

School of Sciences and Engineering

Department of Computer Science and Engineering

Classroom Monitoring Using AI

Project Report

Under Supervision of:

Dr. Sherif Aly Dr. [Hesham Eraqi](mailto:heraqi@aucegypt.edu)

Table of Contents

* Project Objectives and Importance
* Implementation and Testing
* Project Results
* User Guide
* Business Plan
* Project Objectives and Importance:

Cheating is one of the most unpleasant behaviors that is unfortunately common among many students all over the world. According to Bancud et al. in the paper titled “Human Pose Estimation using machine learning for cheat detection”, cheating is the use of unprohibited methods in order to gain an illegal academic advantage over others in exams. The most prevalent methods of cheating are: cheating directly from others’ papers, from electronic devices, or by communicating with others in exams. Taking Egypt, which is our main concern, as a case study to discuss the issue of cheating, we found that at the present time, the number of enrolled students in academic institutions, such as schools and universities has significantly increased, especially in Egypt, which has the highest population in the MENA region. With the increasing rates of enrolled students in academic institutions, higher rates of possible cheating cases can occur due to high-class densities and the inefficiency of the human factor in the process of proctoring the examination processes.

The aim of this project is to create a safe and fair examination environment for students that eliminates the high rates of cheating during examinations in Egypt, by implementing a system that detects possible cheating incidents in physical exams. Consequently, violating academic integrity creates a critical issue that negatively affects the reputation of the national education system and the quality, and competence of graduates.

* Implementation and Testing:
* System Overview and Design:

The functionalities of the project enable the user to be notified of possible cheating incidents in examination processes in a classroom environment. This includes a notification alert to the proctor in the classroom in addition to a snapshot of the possible cheating incident of the student with his/her location in the classroom with an option to dismiss the received possible case in case the proctor does not consider the case as a cheating case. A report is available for the proctor, in case he/she suspects the student’s action. The software generates a report at the end of the examination with detailed information about the dismissed and reported cheating incidents.

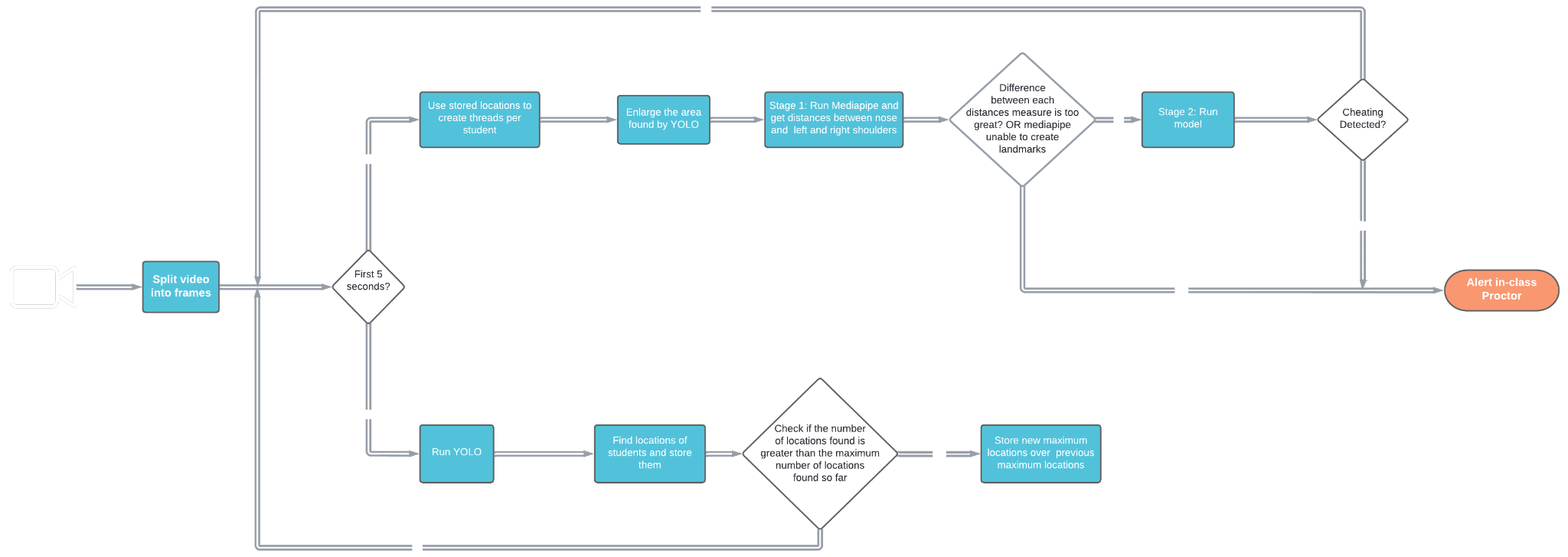
The project has different user classifications and characteristics, where each classification has its own privileges and authorizations, the classifications and characteristics are listed as follows:

* *System Admins*: System admins are responsible for assigning the users (proctors) to the exam rooms and the exam instances while maintaining the proctors’ registered data and credentials.
* *Proctors*: Proctors log in to the system using the provided credentials and are responsible for viewing and checking the real-time notifications received by the system and take decisions on the possible cheating incidents detected by the software.
* *System Support Team*: The system support team is responsible for maintaining the system and fixing possible issues and bugs that the users may experience while operating the system.
* System’s Deep Learning Model Design:



The camera provides the deep learning core a constant stream of frames which is then fed into our object detection system called YOLO where we save the locations of students and then initialize threads that run for each student.

After that, we enter a 2 stage process where we first go to stage 1 and use mediapipe, a framework by google for pose estimation where we measure the distance between the left shoulder and nose and also the distance between the right shoulder and nose. If the absolute value of the difference between the 2 distances exceeds a certain threshold then we assume that the person is cheating. If not then we use our own model to detect cheating and if the value exceeds the sensitivity then we take it into account when we get the average confidence per second which is also compared to the sensitivity. If it's higher then we send the image of the highest confidence cheating to the s3 bucket by AWS and send to our backend system that a cheating case was created which then is shown on our front end GUI to the proctor where they choose to ignore or report it. This is a more detailed diagram of the flow of data inside the python script core.

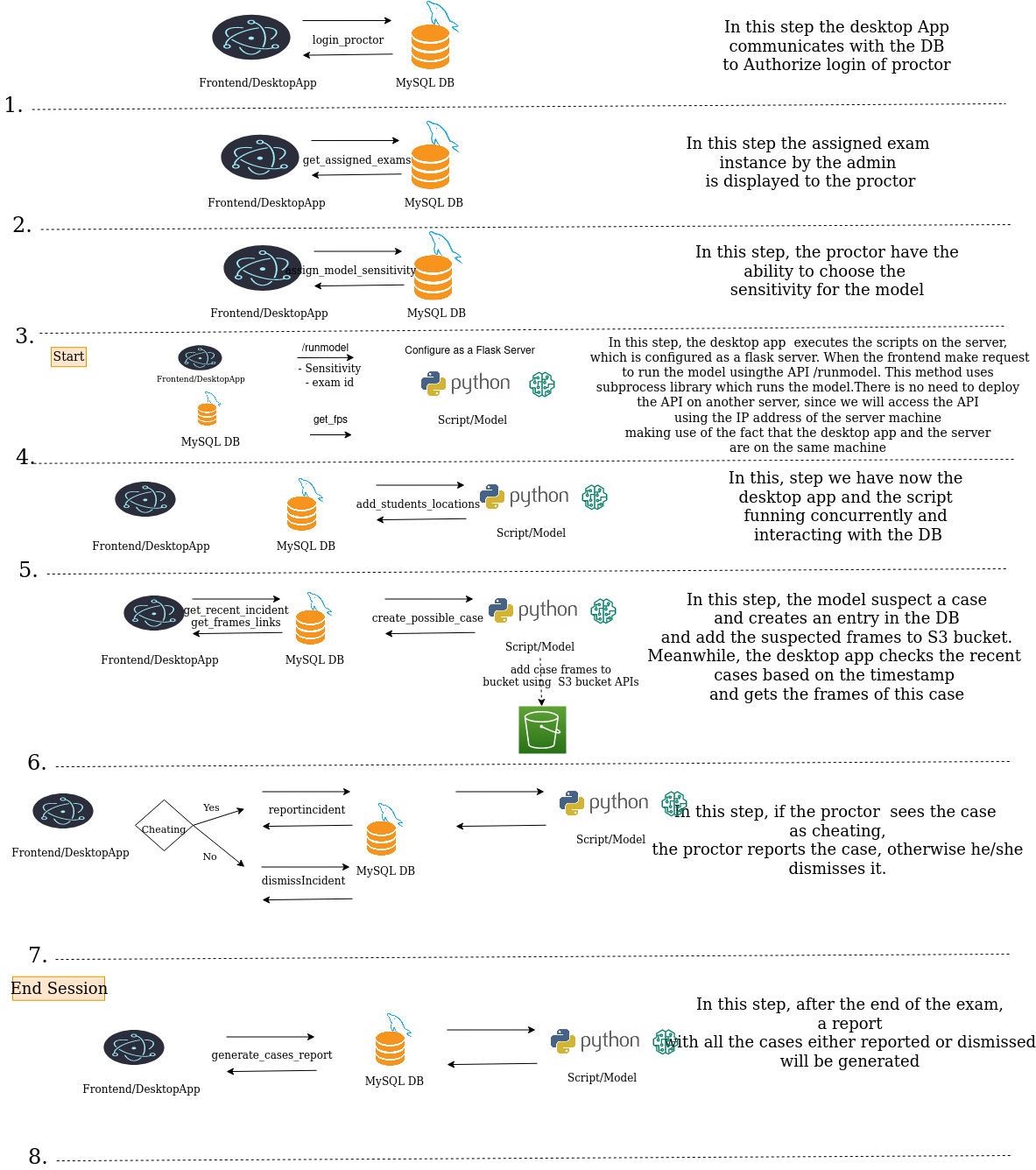


* System’s Backend Design:

The backend of our system is designed using Flask framework which is python based framework, that is known to be robust and open-source [9]. For the design of the database, we used MySQL and deployed the database of EC2 server. The rationale behind choosing MySQL for our database is that we have a well-defined schema and database design. Our APIs are mainly divided into two categories: Admin Level APIs and User Level APIs. The admin-level APIs are mainly used by the system owner/administrator in order to perform important functions. The following part is a use case example for admin-level APIs. If the system owner is a school admin, he/she can create exam instances, register proctors, and assign proctors to the exam instances to ensure that no one other than the admins can manipulate these functionalities. On the other hand, the user-level APIs are used mainly by the system users, i.e invigilators/proctors, and model. The methods utilized by the users are mainly dismissing a case, reporting a case, retrieving the frames of the case, and finally generating the exam report.

The user-level APIs are also consumed by the deep learning model by the fact that the model creates cases based on the feed inputted to the system and uploads the frame detected to AWS S3 bucket. For the data security, the users’ passwords are secured using the salting technique, by which we concatenate the password and application security key then the resulting string is hashed using the message-digest algorithm md5 [10]. The APIs are also authenticated and authorized by embedding tokens in the APIs to make sure that the authorized user is the one that can consume the API. For generating the tokens we used a python library called Flask-JWT, JSON web tokens.

The following communication diagram shows how the backend of the system communicates with the other nodes of the system:



* System’s Frontend Design:

The system’s front end has been designed to fulfill the users’ needs by developing multiple UI/UX designs before starting the development process. In addition to building a system design to effectively communicate with the system’s backend represented in the database and the Application Programming Interfaces (APIs) connecting it.

The software is a cross-platform desktop application that supports multiple operating systems: Linux, macOS, and Microsoft Windows. ElectronJS is a JavaScript framework that uses web development technologies such as HTML5, CSS3, Bootstrap 5, and JavaScript. It has been used to build cross-platform desktop applications and has been chosen to be the frontend development framework for our software, as it has high performance on different platforms, in addition to a smooth and quick development process due to having one code base for the three supported operating systems.

The software consists of 7 frontend interfaces that communicate with a total of 9 Application Programming interfaces (APIs). The frontend interfaces are as follows:

* Software Landing Page.
* User Login Page.
* Assigned Exam Details Page.
* Exam Sensitivity Page.
* Main Home Page.
* End of Exam Report Page.
* User Logout Page.

The software’s frontend communicates with 9 APIs that are hosted on the cloud to retrieve and send data to the system’s database, below are the APIs that are being fetched by the frontend

* */assign\_model\_sensitivity*: Assigns a sensitivity factor to the deep learning model, where the user is asked to choose from three options of: Low Sensitivity (25%), Medium Sensitivity (50%), and High (75%).
* */dismiss\_case*: The API is consumed when the user dismisses a possible cheating case as a noncheating case.
* */end\_exam*: The API is consumed when the user ends an exam session.
* /*generate\_exam\_report*: The API is consumed when the user ends an exam and is redirected to the end of the exam report page.
* */get\_exam\_instance\_details*: The API is consumed when the user logs in to the system successfully and his/her exam details are displayed.
* */get\_frames\_links*: The API is consumed when a new case is created and is used to display an image of the case on the frontend.
* /get\_recent\_case: The API gets consumed every 8 seconds, to fetch any new cases generated by the backend of the system.
* /login\_proctor: The API gets consumed after the user enters the login credentials to ensure user authentication and authorization.
* /report\_case: The API gets consumed the user reports a possible cheating case as a cheating case.
* Project Results:
* **User Guide:**

In this section, a user guide for the system will be introduced to assist the users in understanding the workflow of the software and how to interact with the system’s features.

The section will be divided into 3 subsections, including:

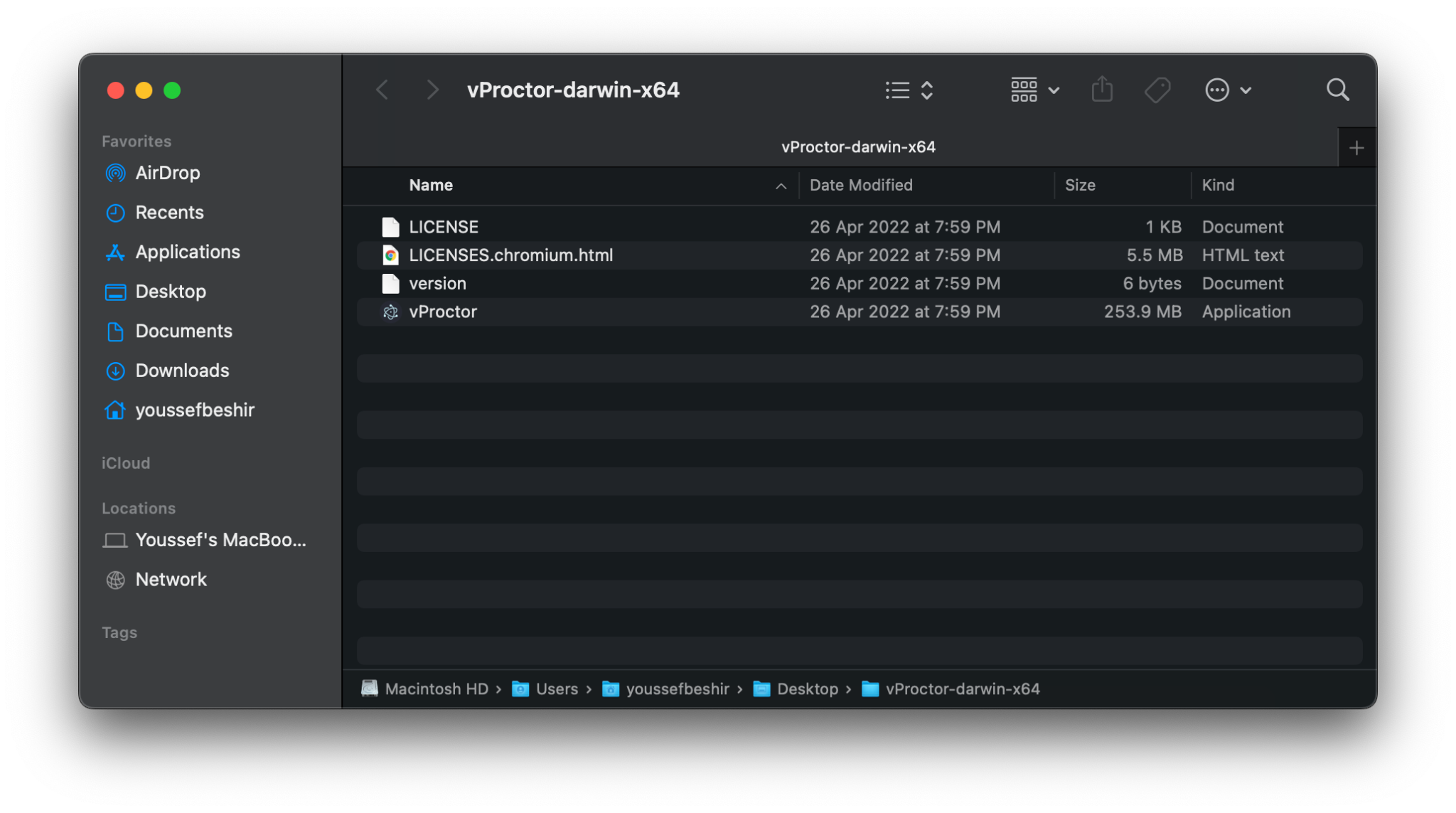
1. Proctor (Frontend) user guide
2. System Administrator (Backend) user guide
3. Server-side user guide.

* ***Proctor User Guide***:

The following guide will illustrate the steps of how the user (proctor) uses the system.

Step 1:

Run the software’s [vProctor] application (file extension may vary according to the used operating system, such as (.exe) for Microsoft Windows or (.app or .dmg) for Apple macOS).

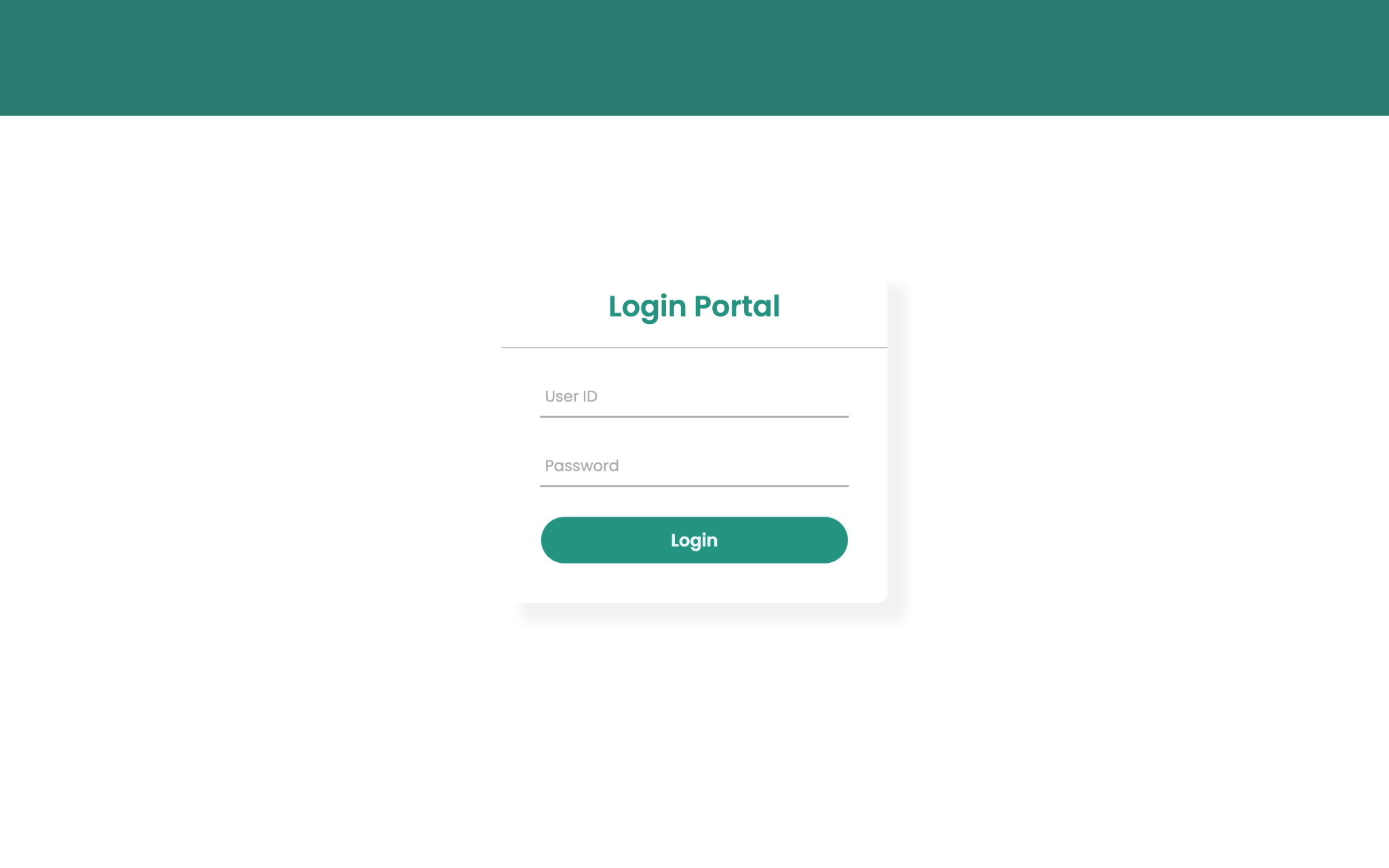


Step 2:

The software will open after the startup on the landing page.

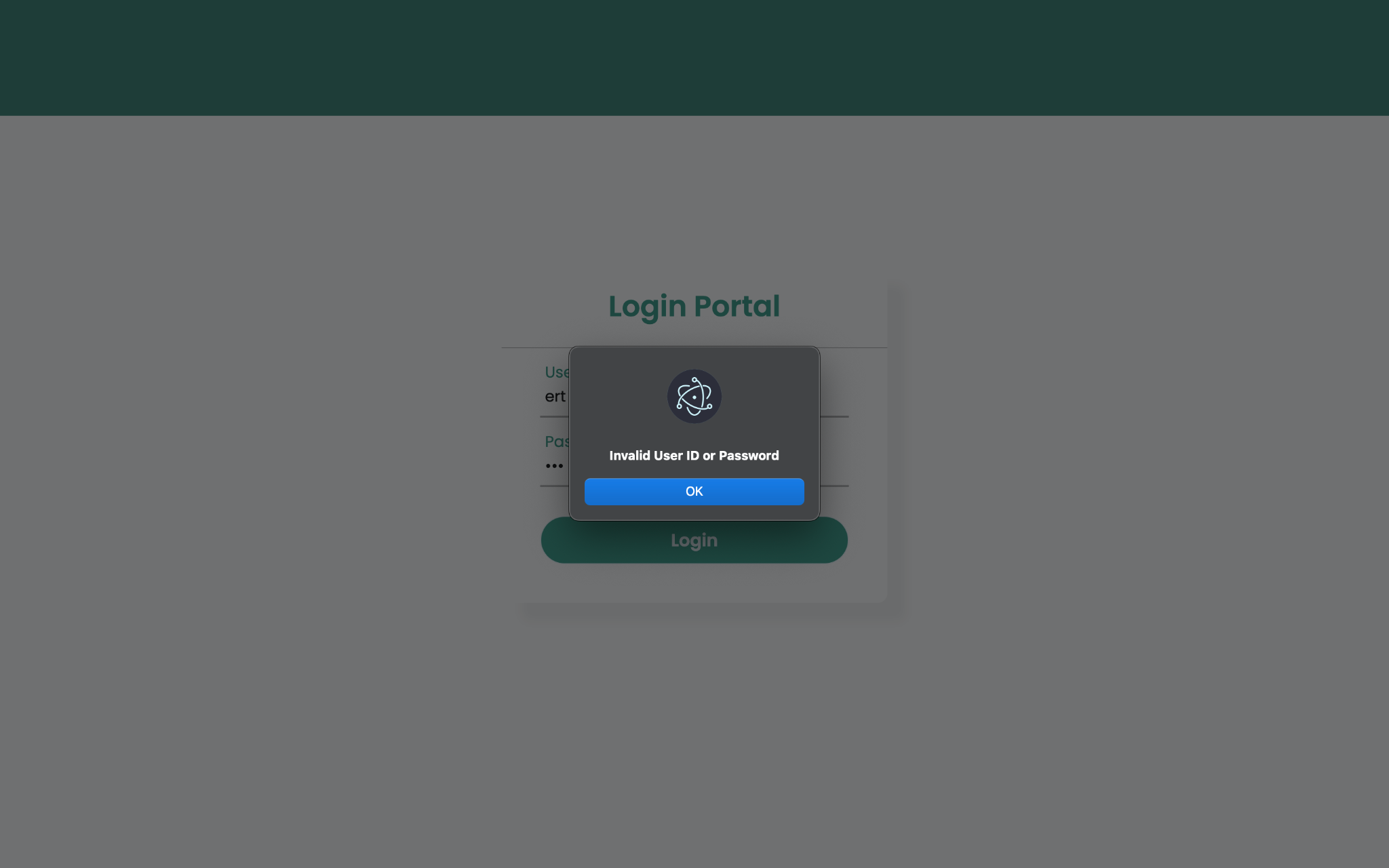
Press “Get Started” to start using the software.



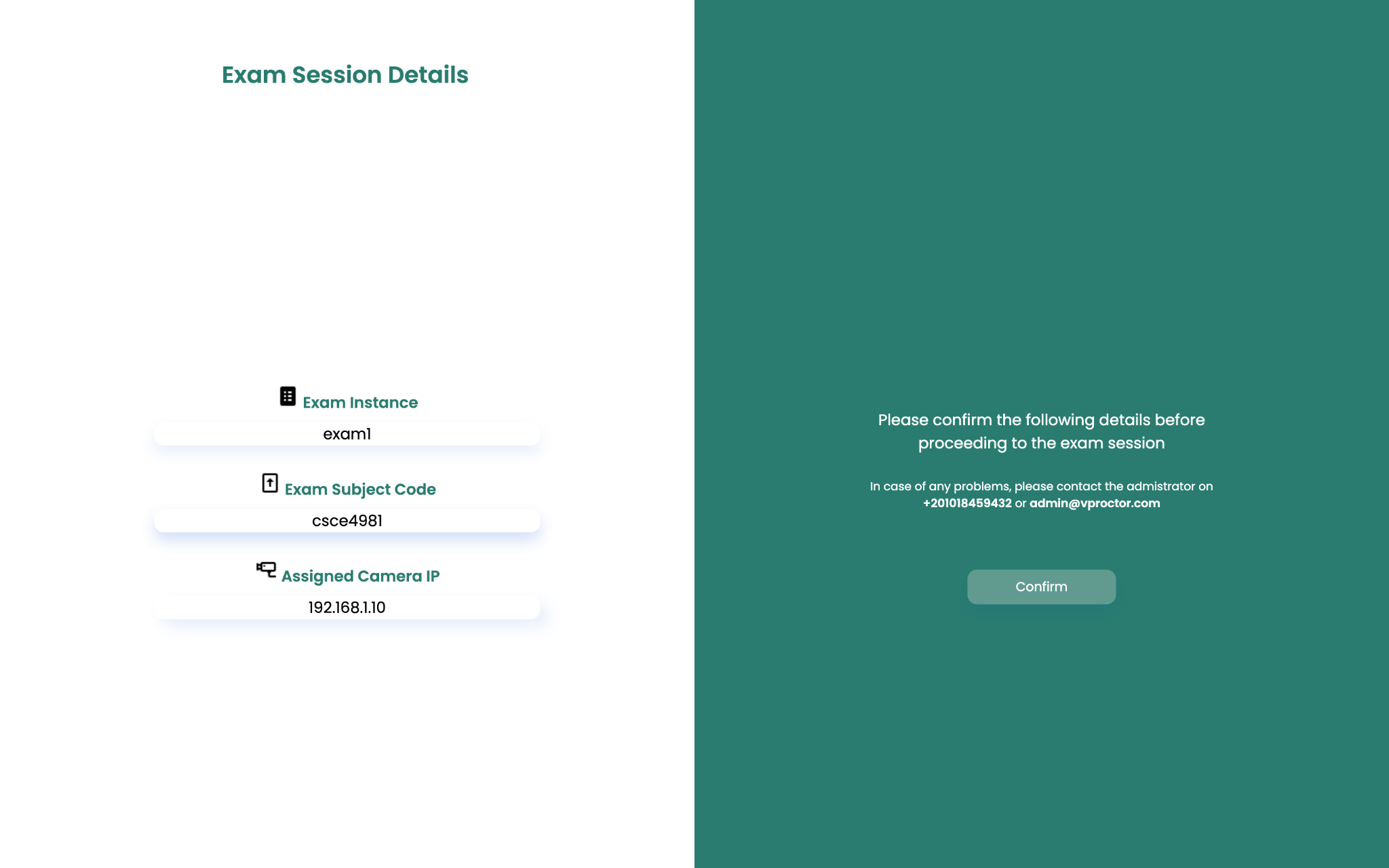
Step 3:

You will be redirected to the software Login Portal. Enter your login credentials: User ID and Password, which are provided by your system administrator.

In case you entered the wrong login credentials an alert message will pop up informing you that an error occurred. Enter the correct login credentials and try again



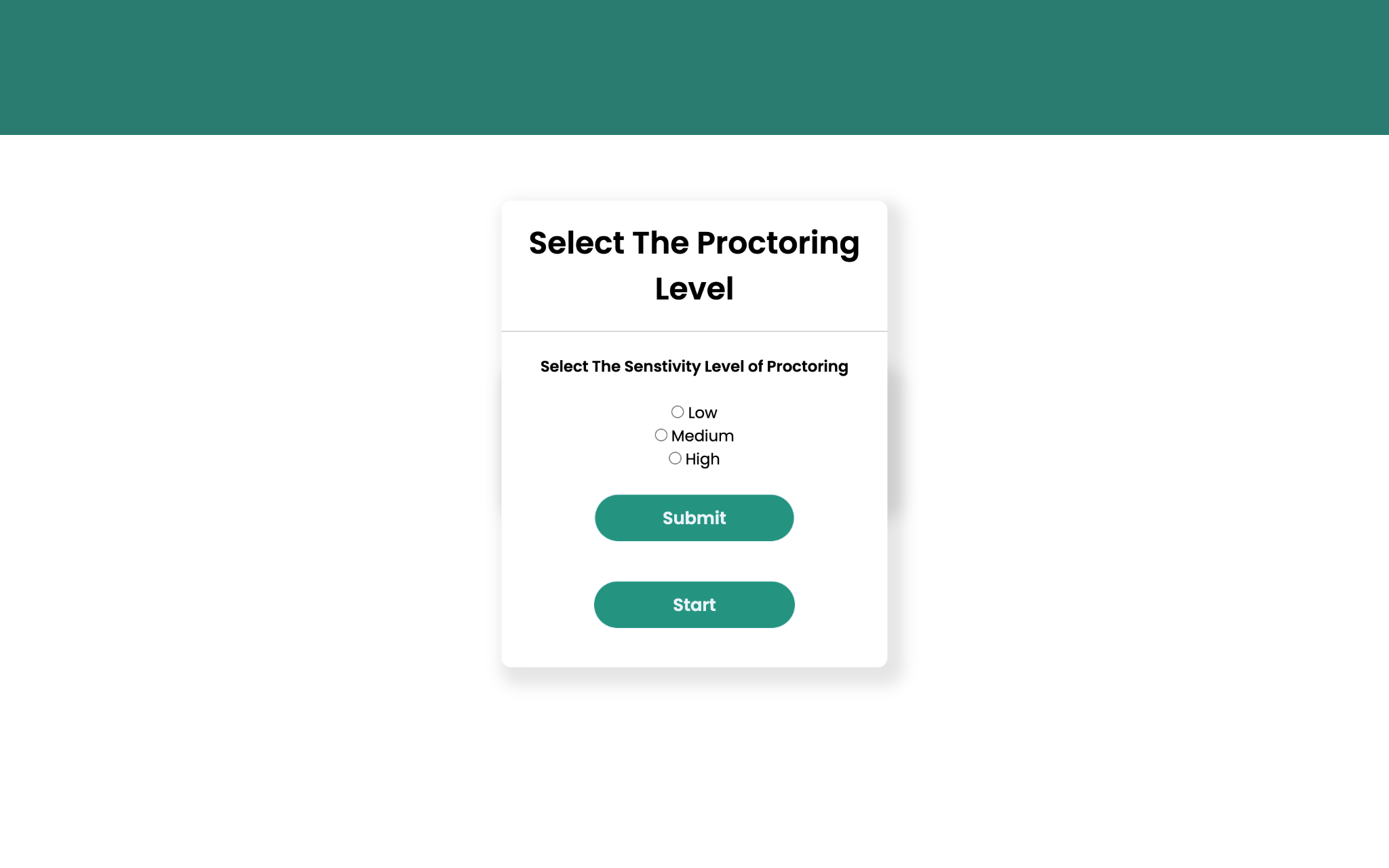
Step 4:

After successfully logging into the system, you will be redirected to the exam session details confirmation page. If the exam details are correct, you should press confirm. In case of any information mismatch, you should contact your system administrator.

Step 5:

You will be asked to select the sensitivity of the software in detecting the possible cheating incidents. There are 3 options:

* **Low** which corresponds to 25% AI Model Accuracy
* **Medium** which corresponds to 50% AI Model Accuracy
* **High** which corresponds to 75% AI Model Accuracy

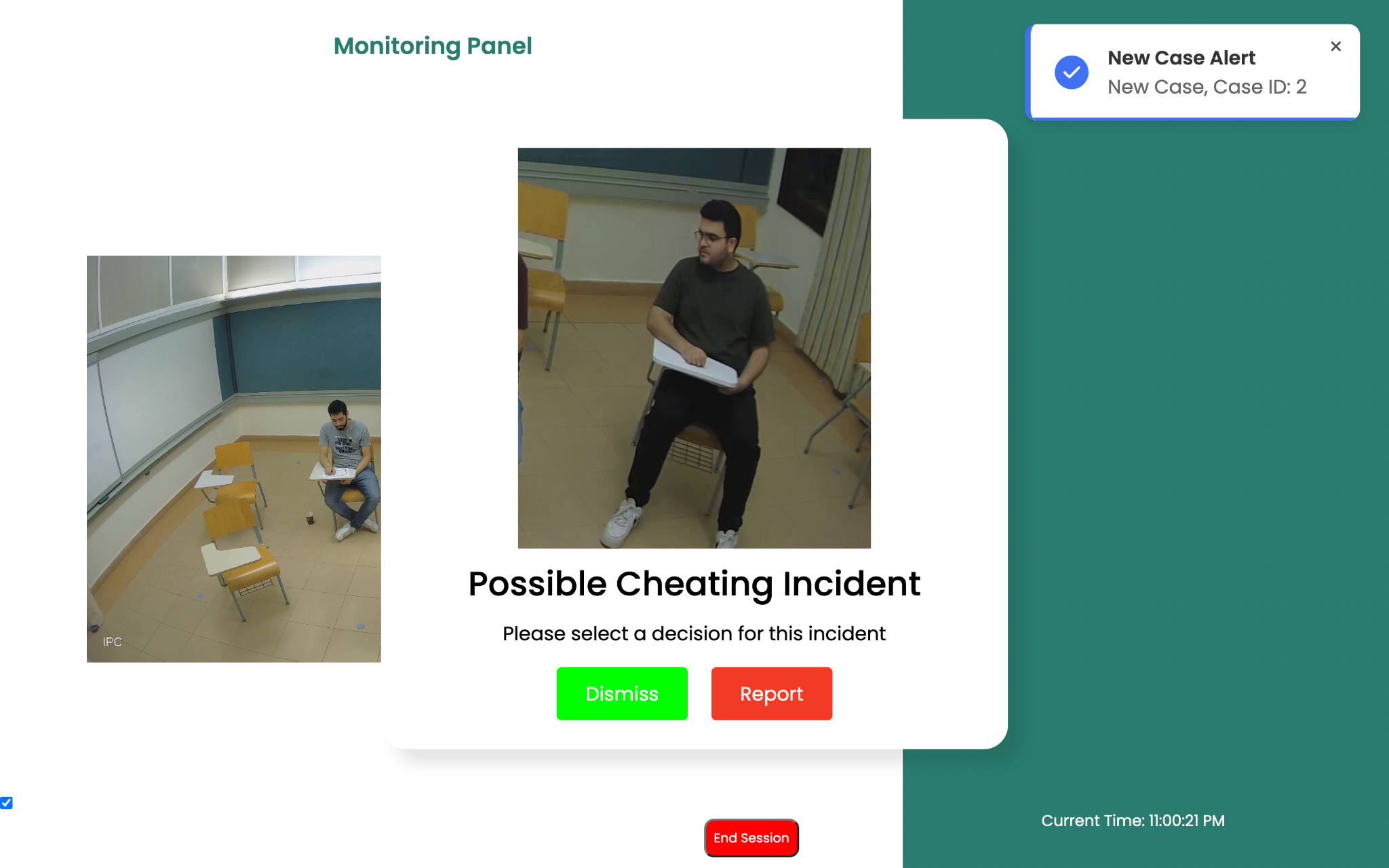
Press Submit and Start to initiate the software.

Step 6:

After successfully selecting the accuracy of the model and starting the exam session, you will be redirected to the main page of the software. In the main page, you will find a live feed from the IP camera that is placed in the classroom. A notification will appear at the most right side of the interface when the software detects a new possible cheating case, as shown below.

Once the notification is clicked, you will be prompted with a pop-up screen showing the case image that has been detected and two options, either to dismiss or report the case.

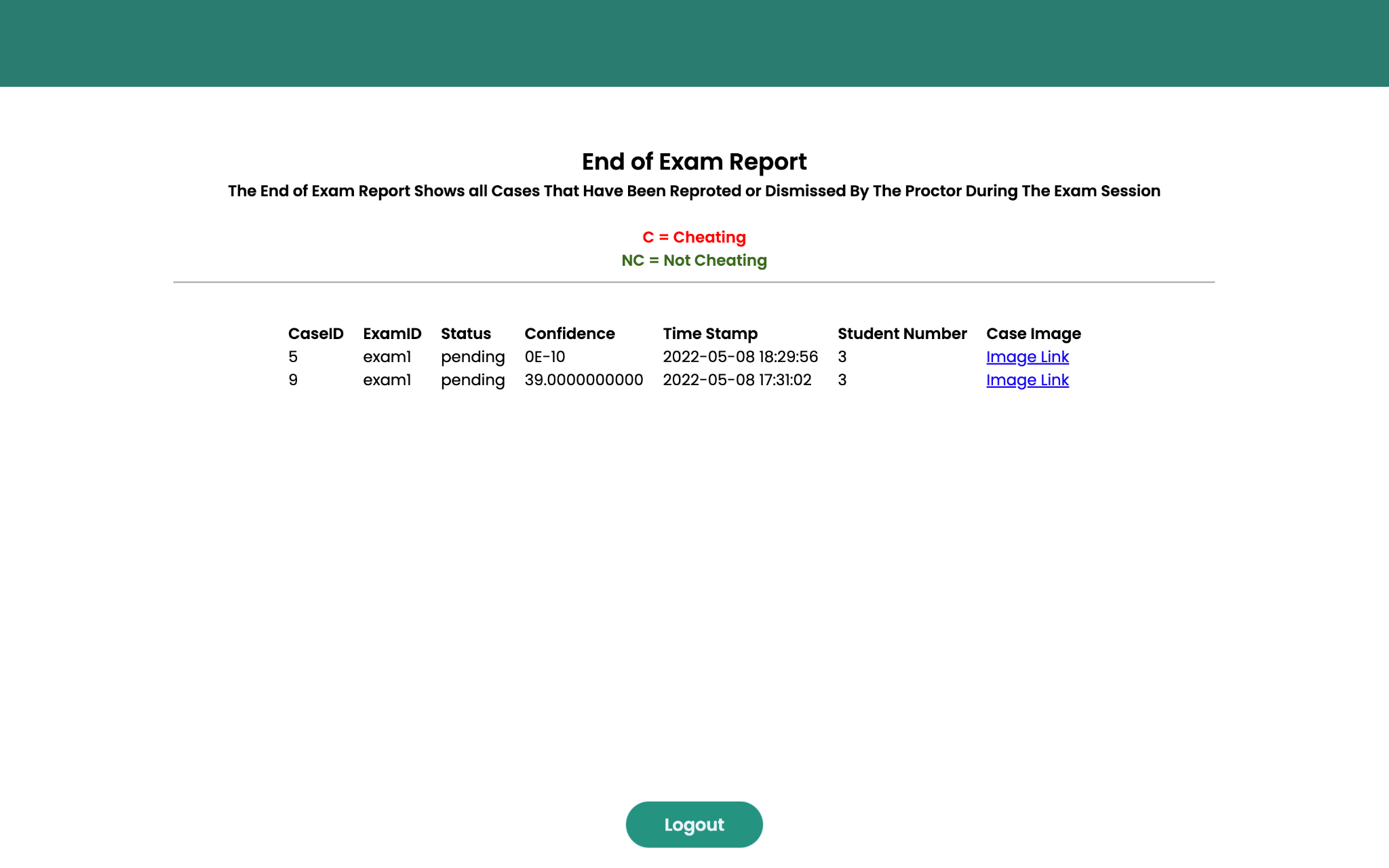




Step 7:

After you end the exam session, you will be redirected to the end of the exam report page where you can view the reported and dismissed cases during the exam session, in addition to the Case ID, Exam ID, Model Confidence, Time Stamp, Student Number, and a link to the case image.

You can click on the image link to view the case image again.



Step 8:

To log out from the software press on logout, and you will be logged out of the software.

* ***Admin User Guide***:

The following guide will illustrate the steps of how the system administrator can use the admin panel.

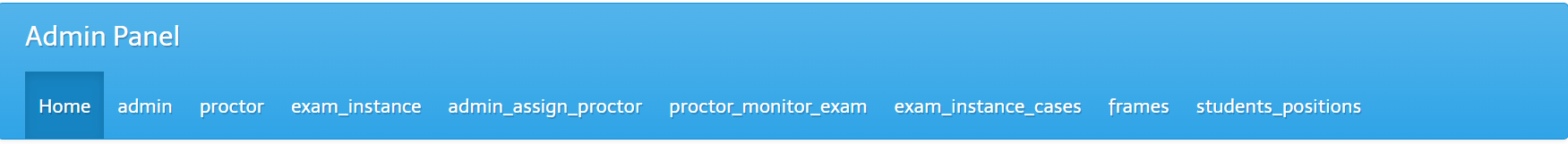
Step 1:

To access the admin panel of the software you need to access this URL: https://classroommonitoring.herokuapp.com/admin/

Enter your administrator credentials to access the panel.

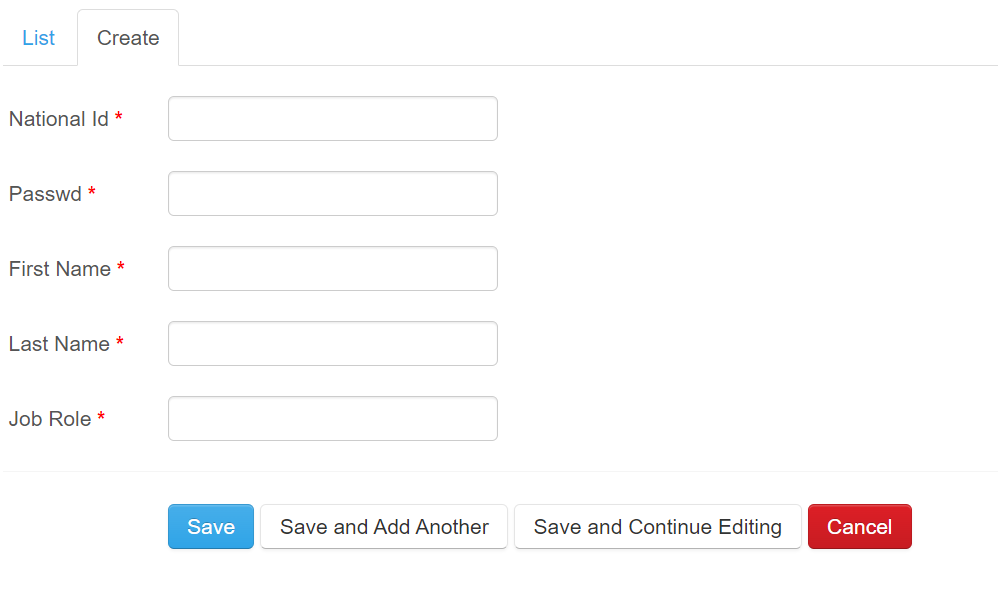
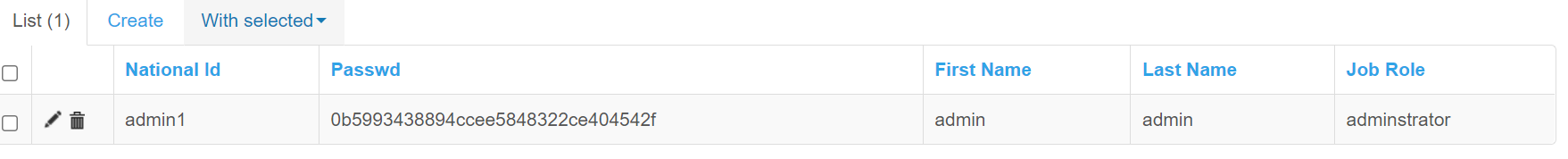
Step 2:

After successfully logging into the admin panel, you will be redirected to the home page. You will find the below options in the tab bar.



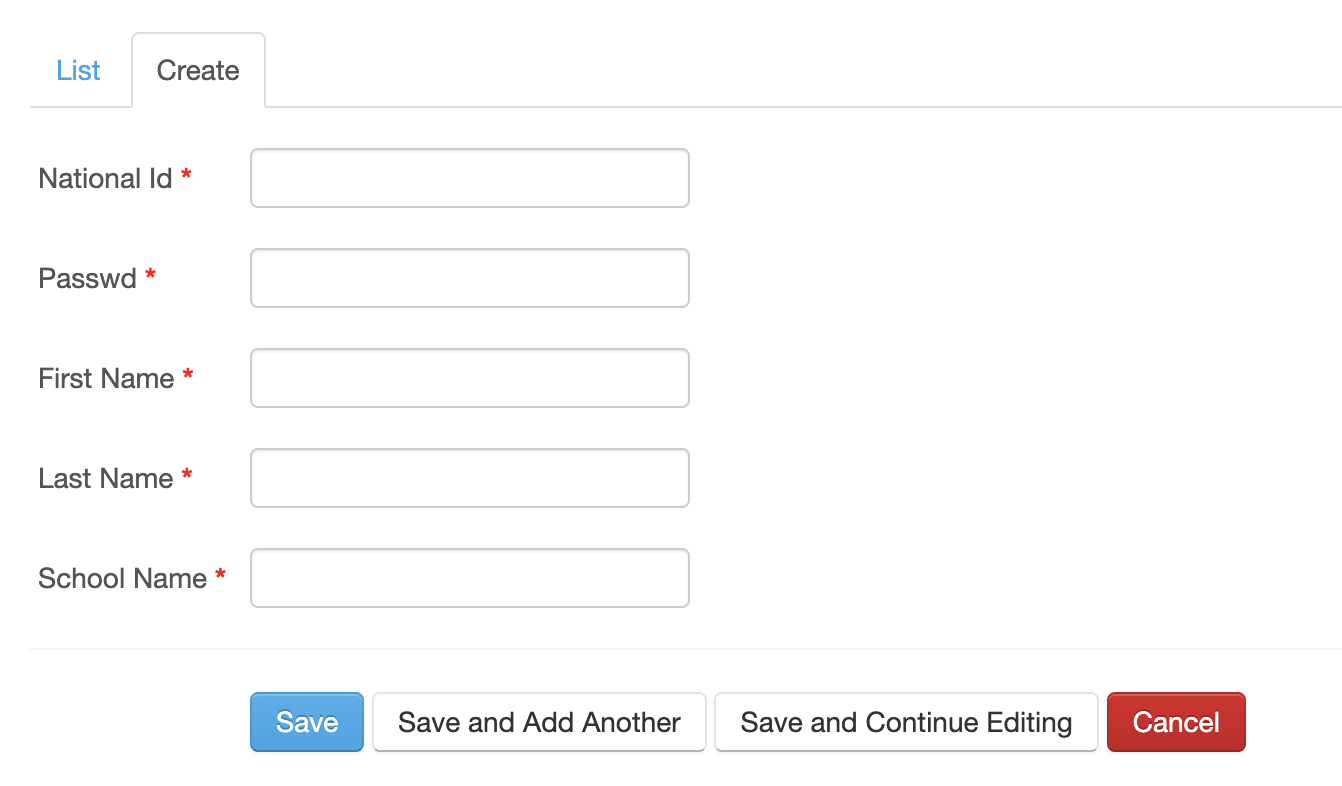
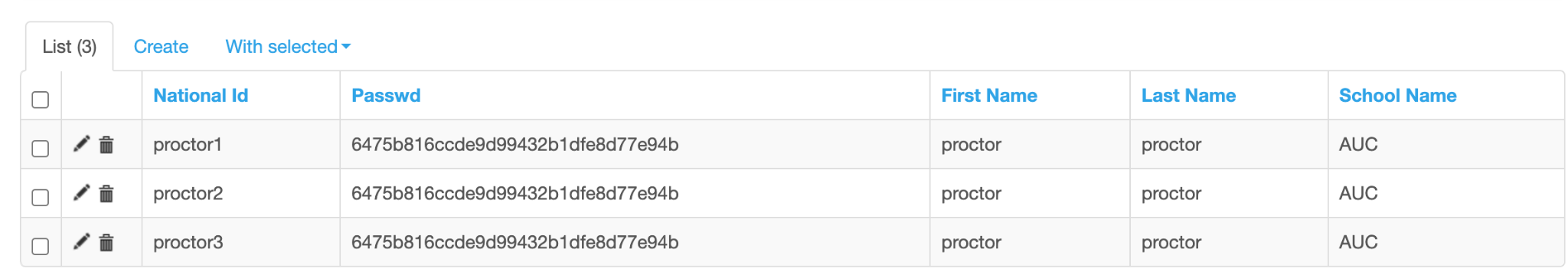
Step 3:

Click on the admin tab to view the information of registered administrators. The information includes the adminID, hashed password, First name, Last name, and job role. You can edit or delete the information by pressing the edit or delete buttons. In case you need to create a new admin user, click on create an option on the top bar and enter the required information.



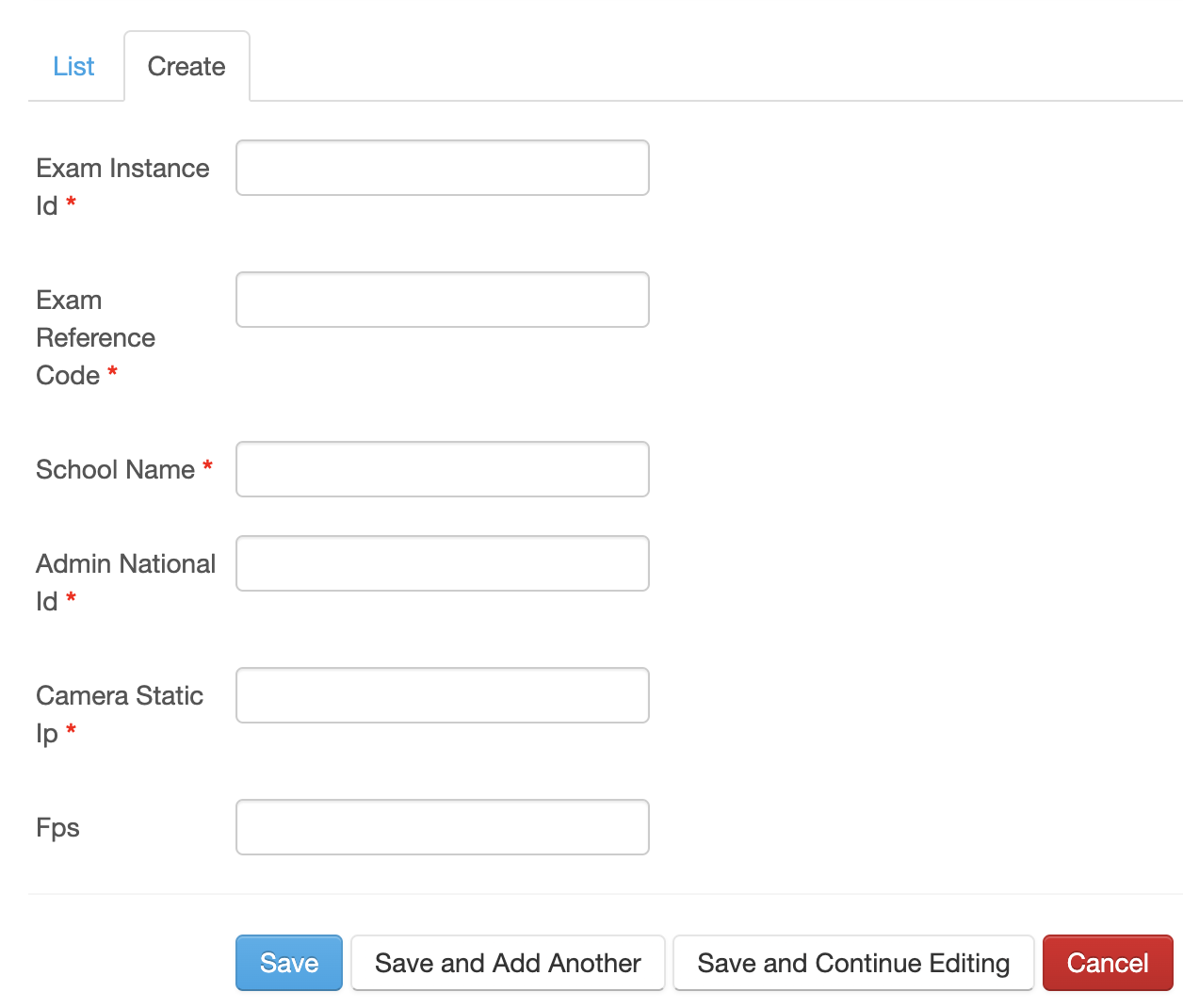
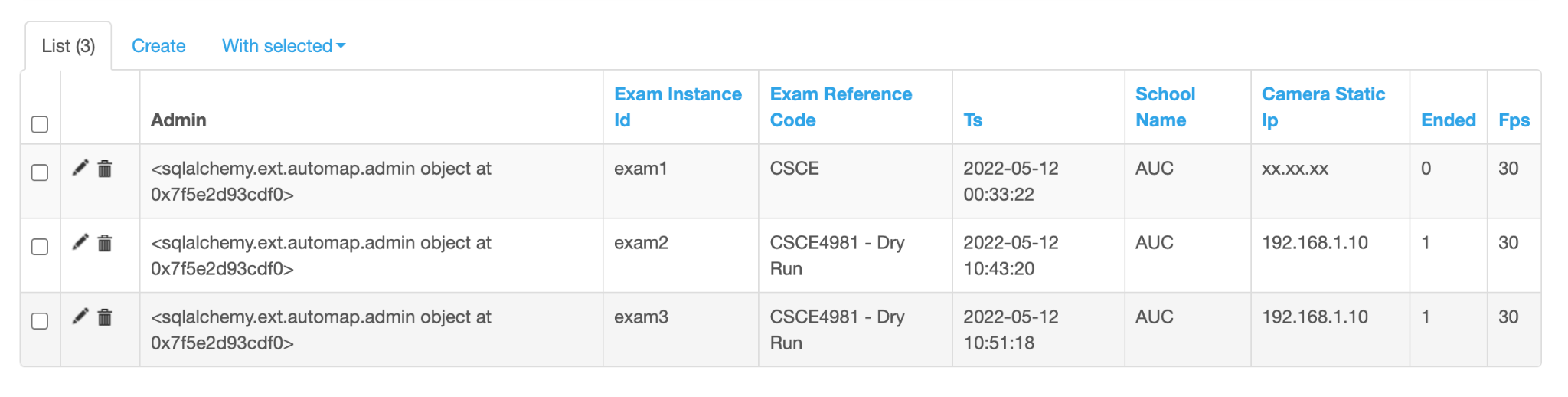
Step 4:

Click on the proctor tab to view the information of registered proctors. The information includes the proctorID, hashed password, First name, Last name, and School name. You can edit or delete the information by pressing the edit or delete buttons. In case you need to create a new proctor user, click on create an option on the top bar and enter the required information.



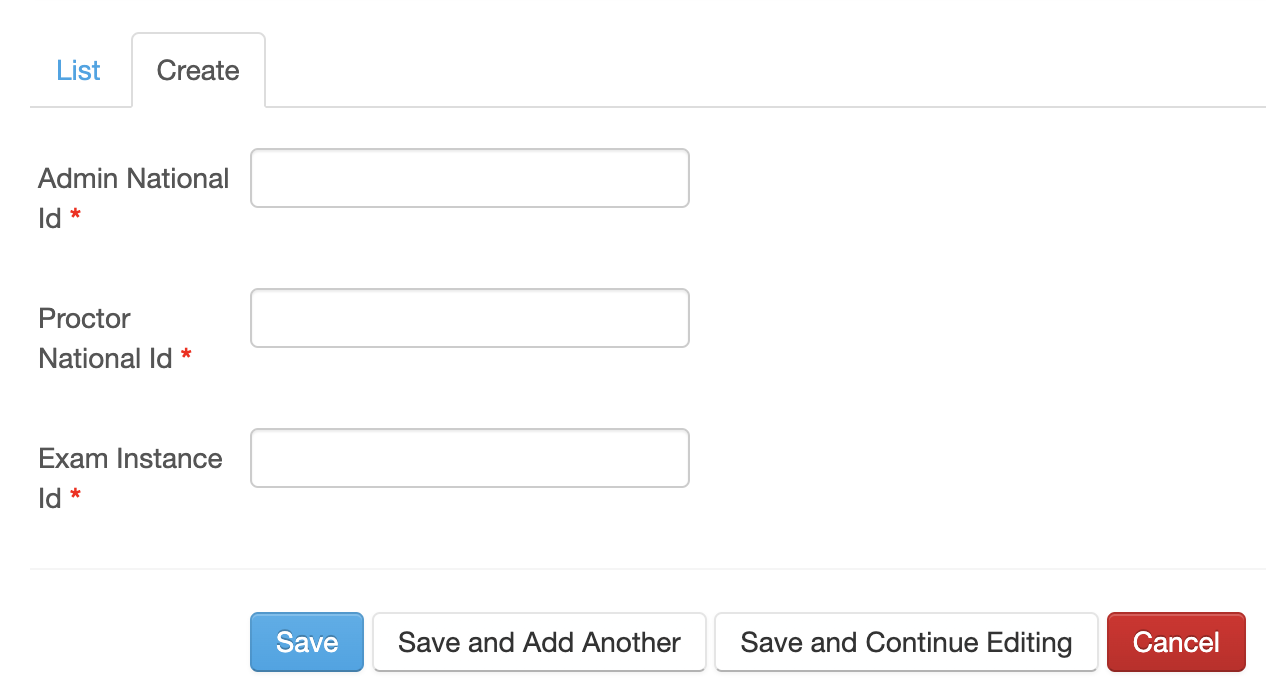
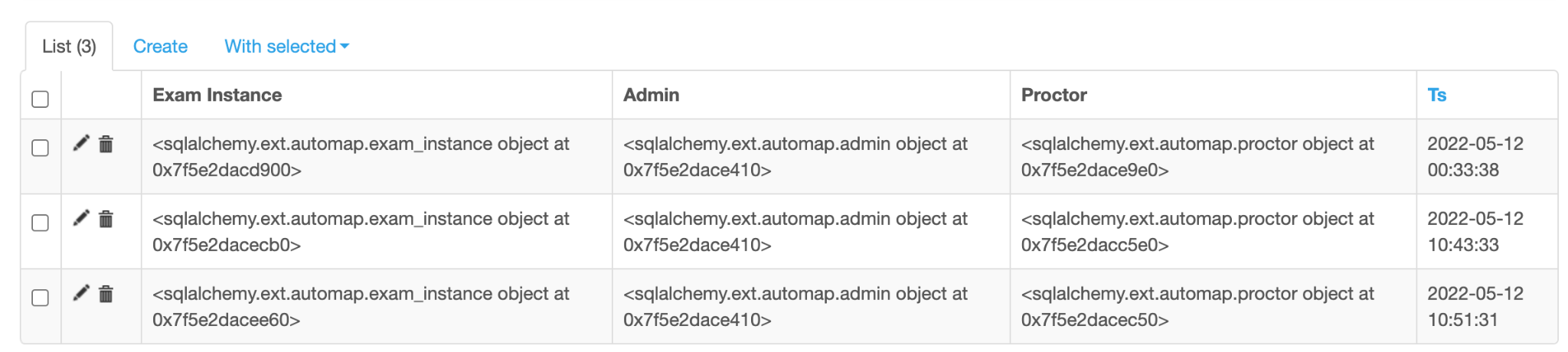
Step 5:

Click on the exam instance tab to view the information of the exam instances. The information includes the admin, exam instance id, exam reference code, time stamp, school name, camera static ip, exam end flag, and the camera fps. You can edit or delete the information by pressing the edit or delete buttons. In case you need to create a exam instance, click on create an option on the top bar and enter the required information.



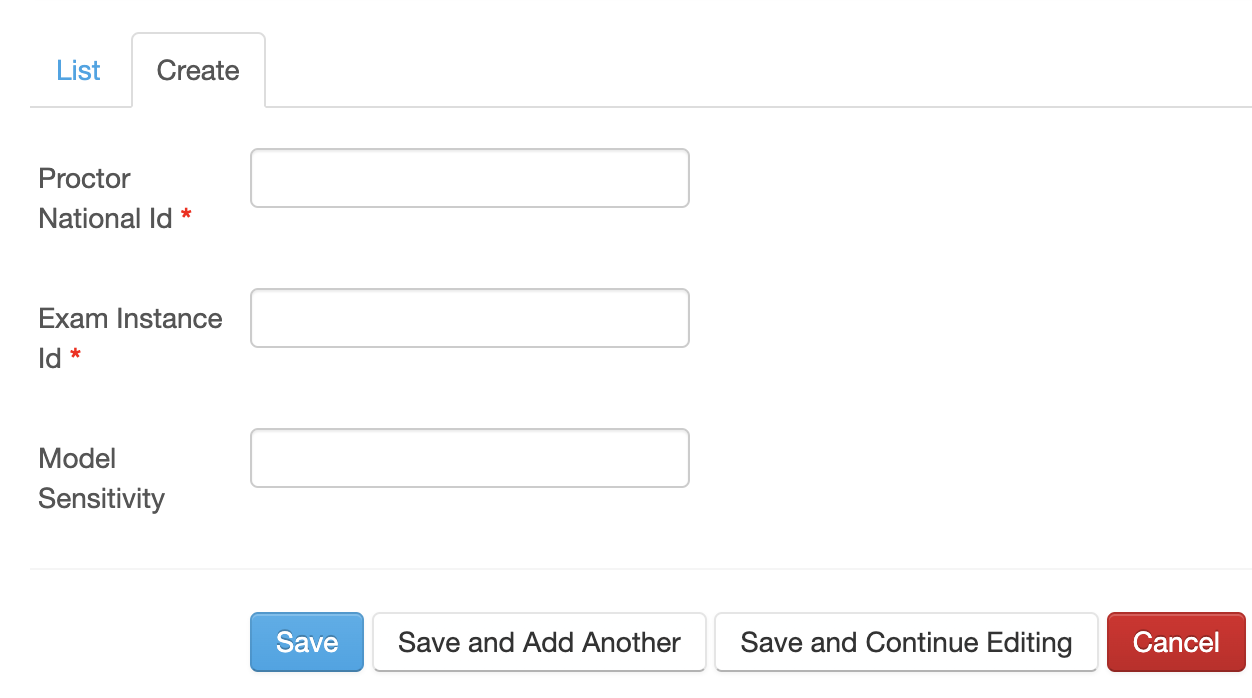
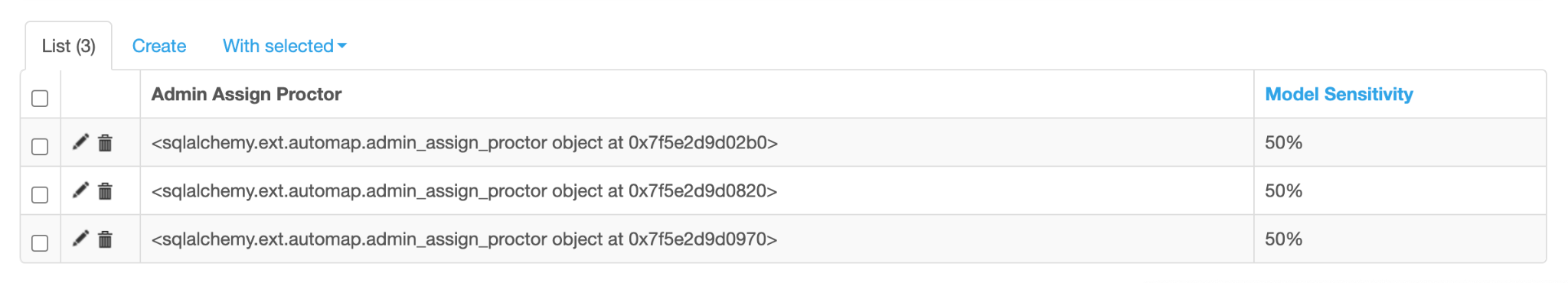
Step 6:

Click on the admin\_assign\_proctor tab to view the information of the assigned proctors and admins. The information includes the exam instance id, admin, proctor, and time stamp. You can edit or delete the information by pressing the edit or delete buttons. In case you need to assign a proctor to the exam instance, click on create an option on the top bar and enter the required information.



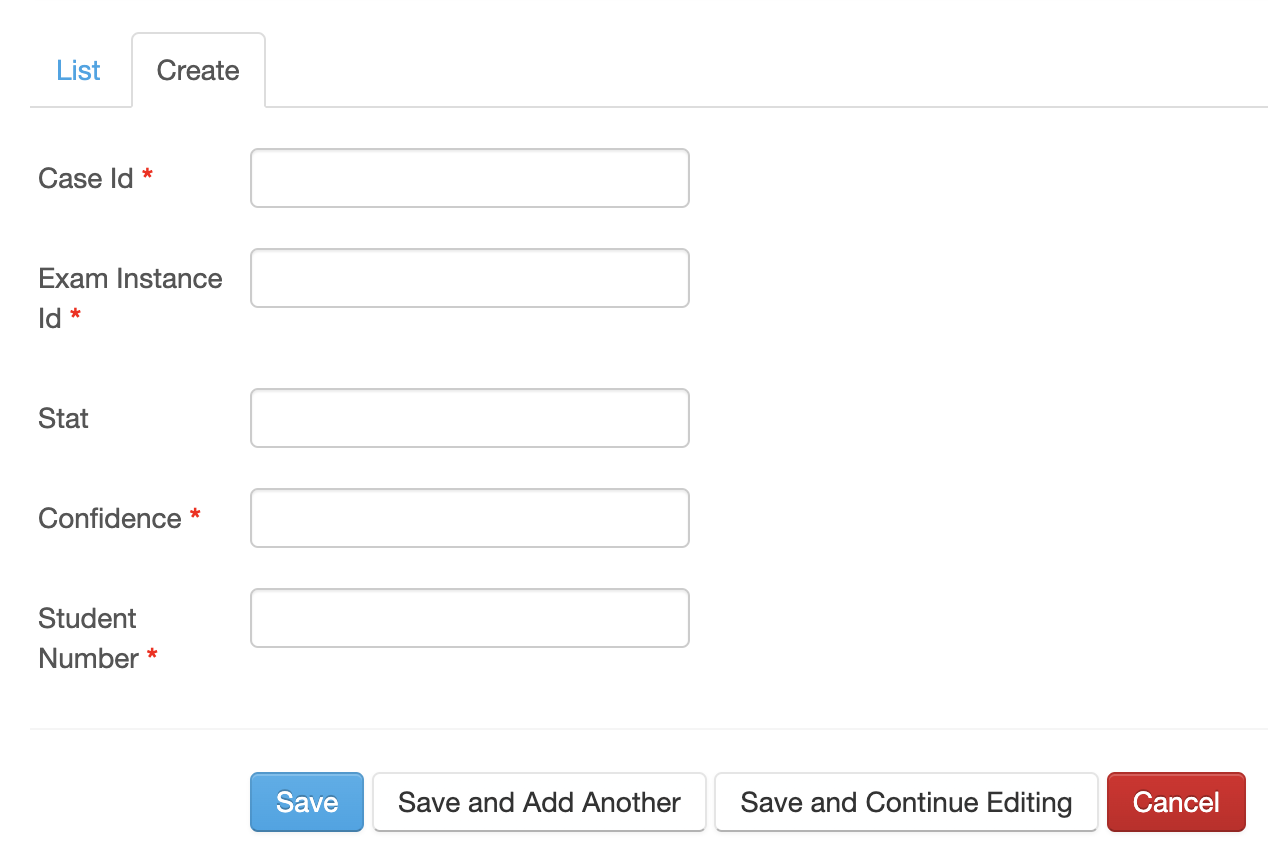
Step 7:

Click on the proctor\_monitor\_exam tab to view the information of the proctors and their assigned model sensitivity. The information includes the assigned proctor by admin and the assigned model sensitivity. You can edit or delete the information by pressing the edit or delete buttons. In case you need to assign model sensitivity, click on create an option on the top bar and enter the required information.



Step 8:

Click on the exam\_instance\_cases tab to view the information of the cheating cases. The information includes the exam instance, case ID, case status, confidence, student number and time stamp. You can edit or delete the information by pressing the edit or delete buttons. In case you need to create a case, click on create an option on the top bar and enter the required information.



* ***Business Model:***

