 8.1 Research Paper Details**

**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 MultiGesture 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 Behavior 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. 238242.
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. 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 devices. 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. 1 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.