

**A Secure Online Examination System**

**Submitted by: Cheong Chin Kai**

**Matriculation Number: U2121198H**

**Supervisor: A/P Chua Hock Chuan   
Examiner: A/P Yakoob Siyal**

### School of Electrical & Electronic Engineering

A final year project report presented to the Nanyang Technological University

in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

**2024**

# Table of Contents

[Abstract i](#_Toc163788382)

[Acknowledgments ii](#_Toc163788383)

[Acronyms iii](#_Toc163788384)

[List of Figures iv](#_Toc163788385)

[“Chapter 1” Introduction 1](#_Toc163788386)

[1.1 Background 1](#_Toc163788387)

[1.2 Motivation 2](#_Toc163788388)

[1.3 Objectives and Scope 3](#_Toc163788389)

[1.4 Accomplishment 3](#_Toc163788390)

[“Chapter 2” Literature Review 5](#_Toc163788391)

[2.1 Research of Existing Online Examination Systems 5](#_Toc163788392)

[2.1.1 Talview 6](#_Toc163788393)

[2.1.2 Synap 8](#_Toc163788394)

[2.1.3 Smowl Tech 9](#_Toc163788395)

[2.1.4 Proctoring Platform 11](#_Toc163788396)

[2.2 Summary of Literature Review 13](#_Toc163788397)

[“Chapter 3” System Overview 15](#_Toc163788398)

[3.1 Design 16](#_Toc163788401)

[3.1.1 Use Case Diagram 16](#_Toc163788402)

[3.1.2 Wireframe 19](#_Toc163788403)

[“Chapter 4” Implementation 27](#_Toc163788404)

[4.1 React Components 27](#_Toc163788405)

[4.1.2 React Routes 28](#_Toc163788406)

[4.1.3 Backend Firebase 29](#_Toc163788407)

[4.2 Implementation 30](#_Toc163788408)

[4.2.1 Login 30](#_Toc163788409)

[4.2.2 Instructor Dashboard 31](#_Toc163788410)

[4.2.3 Interactive Calendar 33](#_Toc163788411)

[4.2.4 Exam Library 36](#_Toc163788412)

[4.2.5 Exams Set Up 38](#_Toc163788413)

[4.2.6 View Exams 45](#_Toc163788414)

[4.2.7 Proctoring Session 48](#_Toc163788415)

[4.2.8 Contact Admin 56](#_Toc163788416)

[“Chapter 5” Conclusions and Future Work 59](#_Toc163788417)

[5.1 Conclusions 59](#_Toc163788418)

[5.2 Recommendation for Future Work 60](#_Toc163788419)

[“Chapter 6” Reflection on Learning Outcome Attainment 61](#_Toc163788420)

[References 64](#_Toc163788421)

# Abstract

In today’s rapidly advanced technology, the integration of digital tools has slowly been integrated into various aspects of our lives. Educational institutions are progressively incorporating digital tools into their teaching methods. With the COVID-19 epidemic, the shift to digital platforms has increased even more since it gives students access to course materials from any location with an internet connection.

As a result, online exams are also administered. Online exams offer flexibility, allowing students to take exams at their convenience. However, this transition introduces challenges, especially regarding educational integrity as there is no physical invigilator to oversee the students.

This project focuses on developing a secure online examination platform by featuring live proctoring capabilities for administrators, instructors, and students. My role will primarily focus on the instructor’s perspective. The role of an instructor is to create, modify, and manage exams seamlessly. Additionally, instructors have a platform to communicate with administrators to address any technical issues or concerns. This collaborative approach ensures that the platform remains adaptable and responsive to the evolving needs of educational institutions.

By integrating the functionality of all three roles, we have successfully developed a comprehensive online secure examination. This platform facilitates remote examination for numerous students, ensuring flexibility and accessibility without compromising the integrity of the assessment process. Additionally, instructors can remotely oversee students during exams, thereby enhancing the security and reliability of the assessment process.

# Acknowledgments

I would like to express my deepest gratitude to Associate Professor Chua Hock Chuan, Final Year Project Supervisor, for his guidance and unwavering support throughout this journey. His insightful feedback and encouragement for our biweekly meetings have greatly contributed to the success of this project.

In addition, I would like to acknowledge the contribution of my group members who participated wholeheartedly. Through collaborative efforts, we tackled the challenges through discussion, brainstorming sessions, and peer tutoring which provided valuable insights and perspectives. I am grateful for the opportunity to work with a talented and motivated team.

March 2024

# Acronyms

|  |  |
| --- | --- |
| AI  ID  JS  SDK  UID  UML  URL | Artificial Intelligence  Identification  JavaScript  Software Development Kit  Unique Identifier  Unified Modeling Language  Uniform Resource Locator |

# List of Figures

Figure 1: A demo view of Talview Application [5]

Figure 2: Use of Secondary Camera [5]

Figure 3: System flagged out suspicious activities [5]

Figure 4: Unable to access exams from non-approved IP address

Figure 5: AI detecting unauthorized students attempting to log in. [8]

Figure 6: Background activities on the device being flagged out [8]

Figure 7: Zegocloud Platform and Documentation

Figure 8: Use Case Diagram

Figure 9: Login Page  
Figure 10: Instructor Dashboard

Figure 11: Group or Module Group List

Figure 12: Library Page to View All Exams

Figure 13: Exams Set Up or View Existing Exams (Figure 18)

Figure 14: Exam Set Up

Figure 15: Set Up Exam Questions

Figure 16: Exam Configuration

Figure 17: Set Exam Details

Figure 18: Module Exam View

Figure 19: List Of Student Attempt the Test

Figure 20: Proctor Page With No Examination

Figure 21: Proctor Page With Examination

Figure 22: View Student’s Activity

Figure 23: Send Message to Administrator

Figure 24: List of Components

Figure 25: React Routes

Figure 26: Firebase Set Up Guide

Figure 27: Firebase Set Up for Connection

Figure 28: Login Page

Figure 29: Error Modal Upon Unsuccessful Login

Figure 30: Successful Login Instructor Page

Figure 31: Tutorial Group Tab

Figure 32: Notifications from the Administrator

Figure 33: Firebase’s Exam collection

Figure 34: Navigate to Destinated Site Based on Instructor Selection Date

Figure 35: Module List on Display

Figure 36: List of all the Examinations

Figure 37: Modal Showing Examination Questions

Figure 38: Exam Page

Figure 39: Set Exam Page

Figure 40: Question Set Error

Figure 41: Error Message for Incomplete Fields

Figure 42: Calendar Restriction on Past and Present Dates

Figure 43: Modal Notification Upon Successful Set-Up

Figure 44: uuidv4 and Storing data to “exams” collection

Figure 45: Firebase Console “exams” collection

Figure 46: Firebase Console “questions” collection

Figure 47: “View Exam” tab

Figure 48: Modal Displaying the Questions

Figure 49: Managing Examination Question

Figure 50: Prompt Instructor on Deleting

Figure 51: Proctor site showing a list of upcoming examinations

Figure 52: A Proctoring Session

Figure 53: Zegocloud Video Code Settings

Figure 54: Live Proctoring Session

Figure 55: Reporting Platform

Figure 56: A “Cheating” Collection on Firebase Console

Figure 57: Contact Admin Page

Figure 58: Firebase Storage

Figure 59: Firebase Documentation on Uploading Files

Figure 60: “Message” collection displaying a message from the instructor

Figure 61: A File Uploaded to Firebase Storage

# List of Tables

Table 1: Software Components

# “Chapter 1” Introduction

## Background

With the evolution of technology in this expeditious world, digital tools have been integrated into part of our lives. Educational establishments are transitioning to virtual learning as one of the preferred methods of learning. This allows students to access materials at any time and from any location so long as they have an internet connection.

Exams play a crucial role in helping students understand their strengths and weaknesses, while also helping teachers to identify and support students who are struggling.

There are two types of examinations, traditional pen-and-paper and online examinations. Traditional pen-and-paper examinations require students to be present at a specific location and time. During the exam period, students will use physical question booklets, answer sheets, and writing materials, while being directly supervised by the invigilator.

In contrast, online exams are conducted over the Internet, enabling students to take tests virtually using a computer or electronic device. This eliminates the need for physical travel, which is more convenient.

Online exams have more advantages as compared to traditional pen-and-paper, such as eliminating logistical issues involving the arrangement of exam venues, scheduling and manpower for invigilating. Not only that, online exams are highly scalable as they can support a large pool of users and hold a huge amount of content. This explains why taking examinations online has become so popular [1].

With the Covid-19 pandemic, all schools around the globe are forced to shut down. This has prompted schools to search for workable substitutes to replace the traditional pen-and-paper approach. Therefore, online exams have been picked as one of the most practical and preferable options for evaluating students. This has caused a surge in demand for more online examination systems to be developed [2].

The primary objective of enabling students to take online exams has been achieved. However, it poses a new challenge, the absence of on-site invigilators supervising during the exam has unintentionally created an environment that encourages cheating and dishonesty.

## Motivation

With the absence of on-site invigilators, students have been using additional tools to gain unfair advantages. Not only does this compromise the academic assessments but also affects the credibility of educational institutions.

There is an interview conducted by TODAY, highlighted that 7 out of 13 recent graduates admitted to cheating or assisting a cheater [3]. Even though universities and existing software have put up security measures such as the use of student webcams and background noise detection, some students were still able to find loopholes and gaps such as the use of additional devices and accessing materials like notes and textbooks [4].

Hence, the motivation to develop a platform that facilitates the smooth administration of exams and ensures fairness in the assessment process through live proctoring, allowing instructors to monitor students' activities closely.

## Objectives and Scope

This project aims to develop a secure and reliable online examination that features live proctoring to increase efficiency and enhance the security of the system, thus, minimizing the likelihood of cheating.

The Online Examination will have three domains which are, Administrators, Instructors, and Students. The Administrators and Students domain will be developed and covered by my team members.

This report mainly focusses on the instructor’s user interface system. The instructor can create, update and delete exam questions, view all examination records, and also proctor the students during the live examinations.

## Accomplishment

Through this project, the completion of the instructor user interface marks a significant milestone. To meet the outlined objectives and scope, the functionalities for instructors are seamlessly integrated into the interface using React.js. The dynamic web application allows instructors to set up, modify, and manage examinations with ease.

One of the key features implemented is the user-friendly calendar displayed on the main dashboard upon instructor login. This intuitive tool provides a comprehensive overview of the month’s schedule, enabling instructors to stay informed about upcoming examinations and plan accordingly.

The proctoring site allow instructors to supervise students taking the examinations via their Webcam. On top of that, the report function is also available for reporting any instances of dishonest behavior. The instructor interface also includes a feedback channel for instructors to communicate with administrators, facilitating prompt resolution of any issues or concerns.

# “Chapter 2” Literature Review

## 2.1 Research of Existing Online Examination Systems

To date, numerous online examination systems have been developed and implemented by institutions around the globe. While there are several variations among these systems, they all share a common objective: to minimize the likelihood of dishonest acts during online examinations.

This section will explore case studies of three existing online examination systems to identify their common key features, advantages, and disadvantages, providing valuable insights into their overall suitability for institutions.

## 2.1.1 Talview

An online examination system that has a user-friendly interface to provide a hassle-free experience for both instructors and students. Live proctoring enables instructors to identify and address any unusual activities promptly and also real-time monitoring of the student’s screen and activities shown in Figure 1.

A screenshot of a video chat

Description automatically generated

Figure 1: A demo view of Talview Application [5]

In terms of user authentication, Talview integrates multi-factor authentication to ensure the legitimacy of users participating in exams. These include the use of features such as Photo ID and voice matching features to authenticate that the student is the actual person taking the exam.

One of the prominent features is the use of the secondary camera for proctoring purposes. The secondary camera is equipped with an advanced algorithm that can identify multiple individuals within a room during the online exam. This feature helps to prevent unauthorized individuals from participating in the exam [5].

The secondary camera’s capabilities are not limited to just identification but are also designed to capture multiple angles within the examination environment. This allows instructors to monitor and address any signs of suspicious behavior that may arise during the examination.

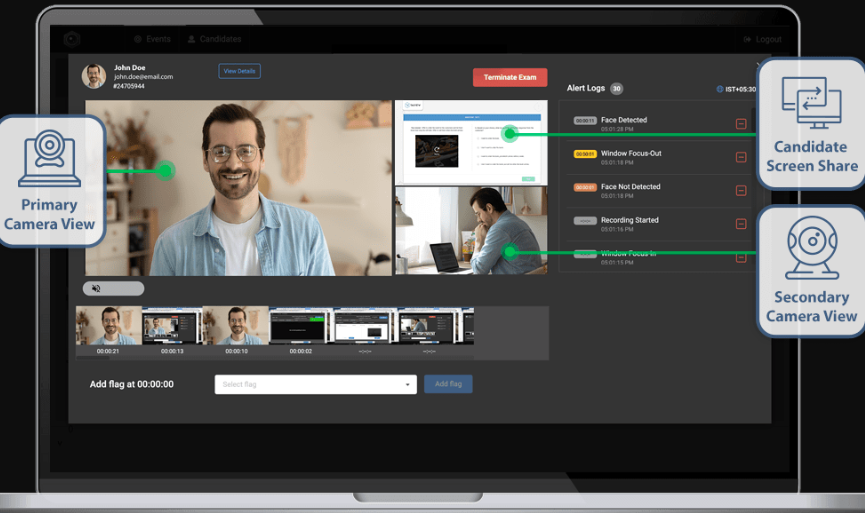


Figure 2: Use of Secondary Camera[5]

Additionally, Talview offers its secure lockdown browser, which plays a crucial role in minimizing malpractice by limiting web browser functionality and access. By restricting additional browser, tab, or application switching, the lockdown browser provides a more controlled and secure examination environment, thereby reducing the chances of cheating or unauthorized access. [6]

## 2.1.2 Synap

A web-based platform that employs a range of security measures to ensure the integrity of exams. This includes the use of 2-Factor Authentication (2FA), secure data transmission, screen recording, flagging system, and in-depth review. [7]

A screenshot of a computer

Description automatically generated

Figure 3: System flagged out suspicious activities [5]

A dialog box will pop up if a user opens additional tabs as shown in Figure 3. To increase exam security even further, the application disables the ability for students to use keyboard shortcuts to copy questions and paste answers. This reduces the chance of cheating and also makes it harder for students to leak questions.

In addition, the system randomizes the order of questions and multiple-choice options, ensuring that each user receives the exam in a unique sequence, which discourages sharing questions and answers among students. Furthermore, timed sections automatically transition students to the next section if the timer expires. Once students move to subsequent sections, previous sections become locked, preventing them from returning to earlier questions.

One of the most notable features of the web app is the use of IP address restriction which is shown in Figure 4. To prevent unauthorized access, measures are implemented to ensure that only intended students can access the exam. This eliminates the risk of exam link sharing and someone else taking the test on behalf of the intended student.

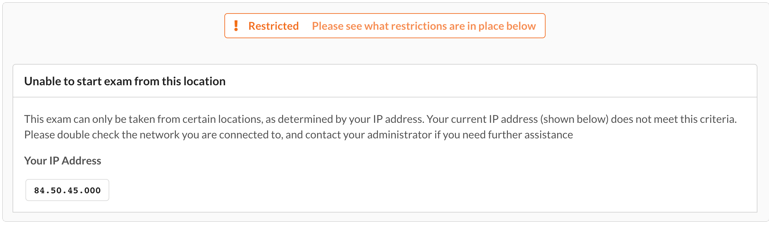


Figure 4: Unable to access exams from non-approved IP address

Furthermore, Synap offers a diverse range of question formats, customized options for exams, and robust analytics tools for performance analysis. These features not only streamline the grading process but also significantly reduce administrative workload and enhancing overall efficiency and effectiveness.

## 2.1.3 Smowl Tech

The online examination software utilizes advanced algorithms and image recognition technologies to enhance student authentication. To ease the authentication process, the software employs Artificial Intelligence (AI) to match student’s faces with a pre-existing database. This approach significantly reduces the time required for manual verification by instructors.

A screenshot of a computer

Description automatically generated

Figure 5: AI detecting unauthorized students attempting to log in. [8]

Through the use of a webcam, the AI captures images of students during tests, monitoring their position and detecting any unusual activities such as students leaving their seats, engaging in conversations or obstructing the view with objects. This surveillance allows the AI system to automatically identify and flag any irregular behavior, alerting instructors of potential breaches of exam integrity. Furthermore, background activities on the device are closely monitored, along with the live proctoring. [8]

A screenshot of a computer

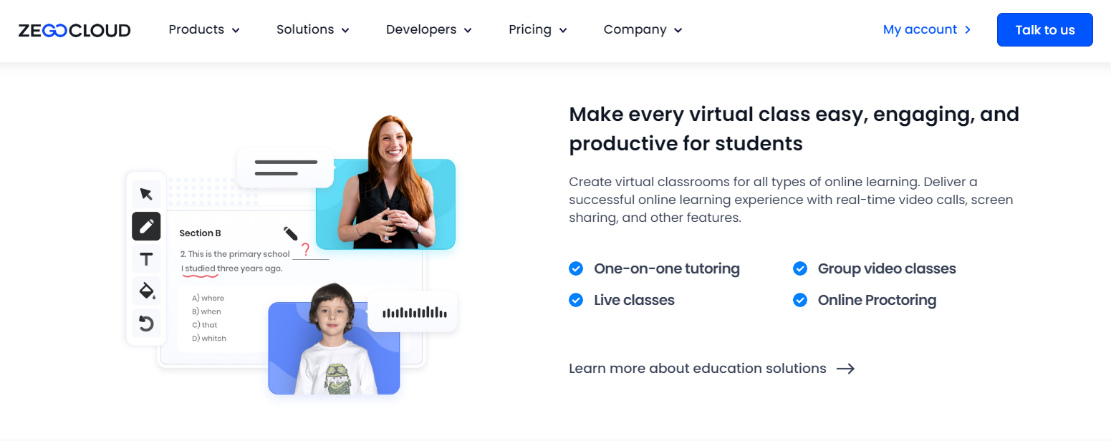
Description automatically generated

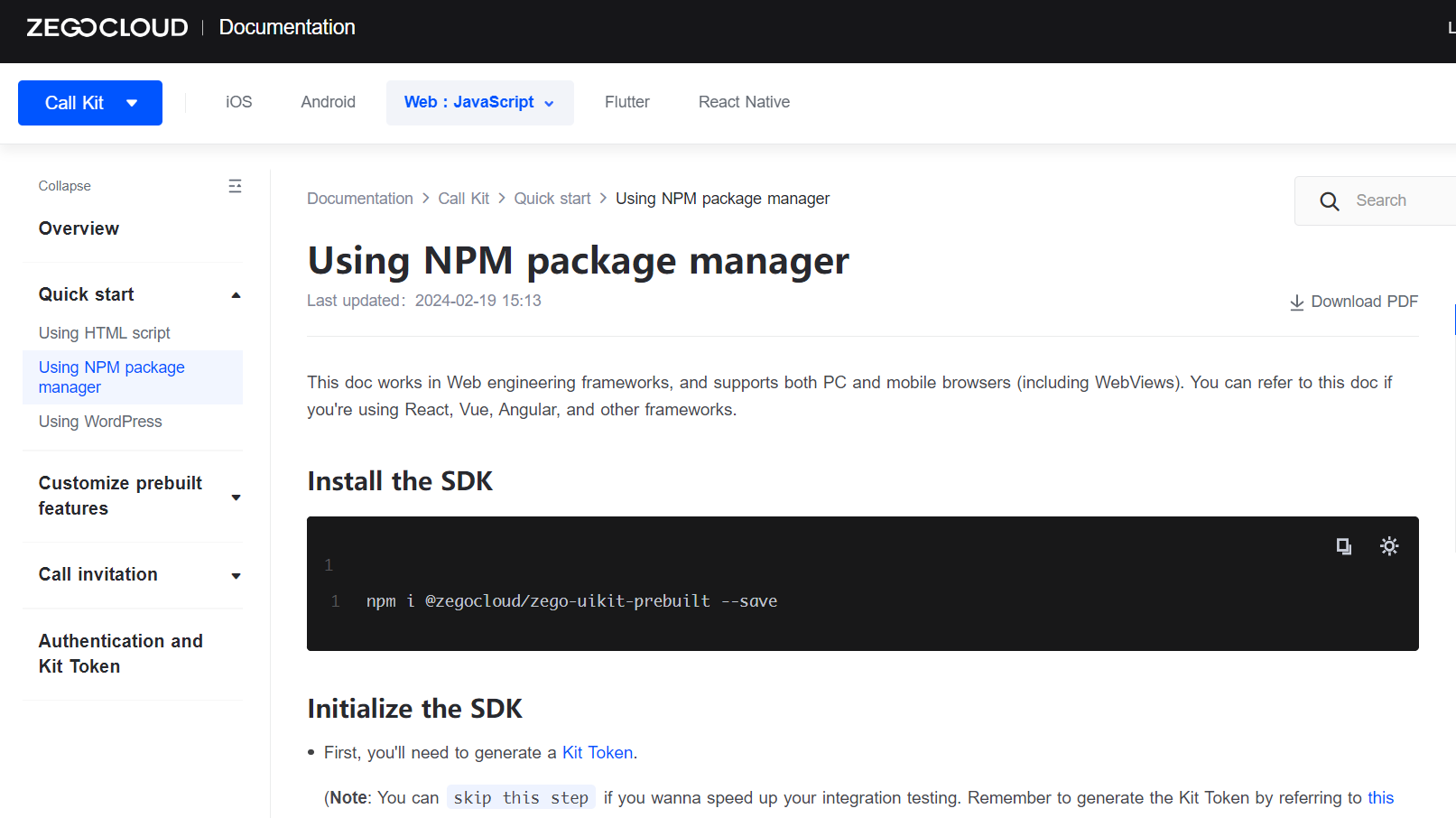
Figure 6: Background activities on the device being flagged out [8]

Instructors were given the privilege to export the student’s results and other information as a CSV file, allowing them to store the records locally for future case studies and analysis. This feature provides an extra layer of security and control over sensitive academic information against potential breaches or system failures.

## 2.1.4 Proctoring Platform

The team has decided to use Zegocloud as the platform for live proctoring. Zegocloud enables easy integration through the use of SDK-embedded high-quality real-time video [9], making it convenient to incorporate into the project system. Moreover, Zegocloud can accommodate up to 500 users [10] and provide detailed documentation to ease the implementation process and utilization of its features [11] (shown in Figure 7). This makes Zegocloud an ideal solution for conducting live proctoring sessions. Especially in situations where there is a need to monitor large groups of students.





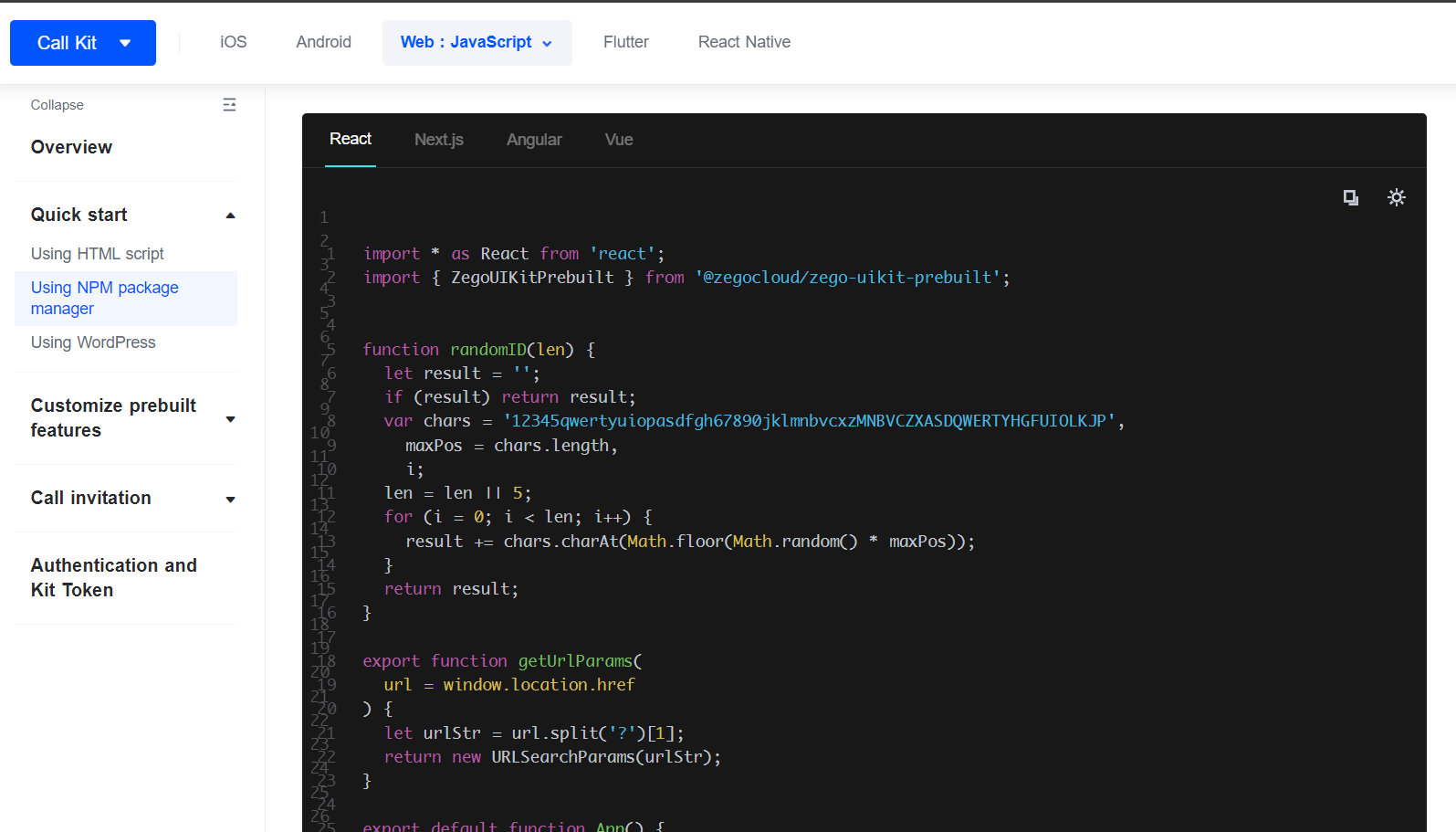


Figure 7: Zegocloud Platform and Documentation

## 2.2 Summary of Literature Review

While each of the previously mentioned online systems incorporates various features and security measures to ensure exam integrity, every system also has its limitations. These limitations are important for designing and implementing the online secure examination.

* **Technical Issues**: As a third-party software, all registration details and past proctoring records are stored in its database. It may not be convenient for instructors to access and review students’ activities. Instructors must go through the process of requesting access from the company, which may result in delays or limited availability of data when needed for review or analysis. Therefore, storing the database locally or using cloud services would be beneficial, enabling instructors and administrators to access it on the go.
* **Recording**: Several systems lack continuous monitoring throughout the entire exam duration. This limitation may pose challenges in detecting and addressing irregularities or dishonest behavior occurring between the scheduled captures. Implementing continuous monitoring could provide a more comprehensive approach to maintaining exam integrity.
* **False Positives:** At times, the AI algorithm may produce false positives or errors during student authentication, leading to disruptions in the exam process or unnecessary alerts for instructors. Implementing visual verification via webcam for instructors to validate student identities could enhance authentication and reduce disruption during exams.
* **Lack of Live Proctoring**: The software does not provide live proctoring capabilities, thus, limiting the ability of instructors to monitor students in real-time during exams. Without live proctoring, there is a higher chance of undetected academic dishonesty.

# “Chapter 3” System Overview

The online examination system consists of three domains: Administrator, Instructor, and Student. The administrator's role includes registering and enrolling students in their respective modules, managing cheating reports, and facilitating communication between instructors.

Instructors are responsible for exam preparations and live proctoring during examinations. Additionally, students are required to have their webcam and microphone checked to ensure they are in good working condition before taking exams. Below is a table which list out the software components used in the project.

|  |  |
| --- | --- |
| Component | Description |
| Frontend | React.js – Used to provide a user-friendly interface and enhance overall user experience [12]. |
| Backend | Firebase Console – A cloud computing platform developed by Google, used to manage data and ensure smooth system functionality [13]. |
| Collaboration | GitHub – Main platform for team collaboration, allowing team members to view updates, track changes and manage subsystem branches effectively [14]. |
| Development Tool | Visual Studio Code – Integrated development environment used for coding and debugging. |
| Cloud Communication | ZegoCloud – A real-time audio and video communication platform which can be integrated easily through its SDK. |
| Design Collaboration | Figma – A collaborative interface design tool that enable team to work together in real-time on interface design task. |

Table 1: Software Components



## Design

The system design process utilized a use case diagram to illustrate the various interactions between users and the system. This diagram served as a blueprint to outline the functions and user roles within the platform.

As the design progressed, it transitioned to Figma, a user-friendly design tool where a more visually appealing representation of the platform was developed. Figma enables the creation of more detailed mockups and prototypes, thus, providing an excellent visualization of the system’s interface and overall user experience [15].

### Use Case Diagram

The use case diagram enables developers to design a system from the end user’s perspective, using UML diagrams to represent the interaction between actors and a system to accomplish specific goals. This provides a high-level overview of the system making it easier for everyone involved to understand how the system works [16].

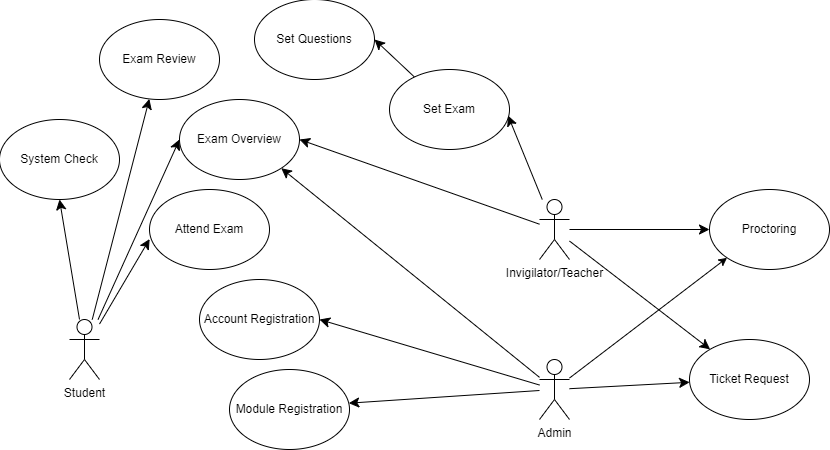


Figure 8: Use Case Diagram

In Figure 8, the primary interactions and functionalities within the three key user roles: Administrator, Instructor, and Student are showcased.

**Administrator:**

* Student Registration: The Administrator registers the student into the system.
* Student Enrollment: Registered students are then enrolled in specific modules for examination purposes.
* Communication with Instructors: Administrators communicate with instructors to ensure smooth coordination and resolution of issues.

**Instructor:**

* Prepare Exams: Instructors prepare exams by creating questions, setting up exam parameters, and scheduling exam sessions.
* Conduct Live Proctoring: Oversee live proctoring during examinations to monitor student behavior.

**Student:**

* Take Exam: Students access exams, answer questions, and submit their responses within the allotted time frame.
* System Check: Before taking exams, student will need their webcam and microphone checked to ensure proper functionality for exam proctoring.

### Wireframe

The wireframe was drafted with Figma, and each component of the interface was designed and arranged to ensure optimal usability and navigation for Instructors.

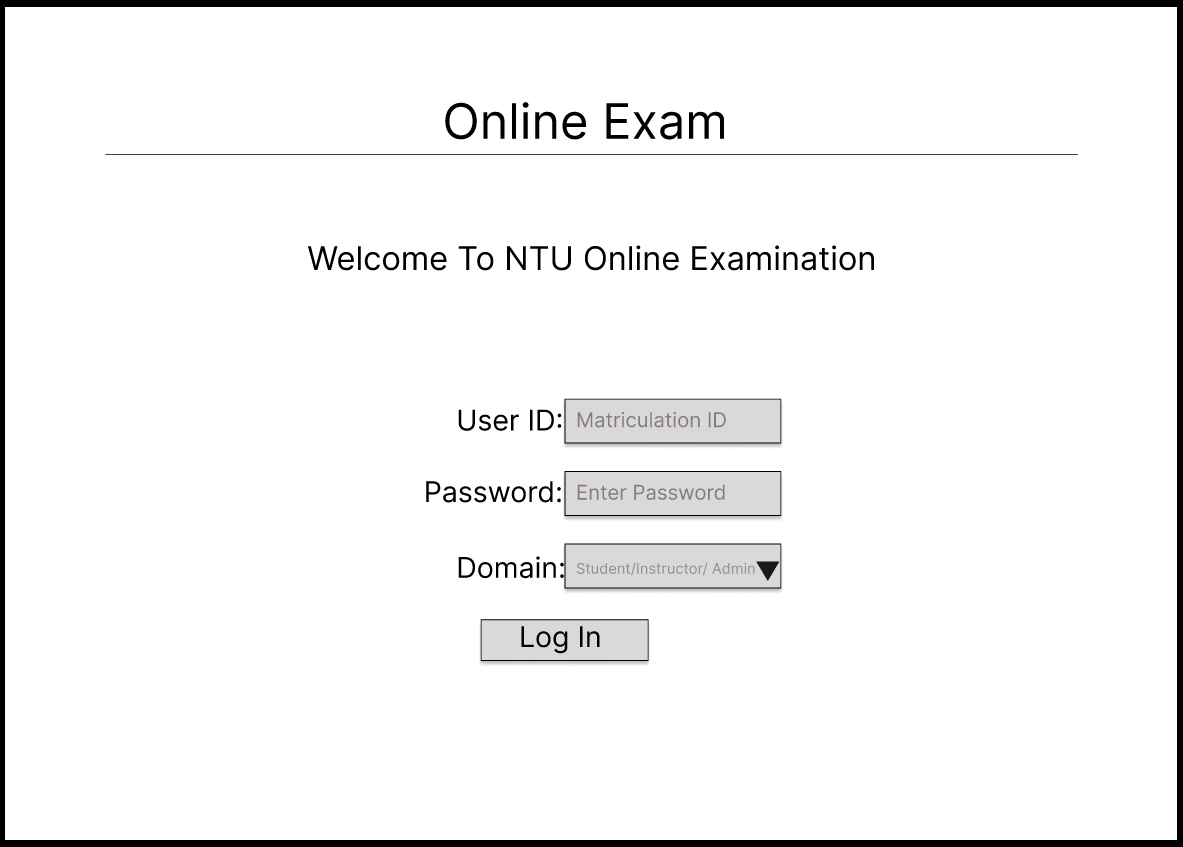


Figure 9: Login Page

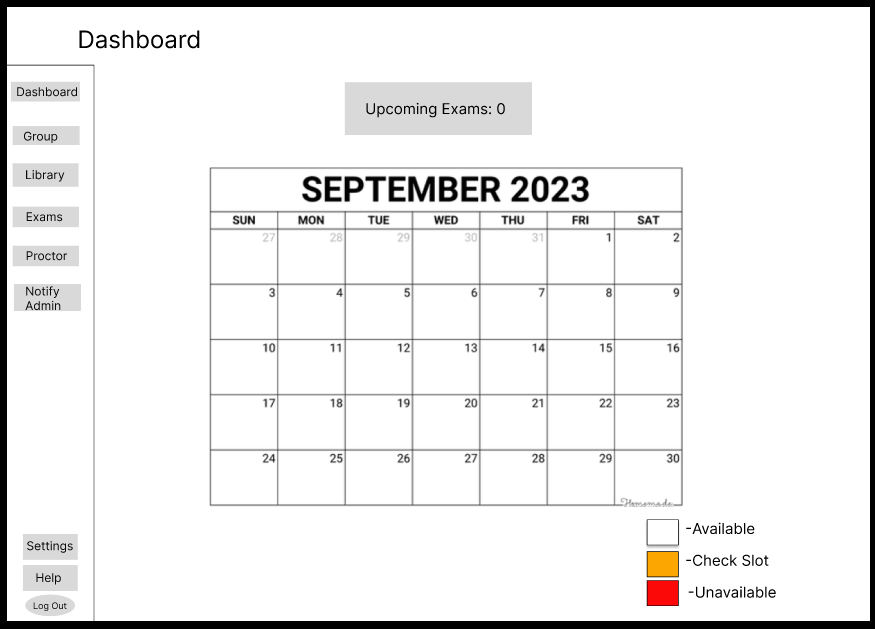


Figure 10: Instructor Dashboard



Figure 11: Group or Module Group List

In Figure 11, it shows the list of students that were enrolled in a particular module and their respective details such as Name, Matriculation Number, and Email.

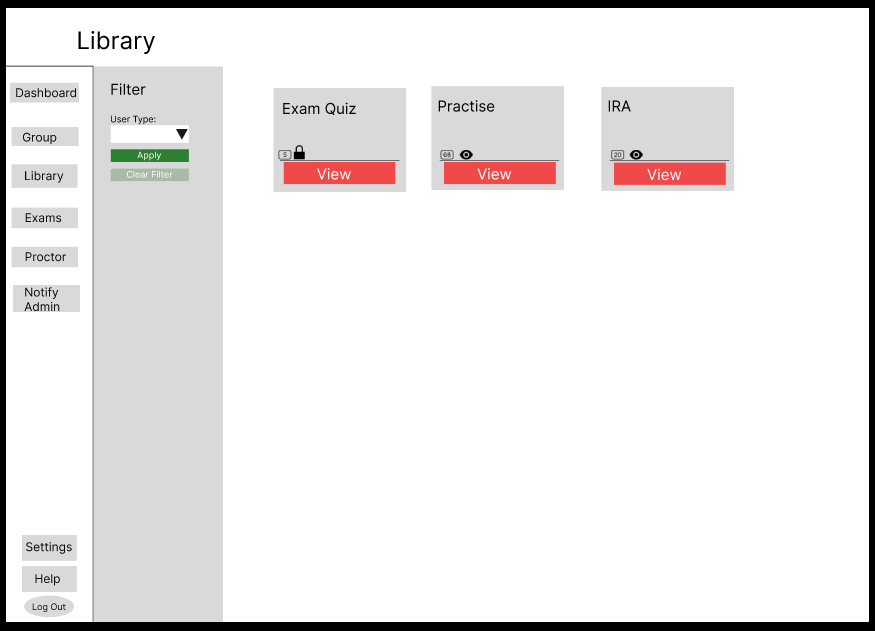


Figure 12: Library Page to View All Exams

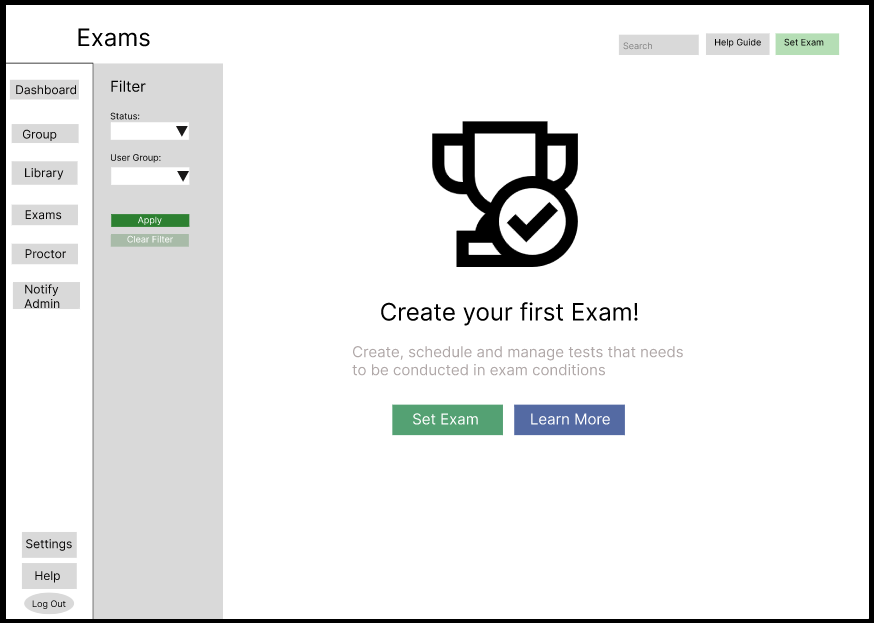


Figure 13: Exams Set Up or View Existing Exams (Figure 18)

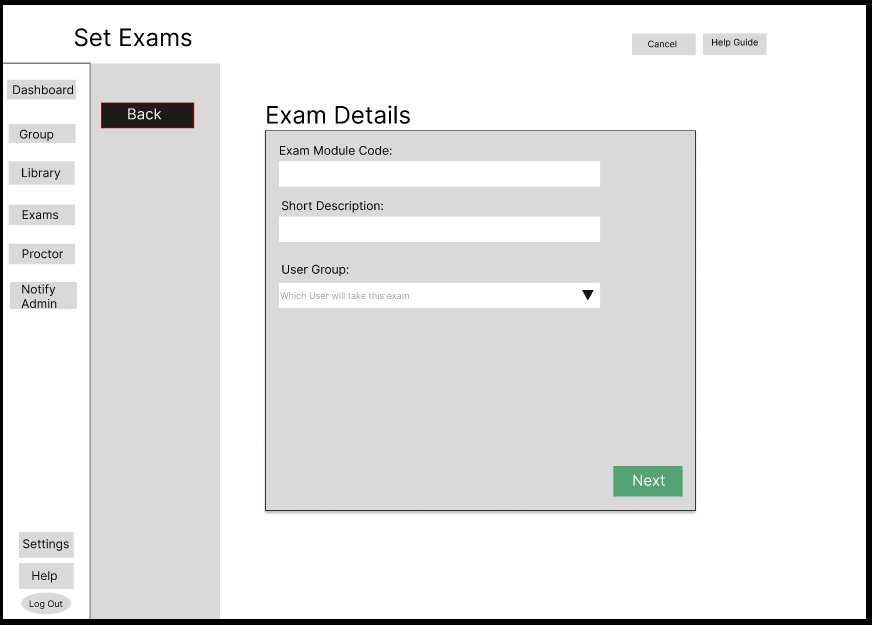


Figure 14: Exam Set Up

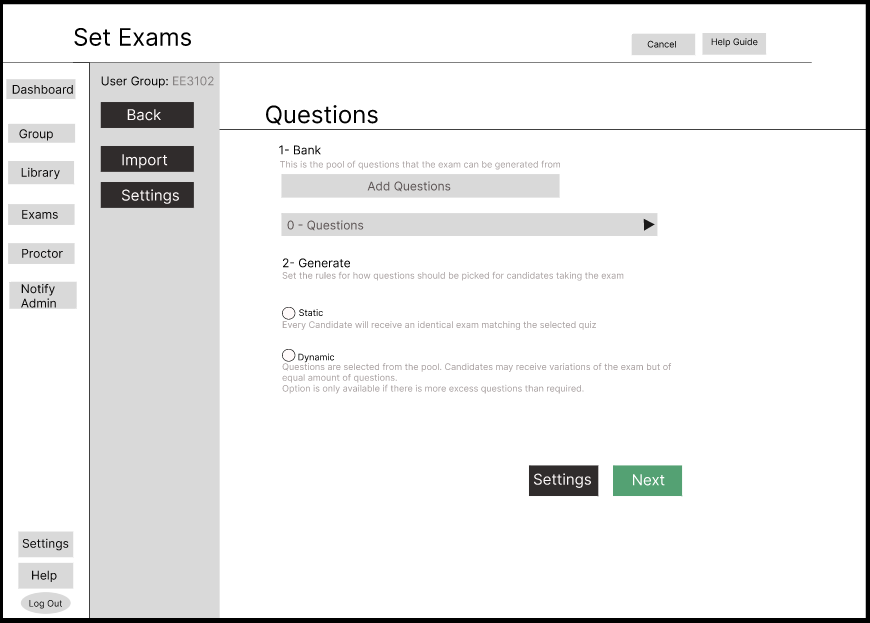


Figure 15: Set Up Exam Questions

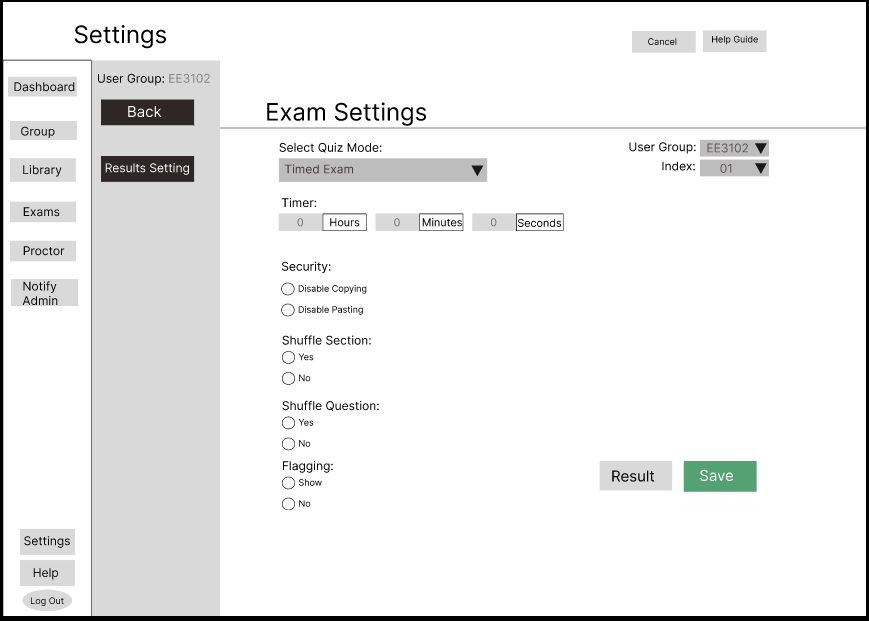


Figure 16: Exam Configuration

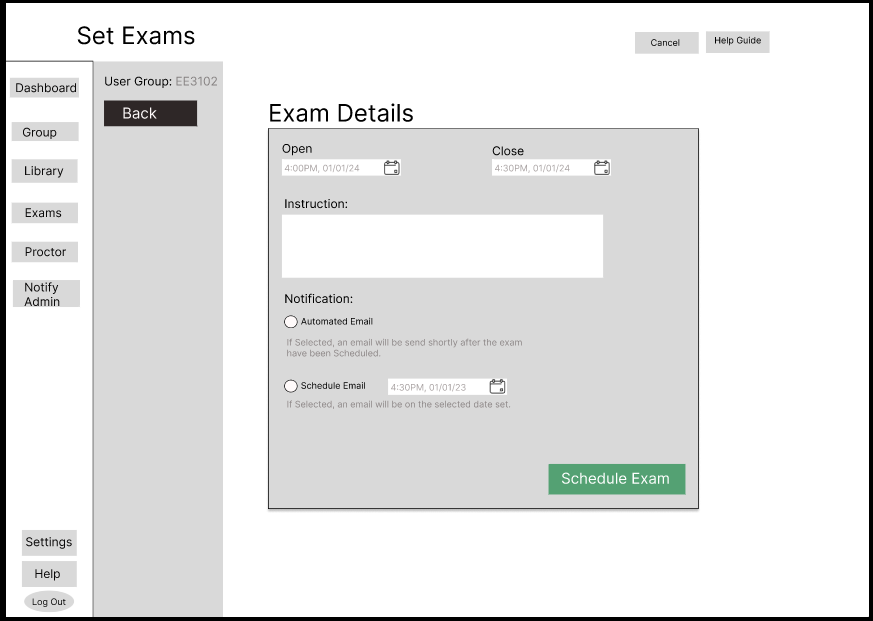


Figure 17: Set Exam Details

In Figure 17, all past and present examinations for the module are displayed. However, in the absence of an exam (default), it will be shown as depicted in Figure 13.

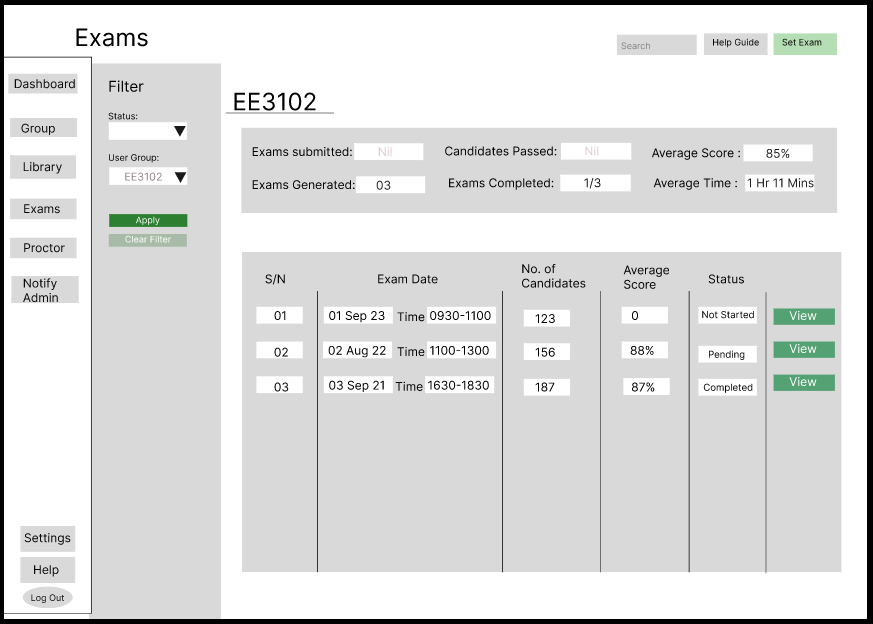


Figure 18: Module Exam View

Clicking on the “View” (Red Box) will bring the instructor to the student list. (Shown in Figure 18)

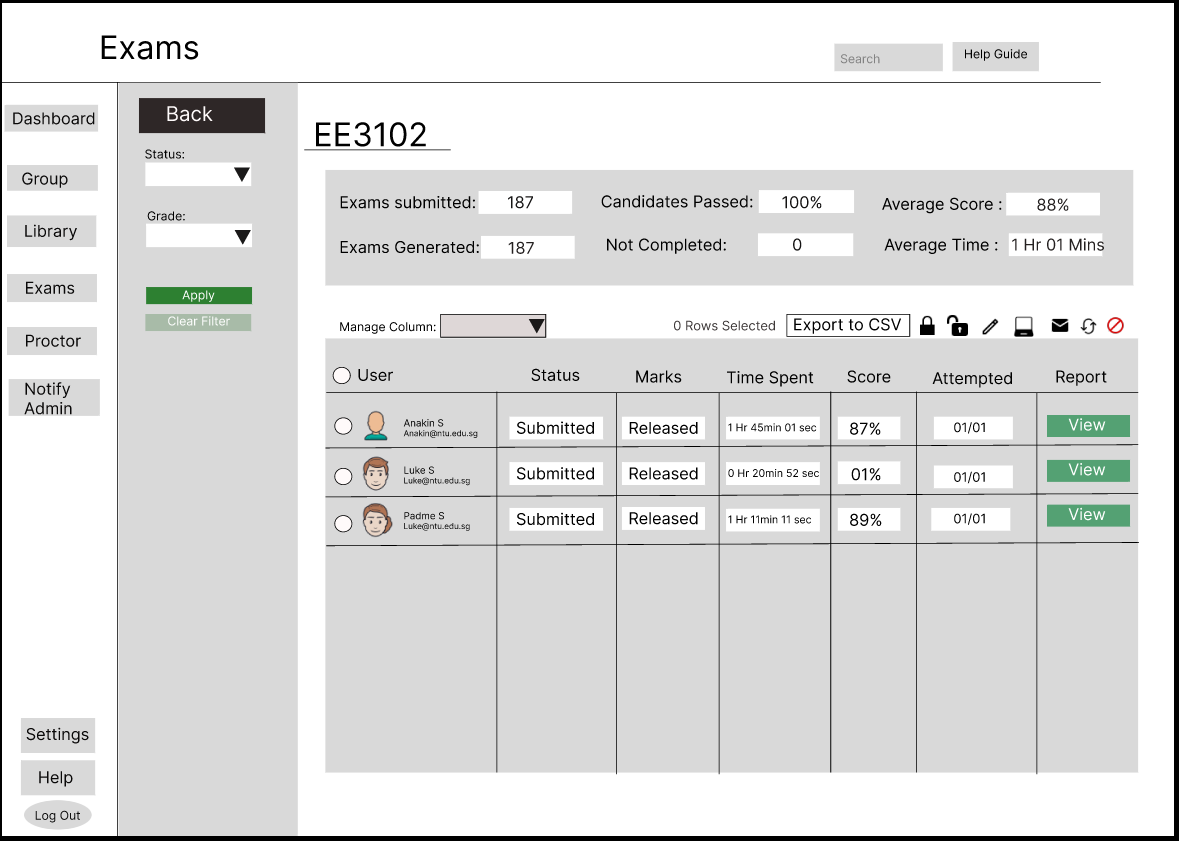


Figure 19: List Of Student Attempt The Test

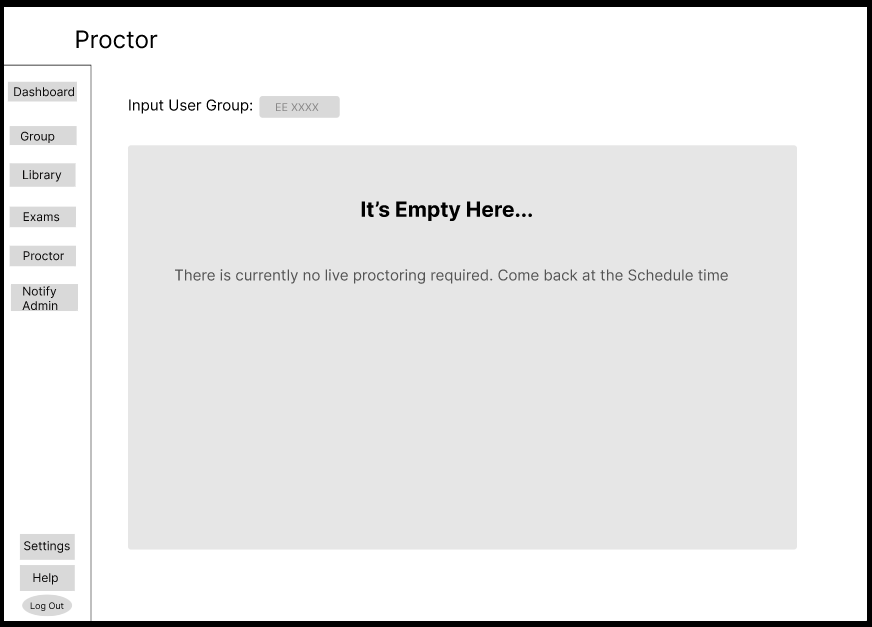


Figure 20: Proctor Page With No Examination

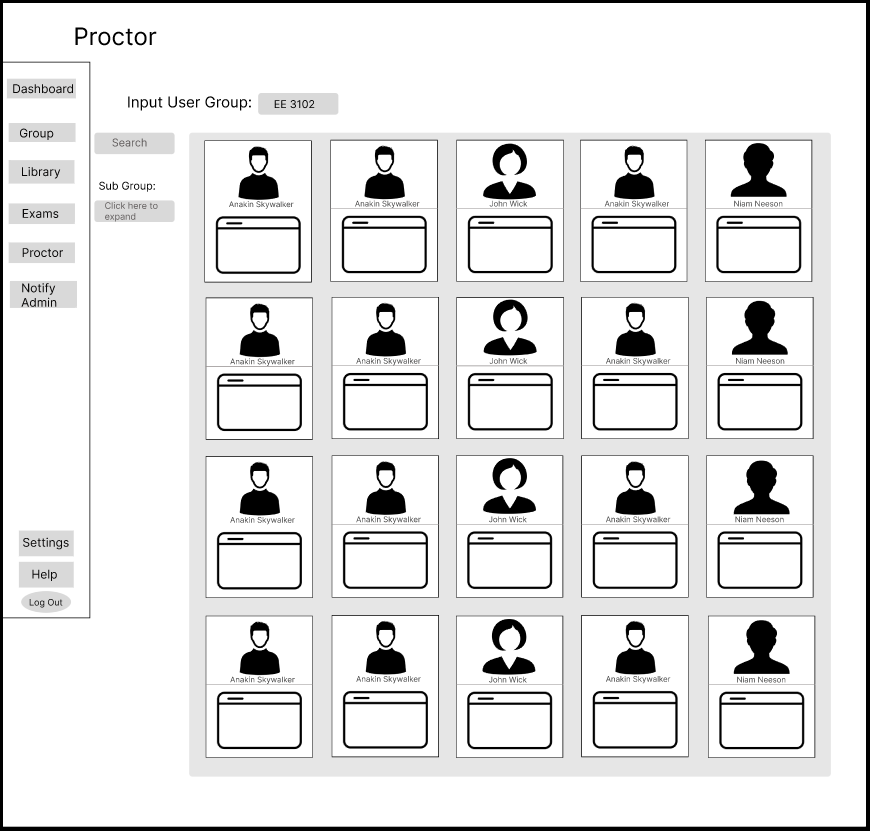


Figure 21: Proctor Page With Examination

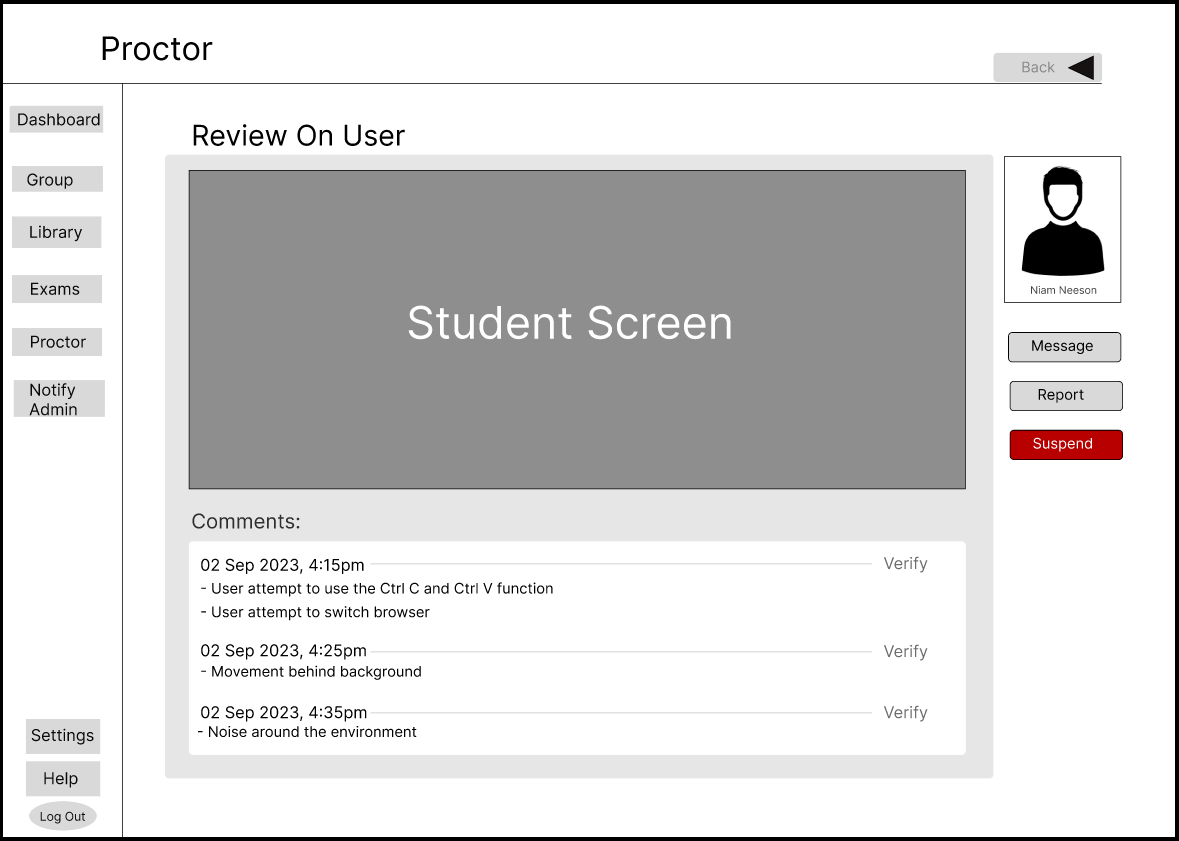


Figure 22: View Student’s Activity



Figure 23: Send Message To Administrator

# “Chapter 4” Implementation

# 4.1 React Components

Using React to create a common login interface for all three domains: Administrator, Instructor, and Student to maintain a consistent and efficient layout. The web application was developed to run locally using the Visual Studio Code. The application is started via ‘NPM start’ command and accessed in localhost: 3000.

Utilizing React’s component-based feature, components for elements such as buttons, modals, and input fields can be reused. This ensures consistency in design and functionality across the various domains.

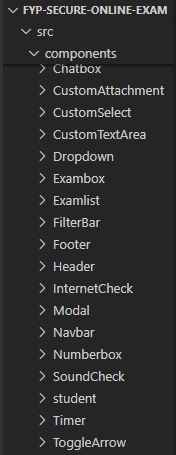


Figure 24: List of Components

# 4.1.2 React Routes

Additionally, React Router was employed to manage navigation within the platform. Routes were defined to direct the instructors to respective pages based on the agenda.



Figure 25: React Routes

# 4.1.3 Backend Firebase

The backend usage of Firebase, which eliminates the need for server management [17]. This is due to Firebase’s cloud-based platform which operates on a serverless model.

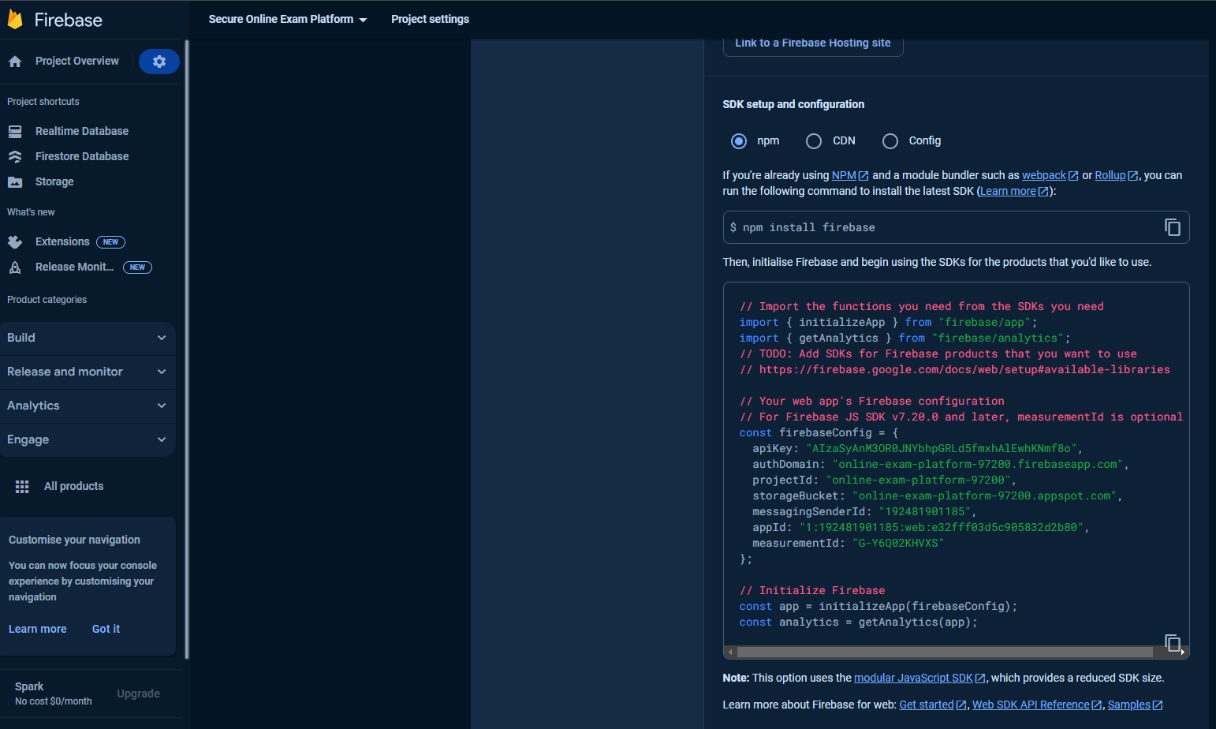


Figure 26: Firebase Set Up Guide

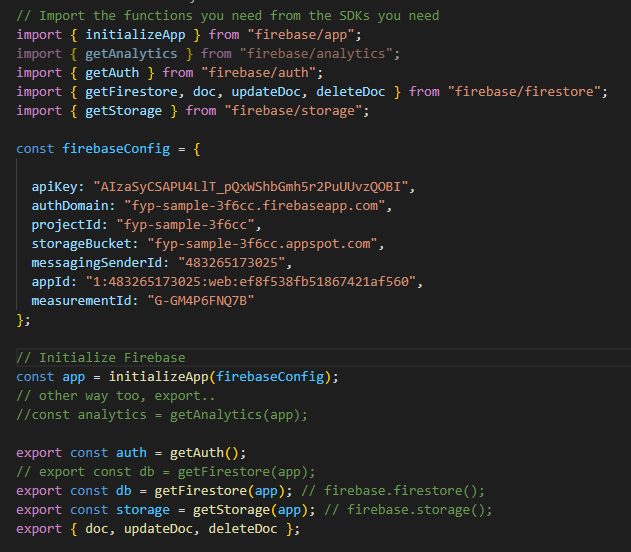


Figure 27: Firebase Set Up for Connection

# 4.2 Implementation

# 4.2.1 Login

This is a common login page designed to accommodate users across the three domains: Administrator, Instructor and Student (shown in Figure 28). The login page implements role-based access control to differentiate between users from different domains.

Students are granted access to take examinations and view profiles, while instructors and administrators have additional features such as exam, account and proctoring features.

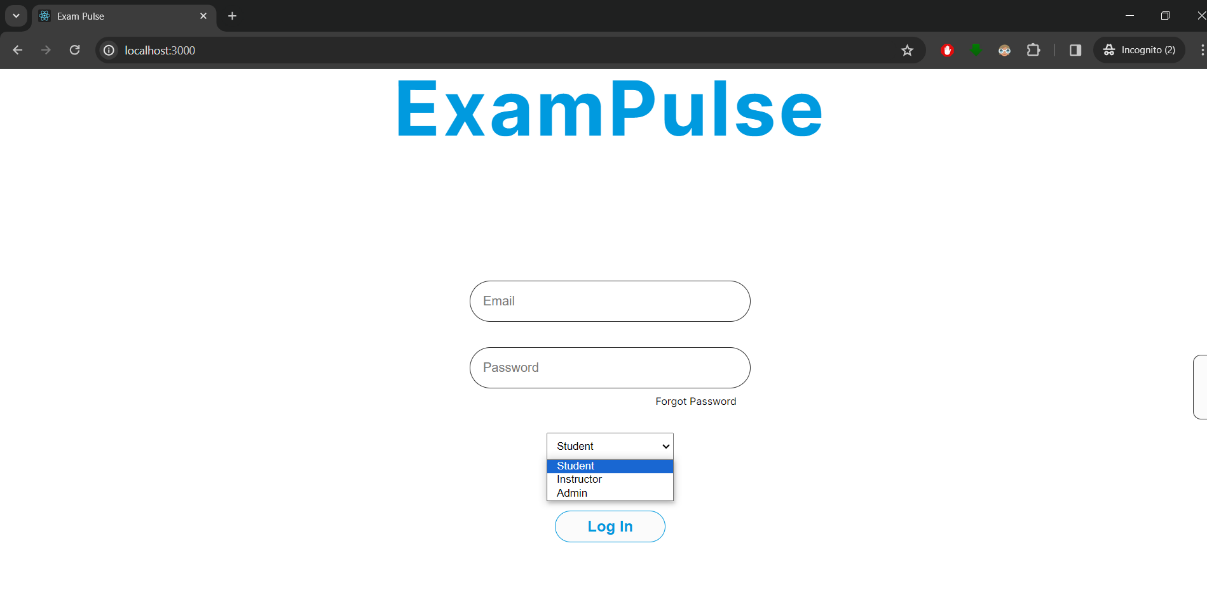


Figure 28: Login Page

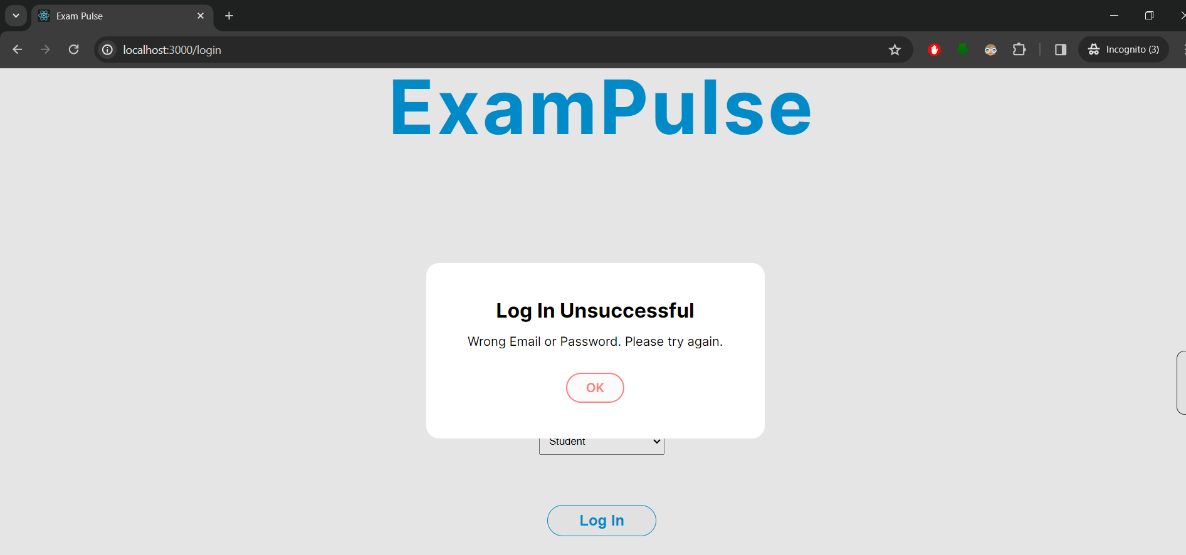


Figure 29: Error Modal Upon Unsuccessful Login

# 4.2.2 Instructor Dashboard

Upon successful login, instructors are directed to the Dashboard where they can easily navigate between different sections and access relevant information. The calendar highlights events such as past, current, or upcoming examinations from the Firebase Database. Clicking on the dates will bring the user to the relevant tabs for more information.

A screenshot of a computer

Description automatically generatedFigure 30: Successful Login Instructor Page

A screenshot of a computer

Description automatically generated

Figure 31: Module Group Tab

In figure 31, the Module Group tab will diplay a list of modules along with the current enrollment count. Upon selecting the “View Students”, it will show the names of students listed in the module.

A screenshot of a computer

Description automatically generated

Figure 32: Notifications from Adminstrator

# 4.2.3 Interactive Calendar

The following code shows the interactive calendar integrated with Firebase exam data, which displays the examination dates. Additionally, it includes a function to redirect instructors to relevant pages upon clicking the examination dates.

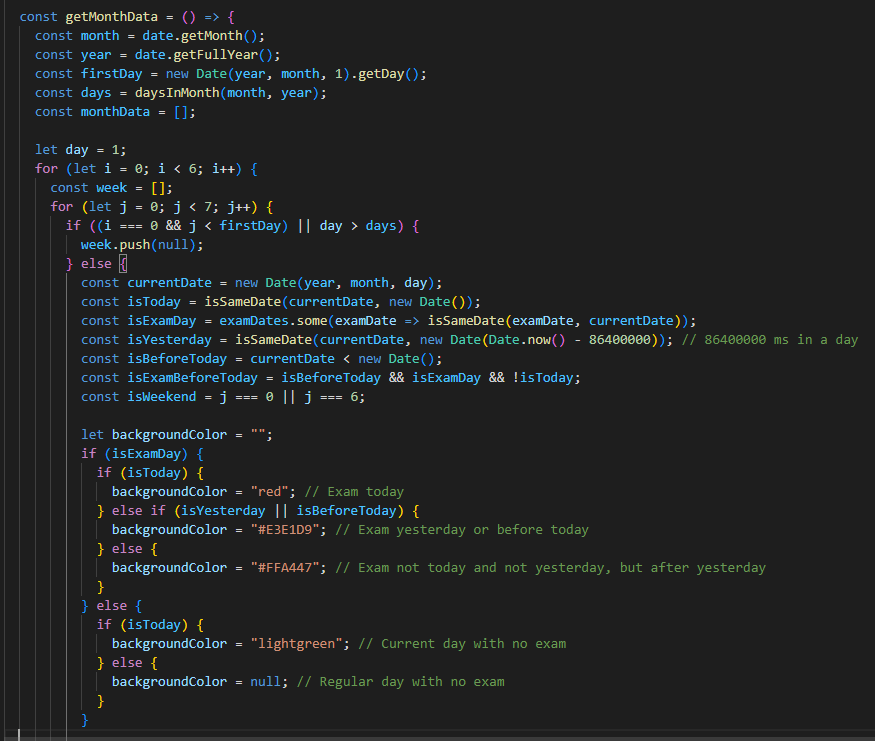


The first function, 'fetchExams', retrieves data from the Firebase collection 'exams' to query all the exam start dates from the firebase console. The firebase console exam dates is shown in Figure 33 (highlighted in the red box).

A screenshot of a computer

Description automatically generated

Figure 33: Firebase’s Exams collection



After fetching all the exam dates, the calendar is generated. Each date on the calendar is color coded to indicate whether it is a past, present or future examination date. The color classification is determined using the conditional “if-else” statement (shown in red box).

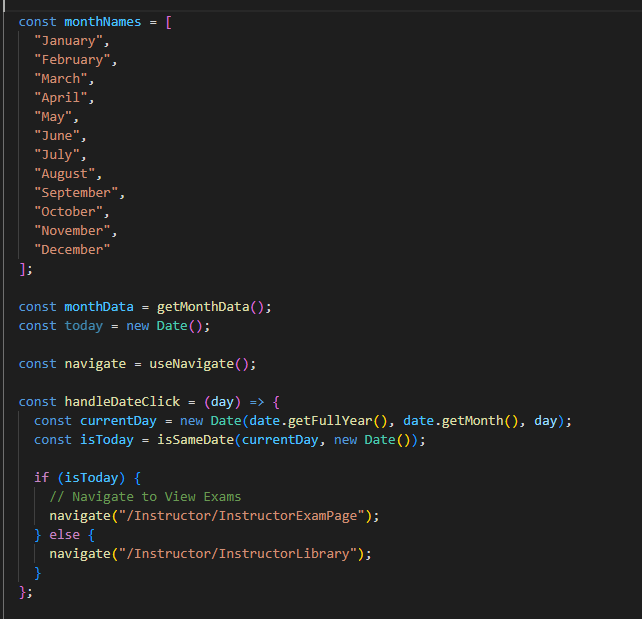


Figure 34: Navigate to Destinated Site Based on Instructor Selection Date

Furthermore, the function “handleDateClick” respond to calender clicks. If the clicked date corresponds to a past examination, instructors are redirected to the exam library page. Conversely, if it is a present or future examination date, instructors will be directed to the exams page (shown in figure 34).

# 4.2.4 Exam Library

The exam library page provides instructors an overview of all modules at a glance, along with the number of exams associated with each module indicated by the “number of exam” (shown in Figure 35, red box).

A screenshot of a computer

Description automatically generated

Figure 35: Module List on Display

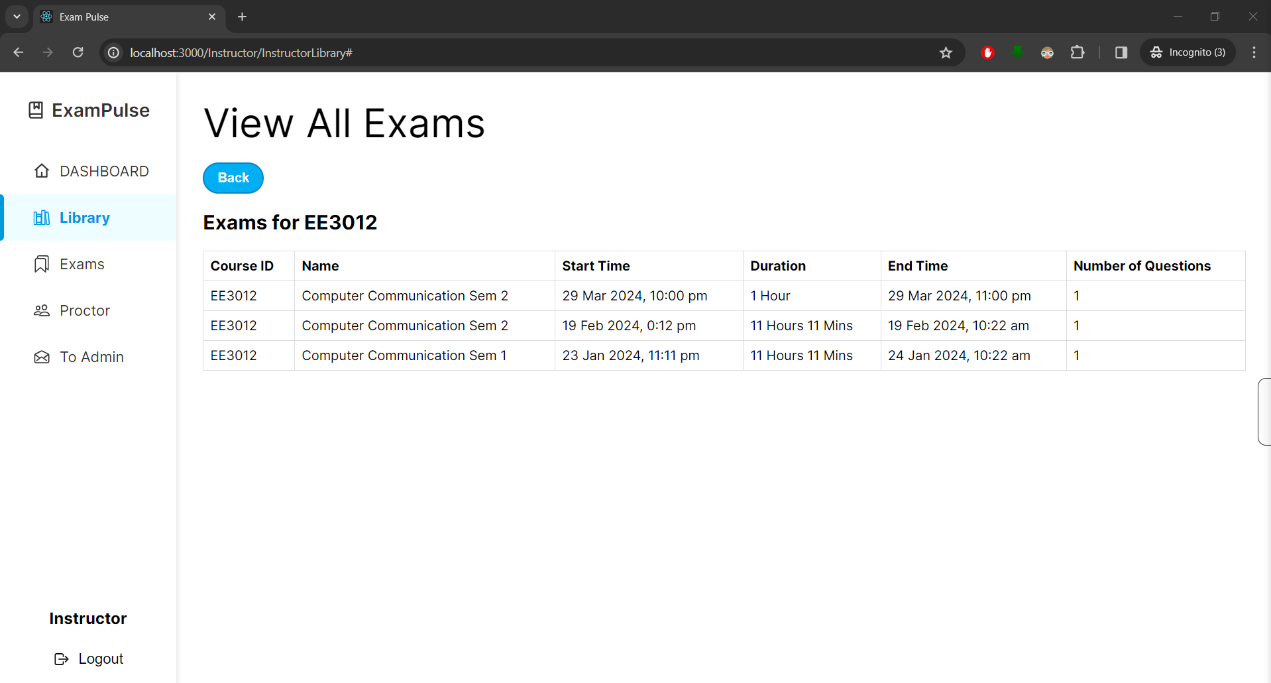


Figure 36: List of all the Examinations

By navigating to a specific module, instructors can view all past, current, and upcoming exams linked to that module (shown in figure 36).

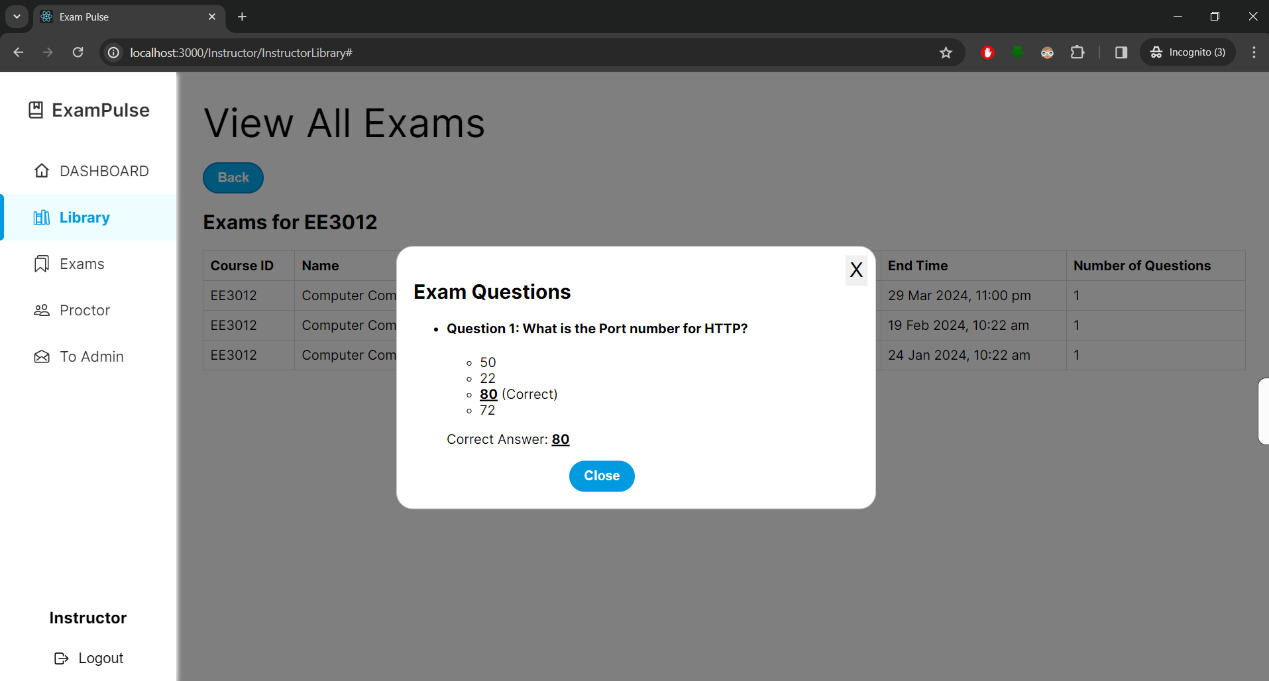


Figure 37: Modal Showing Examination Questions

When instructors select the start time, end time, or number of questions, a modal window appears (shown in Figure 37). This modal provides a detailed view of the questions set in the exams, allowing the instructors to manage and review exam contents

Instructors are only allowed to edit the questions for upcoming exams, not past examinations. Therefore, the model shown in Figure 37 will appear differently when viewed in the ‘Exams’ tab as shown in Figure 48. A detailed explanation of how each question will be tied to its respective examination will be shown in the next section (4.2.5 Exams Set up).

# 4.2.5 Exams Set Up

In the exam setup section, instructors have the flexibility to set any number of questions, along with selecting the correct answers. Details such as course ID, exam name, date, time, and duration are required, as all this information will be stored on the Firebase console. Upon successful setup, a modal with a confirmation message will be displayed.

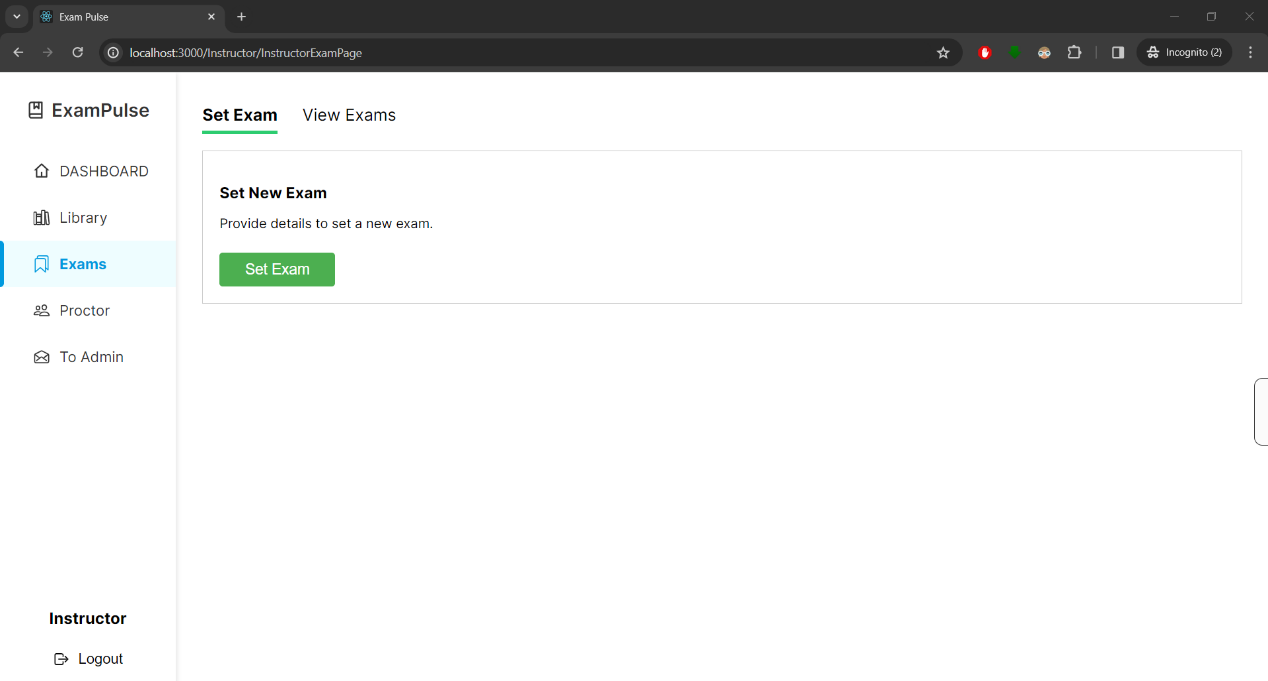


Figure 38: Exam Page

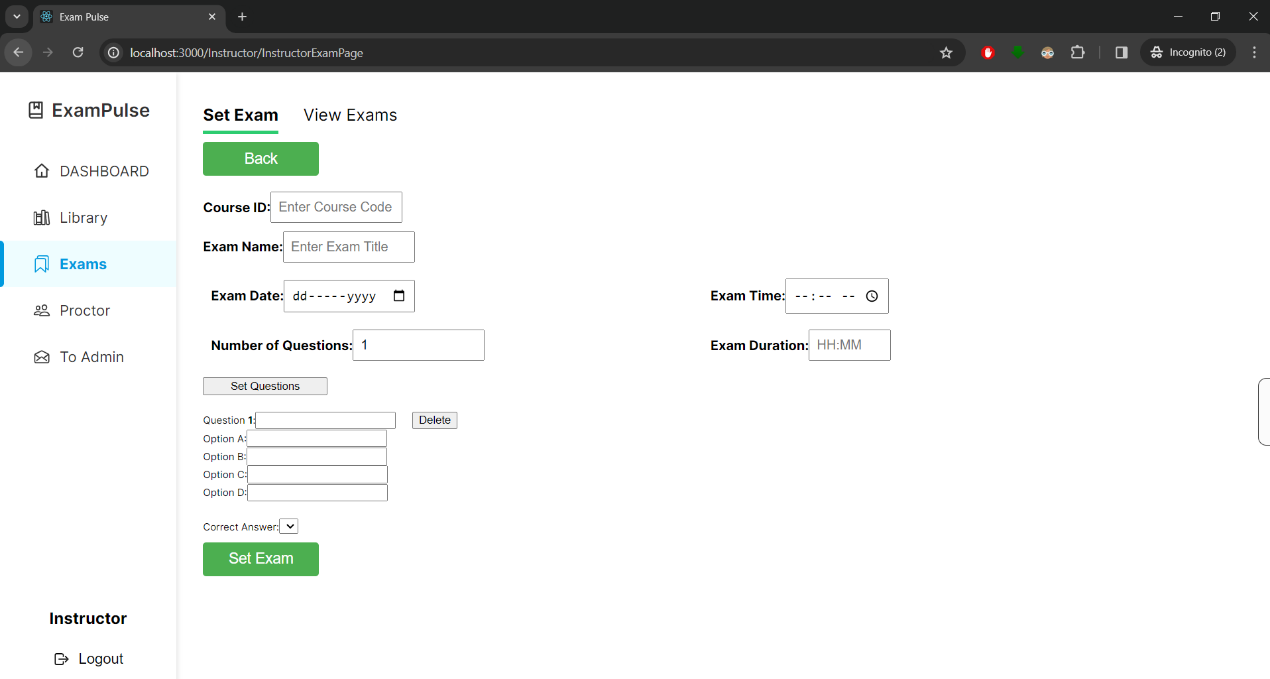


Figure 39: Set Exam Page

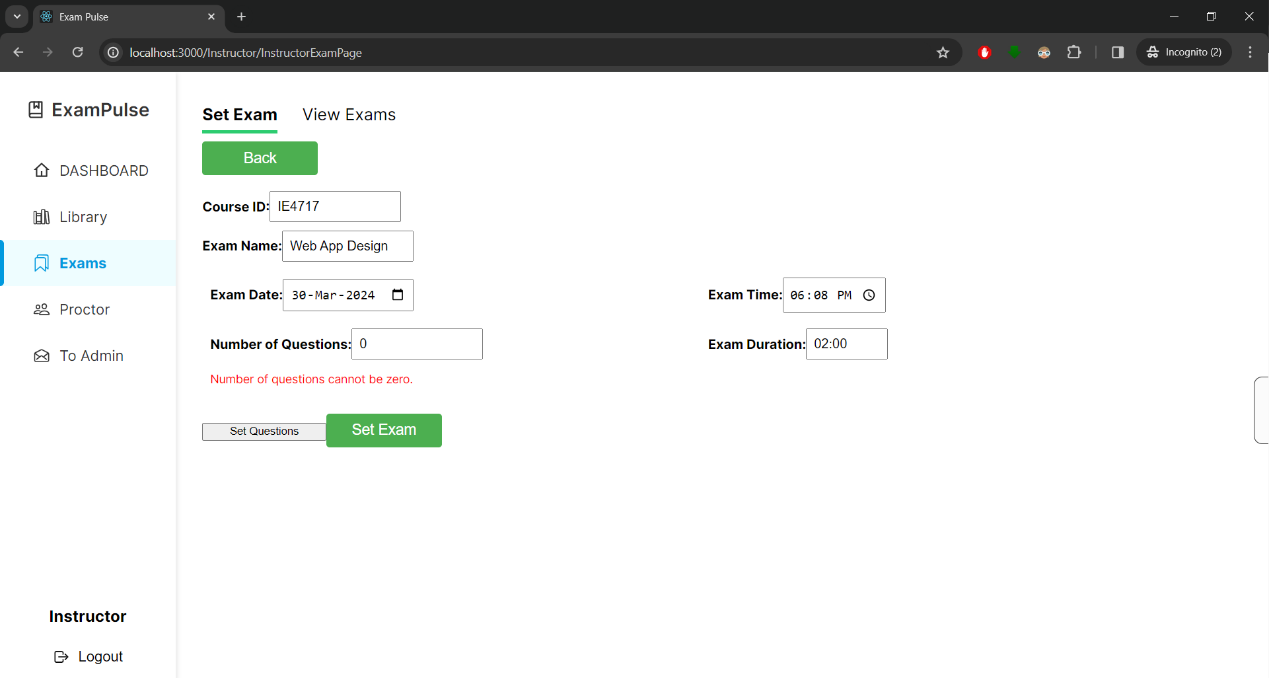


Figure 40: Question Set Error

Any incomplete fields will prevent submission of the form, accompanied by the respective error message displayed to notify instructors (shown in Figures 40 and 41). This ensures that the database does not collect any incorrect information, thereby eliminating potential issues on the student platform during the examination period.

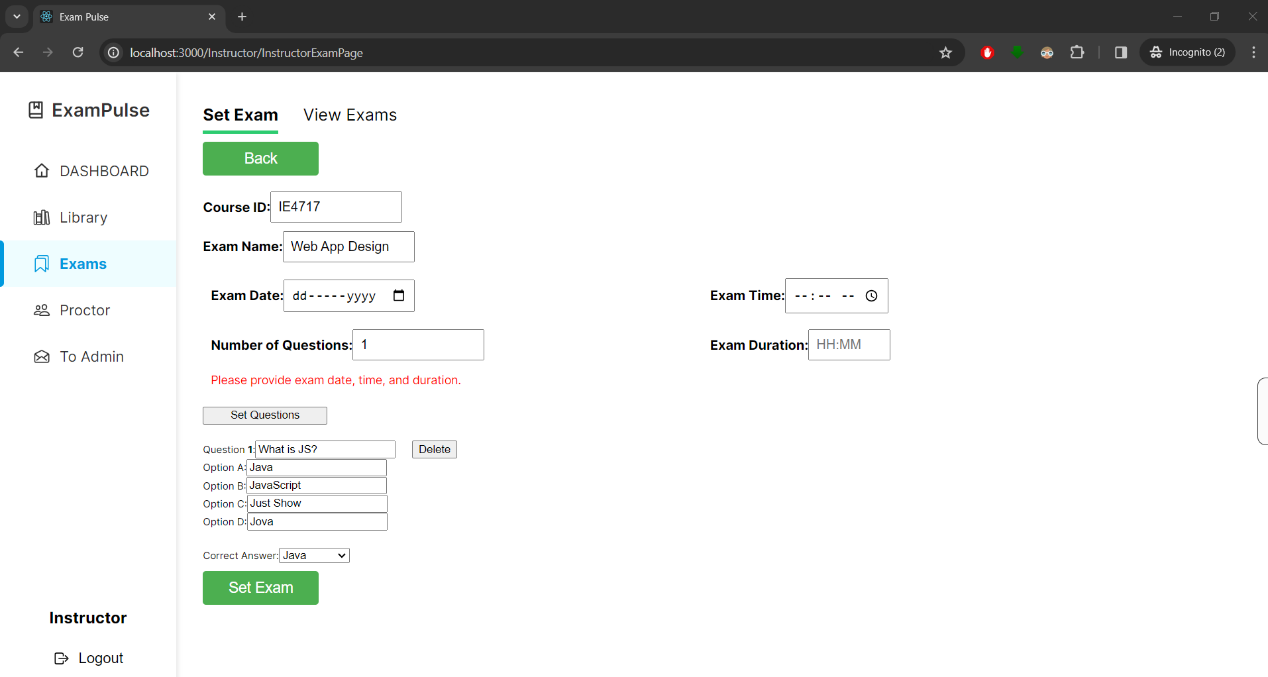


Figure 41: Error Message for Incomplete Fields

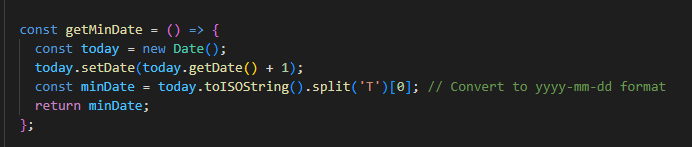


Figure 42: Calendar Restriction on Past and Present Dates

In addition, the date selection has been modified to only allow dates after the current date (shown in figure 42). This prevents the instructors from tampering with examination dates or setting up exams on past dates, thereby avoiding potential confusion for students.



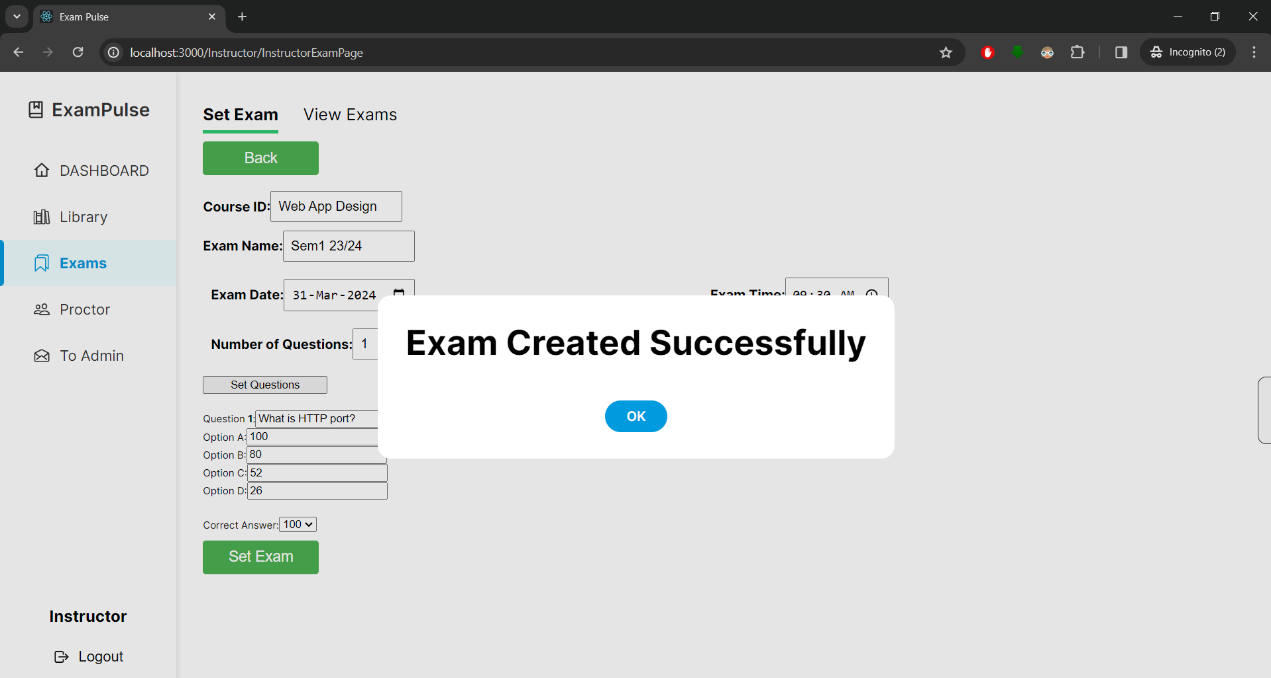


Figure 43: Modal Notification Upon Successful Set Up

After successfully setting up the examination, a modal confirming the successful setup will notify instructors (shown in Figure 43). The page will automatically refresh after a second delay with the use of **‘window.location.reload()”** function.

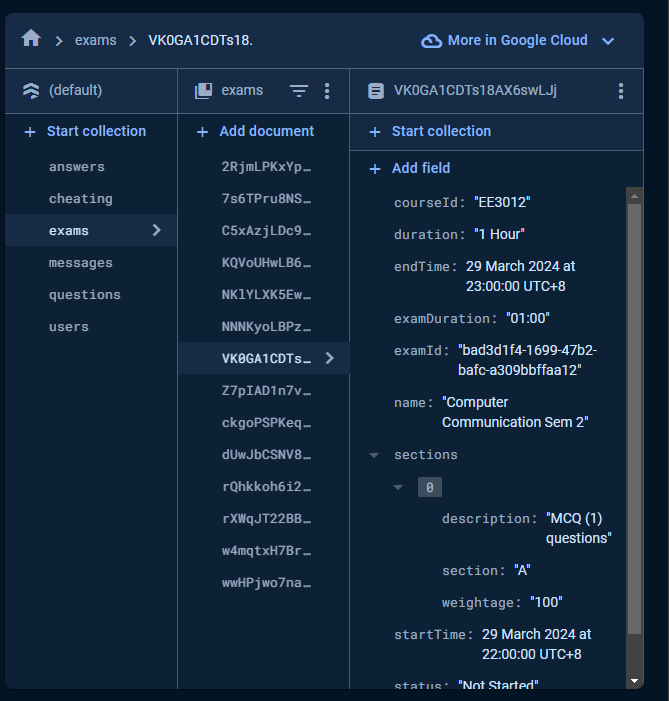


Figure 44: Firebase Console “exams” collection

The data is stored in two sections within the Firebase Console. The primary section is the “exams” collection (as shown in Figure 44), contains general information about the examination. The second section, titled the 'questions' collection (as depicted in Figure 45) is responsible for storing individual questions associated with the particular examination.

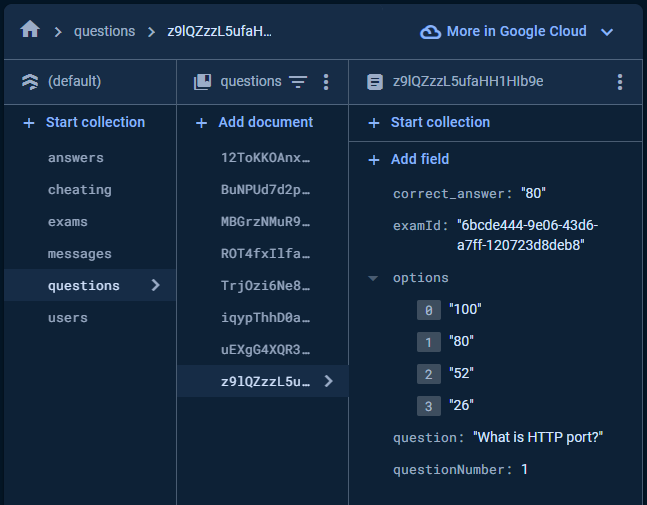


Figure 45: Firebase Console “questions” collection

A screen shot of a computer program

Description automatically generated

Figure 46: uuidv4 and Storing data to “exams” collection

Each examination created is assigned a UID generated randomly using the uuidv4 function (shown in Figure 46). This uuidv4 is a universal unique identifier that generates random numbers, ensuring the uniqueness of each identifier [18].

The UID is crucial for linking each examination to its corresponding set of questions within the ‘questions’ collection. This ensures that even if multiple examinations share similar course IDs and are created within the same timeframe, each examination is uniquely identified. Therefore, students taking the examination can only access questions linked to that specific examination’s UID.

The ‘examId’, shown in Figures 44 and 45 (underline in red), serves as the UID tied to both the 'exams' and 'questions' collections.

# 4.2.6 View Exams

In the View Exams tab, instructors can view both current and upcoming examinations. Clicking on the number of questions opens a modal that displays all questions associated with the exam UID (as shown in Figure 48).

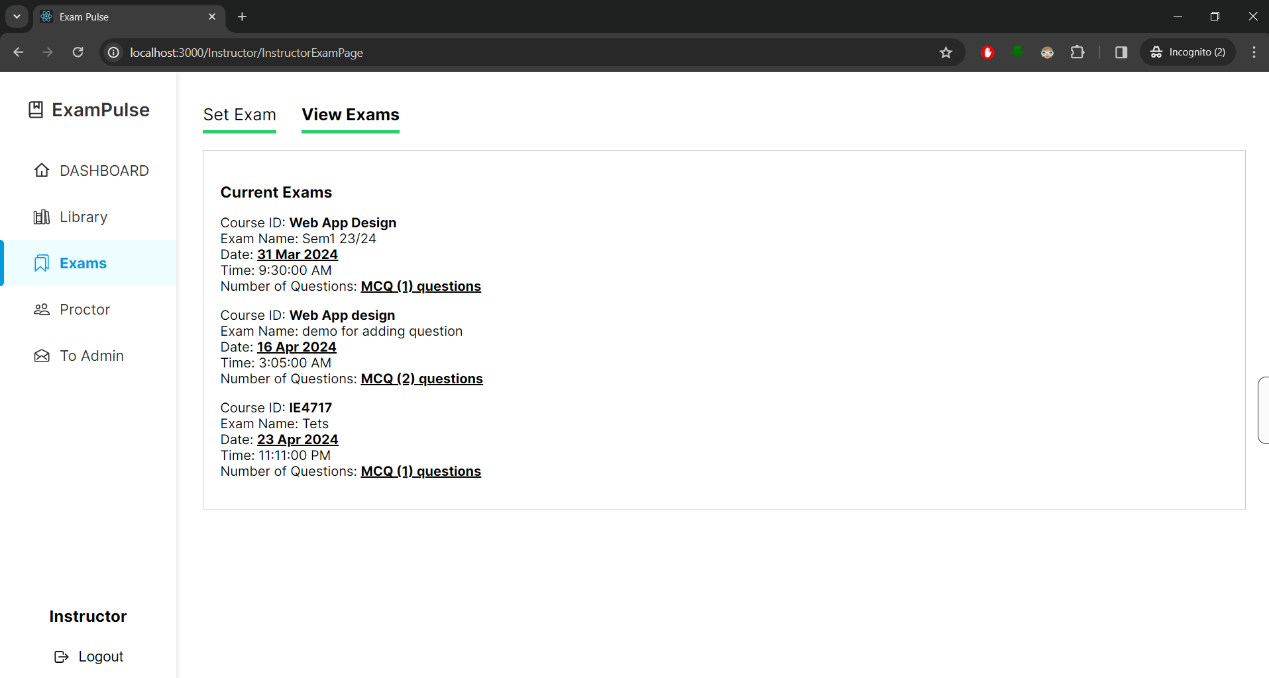


Figure 47: “View Exam” tab

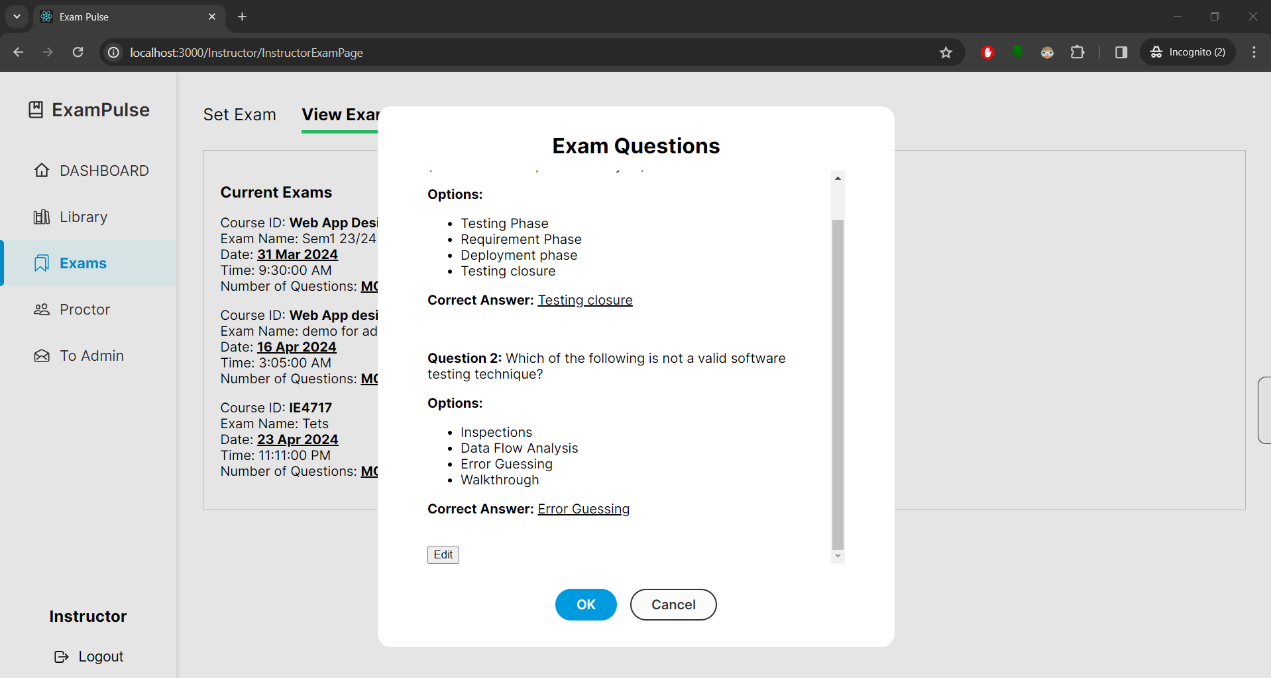


Figure 48: Modal Displaying The Questions

In addition, an edit button (highlighted in a red box in Figure 48) allows instructors to modify the questions, answer options and correct answers, as well as add or delete questions. Upon clicking the “Save” button, the changes are updated in the Firebase console.

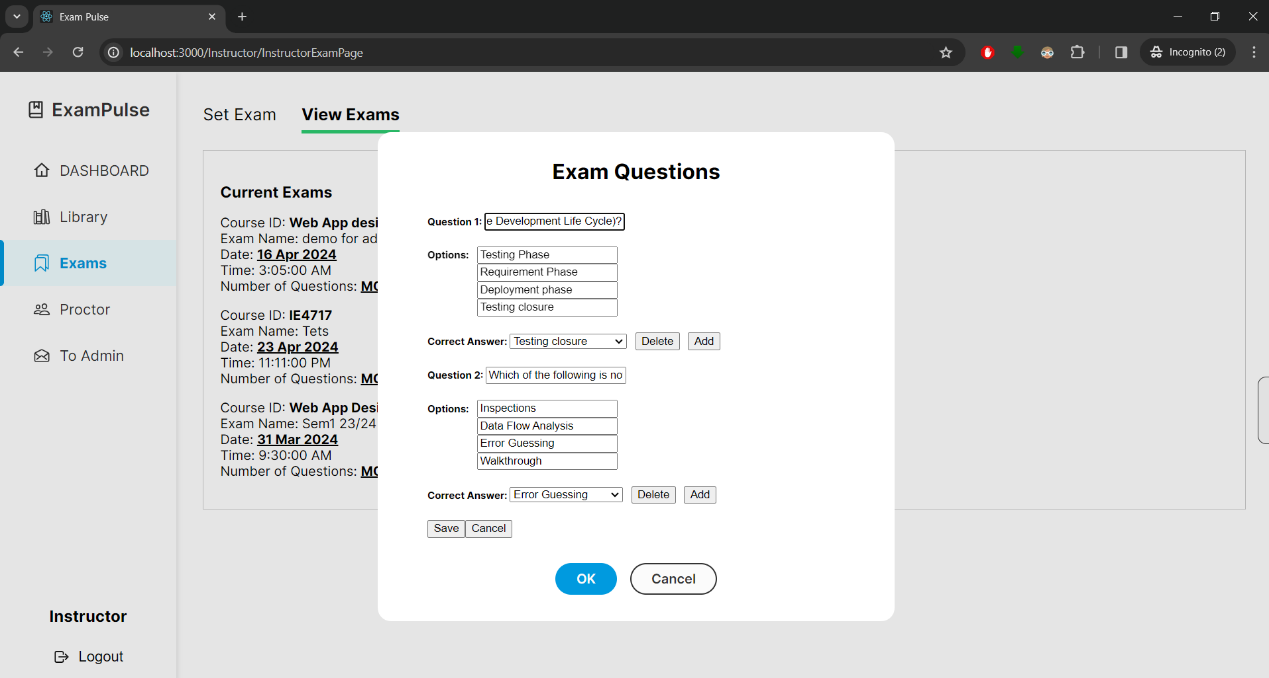


Figure 49: Managing Examination Question

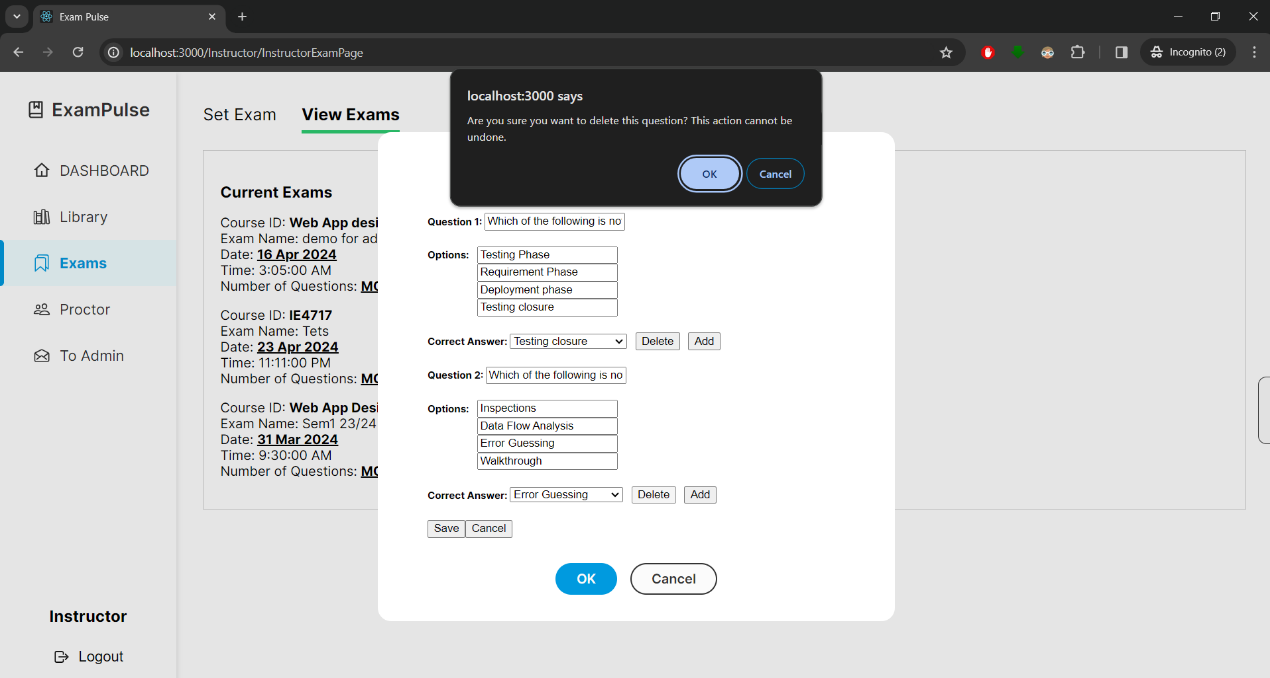
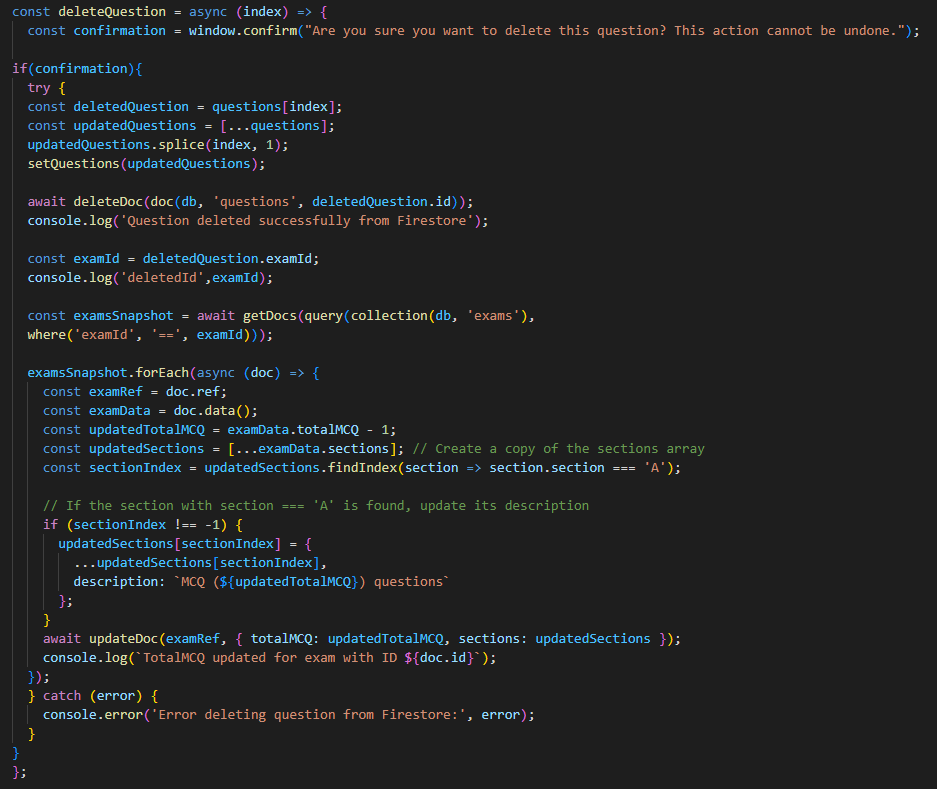
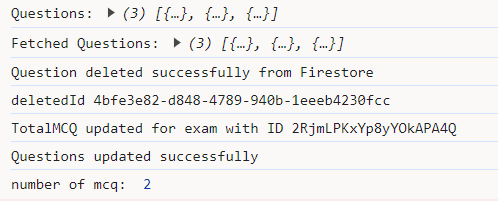


Figure 50: Prompt Instructor on Deleting

When deleting a question, a confirmation dialog window will prompt instructors (shown in figure 50), as there is no way to recover deleted questions. Once confirmed, the question is removed from the “questions” collection and updates are reflected in the “exams” collection automatically.





The code snippet here demonstrates what occurs when instructors click “Ok” upon deleting the question. The function removes the question from the local array and updates the state. Concurrently, it deletes the corresponding question document from the 'questions' collection using the 'deleteDoc' function and retrieves the exam ID of the question.

The ‘examId’ of the deleted question queries the Firebase console in the ‘exams’ collection. It then updates the total number of MCQs according to the deleted question using the 'updateDoc' function.

# 4.2.7 Proctoring Session

The proctoring site shares a similar function as the ‘View Exams’ tab (section 4.2.6 Exam) in terms of displaying the list of upcoming examinations. It queries the Firebase ‘exams’ collection and filters the examination based on start time, displaying them in ascending order. Examinations starting earliest will appear at the top of the list, while subsequent ones will be listed below.



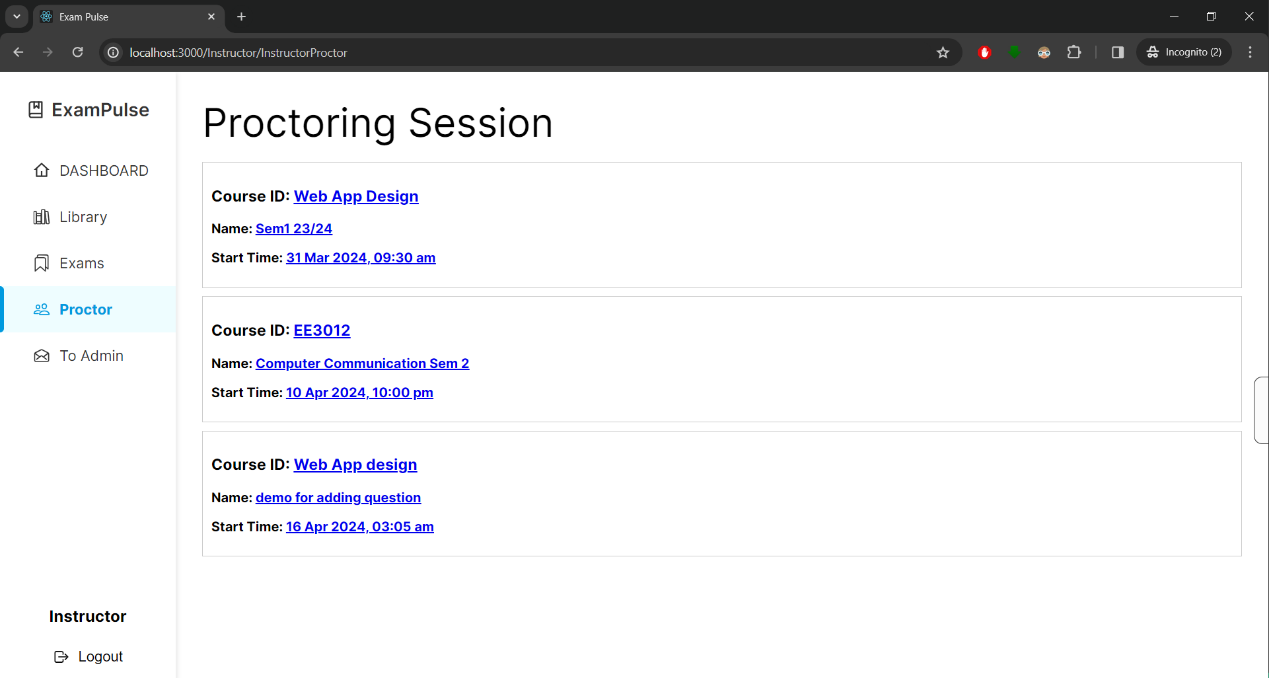


Figure 51: Proctor site showing a list of upcoming examinations

Clicking on the hypertext (in figure 51) directs instructors to the respective examination room, where students can join once the examination starts.



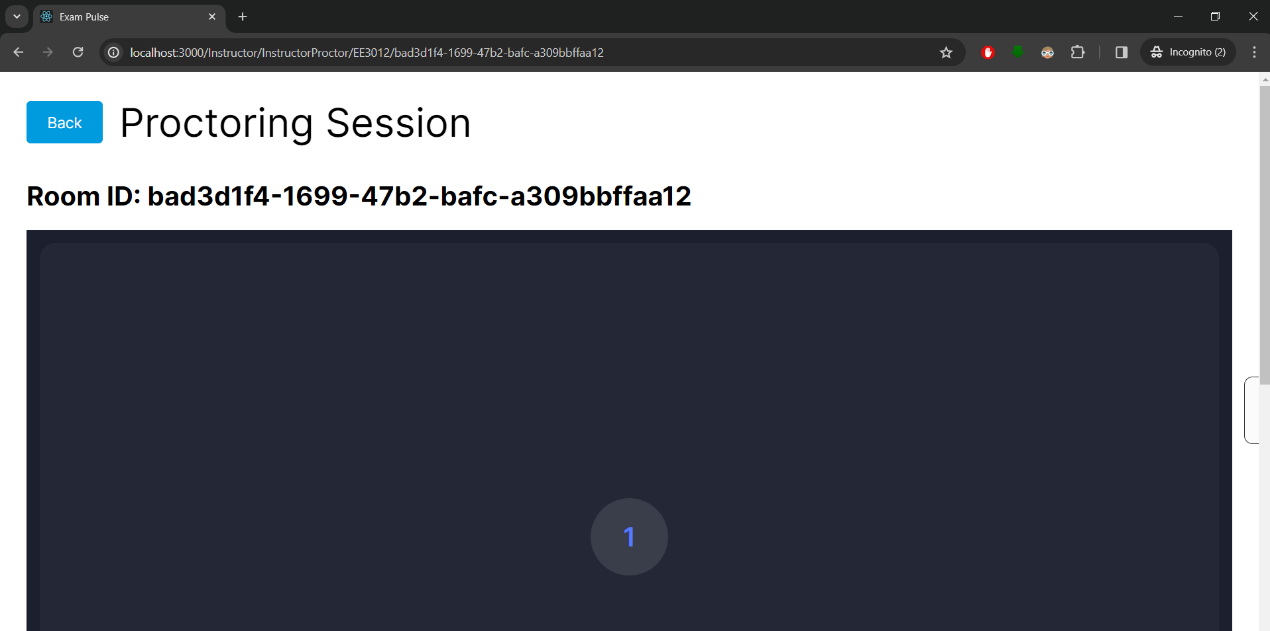


Figure 52: A Proctoring Session

Each room is uniquely routed with the use of the ‘useParams’ function to retrieve its ‘courseId’ and ‘examId’. ‘useParams’ is a feature provided by React Router, which accesses parameters defined in the URL of the route [19]. This ensures that the instructors and students are directed to the correct examination room (shown in Figure 52).

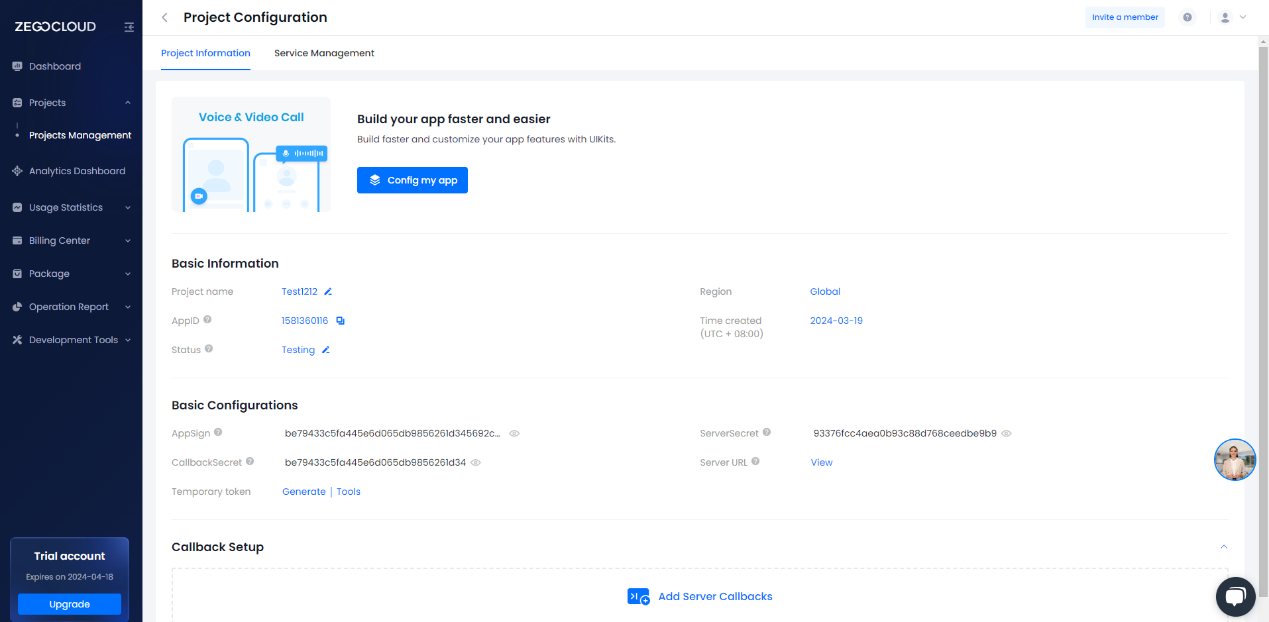




Figure 53: Zegocloud Video Code Settings

Utilizing the Zegocloud video call feature to host examination rooms enables instructors to conduct live proctoring sessions. The platform allows for the modification of settings such as chat functionality, screen recording, and various other settings. Figure 53 displays the code and the room settings.

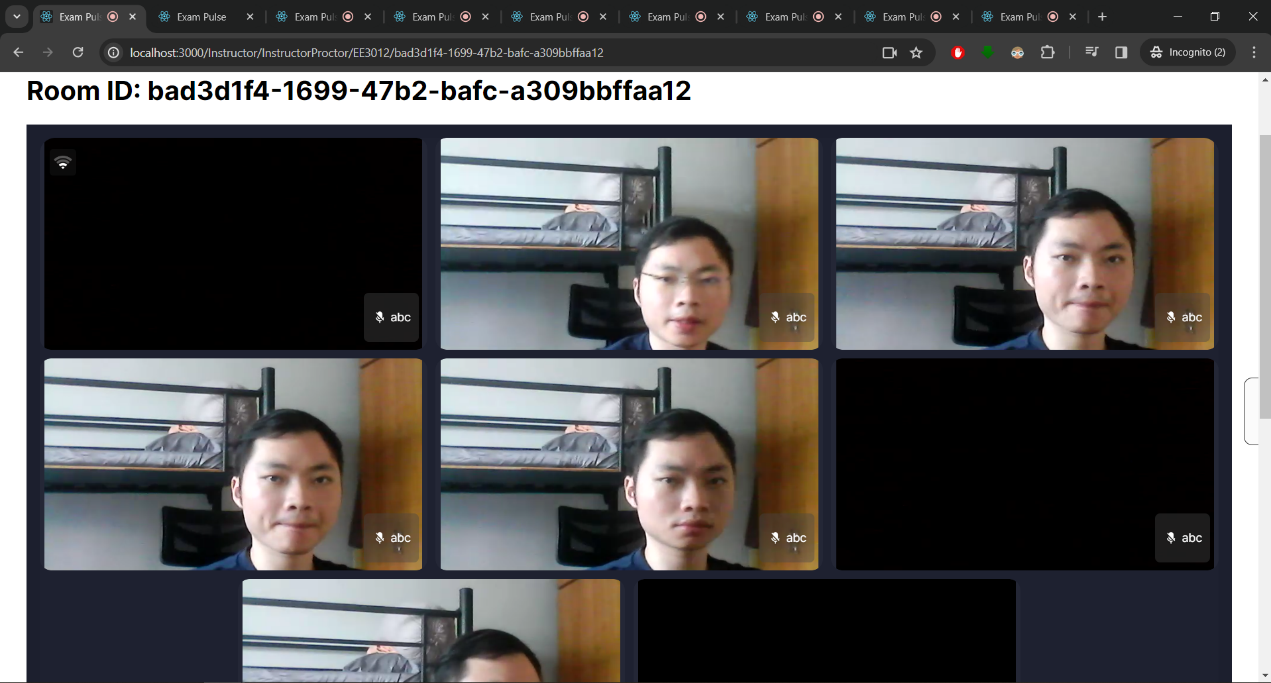


Figure 54: Live Proctoring Session

Figure 54 shows the proctoring sessions, where multiple tabs are used to simulate the students taking the exams with webcam feeds, allowing instructors to monitor multiple students simultaneously during the examination.

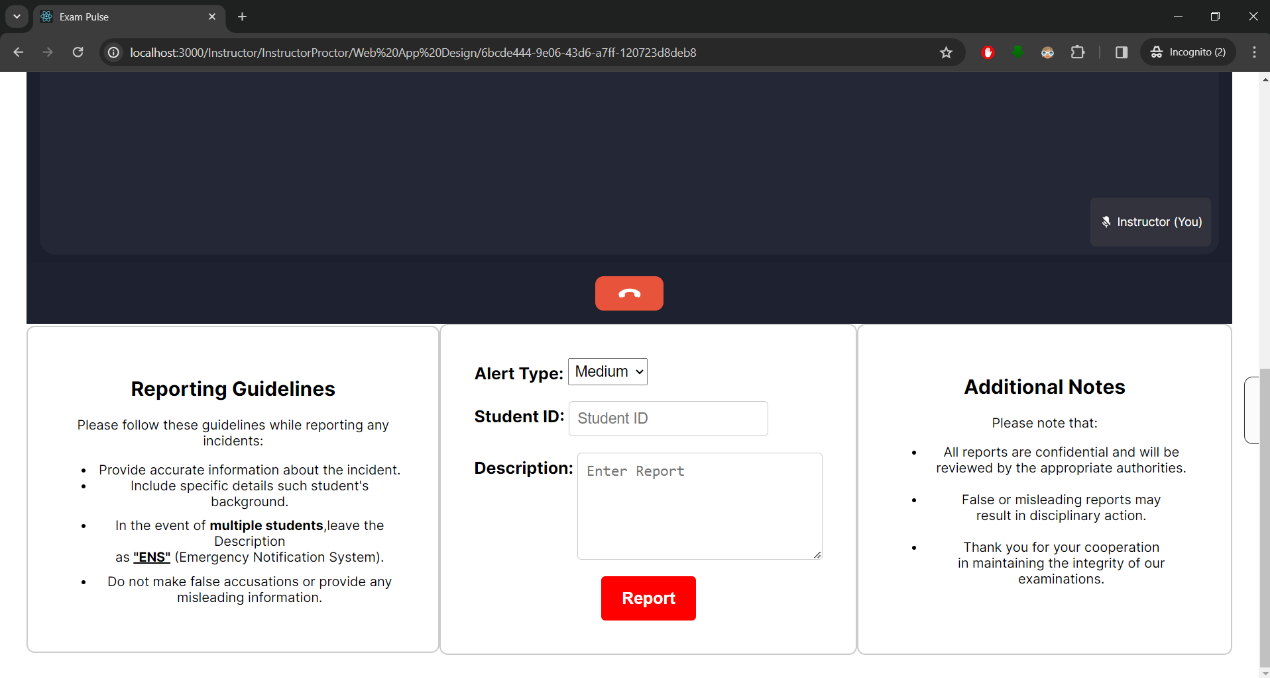
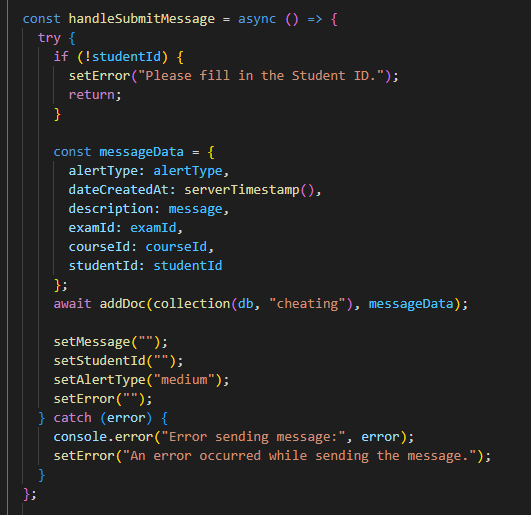


Figure 55: Reporting Platform

In addition, a reporting platform is available at the bottom of each session (shown in Figure 55), allowing instructors to report students in the event of dishonest behavior and set the alert as medium or high. Reporting guidelines are provided to assist instructors through the reporting process.

Reports are logged onto the Firebase ‘cheating’ collection, where instructors only have to input the student ID and a description of the incident. Details such as examId, and reporting date-and-time are automatically recorded upon submission of the report, ensuring prompt and accurate documentation of reported incidents.



The following code snippet demonstrates the function triggered after the report button has been clicked. The function creates a ‘messageData’ to store information, including Alert Type, Student Id, Description, Exam Id and Course Id. This object is then added to the ‘cheating’ collection using the ‘addDoc’ function.

After successfully adding the document, the function clears 'message' and 'studentId' fields, and resets the alert type to ‘Medium’. Additionally, any previous error messages are cleared to provide an efficient reporting system.

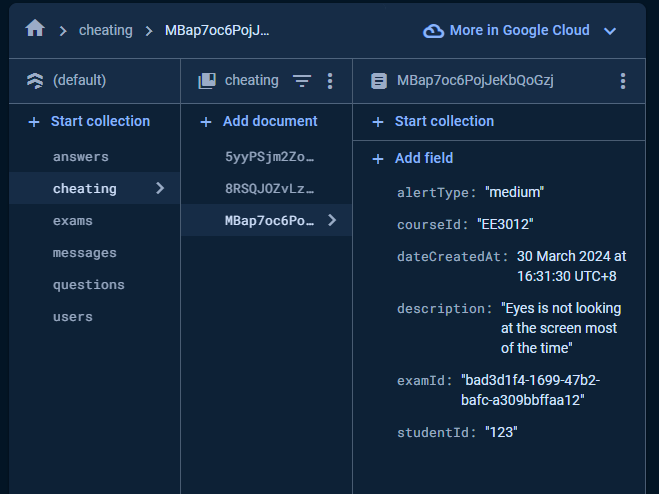


Figure 56: A “Cheating” Collection on Firebase Console

Figure 56 illustrates an exmaple of a reported incident that has been flagged by the instructor and subsequently recorded in the Firebase cheating collection. This cheating collection serves as a repository for instances of academic dishonesty, providing administrators with a comprehensive record of such occurrences.

With this, administrators can monitor and address issues related to cheating more effectively. They can analysis the patterns and keep track of repeated offenders to implement appropriate disciplinary actions, such as suspending student’s activities.

# 4.2.8 Contact Admin

The contact admin page serves as a communication platform where instructors can reach out to administrators regarding any issues or feedback. Queries may fall under several categories, including General, User-related, Bugs, and Account help.

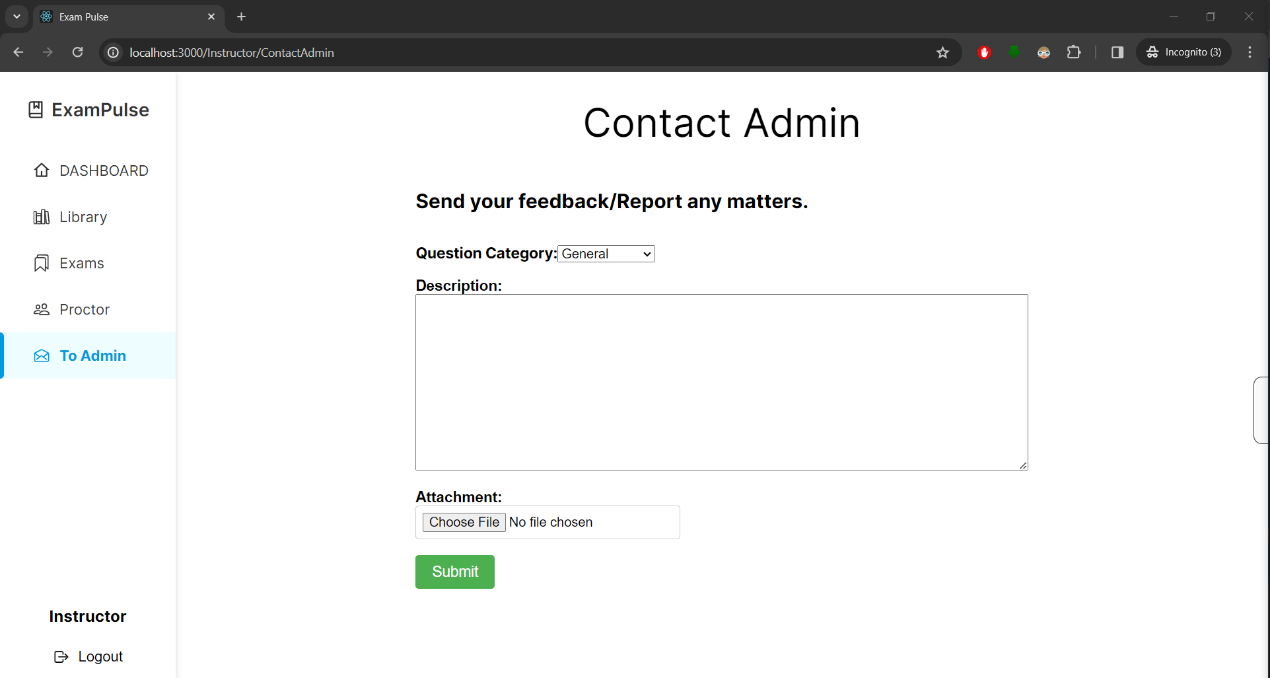


Figure 57: Contact Admin Page

In addition to submitting text descriptions, instructors have the option to attach various file types as Firebase Storage does not have strict limitations on file types, except for executable files for Windows [17]. The attachments are stored in the Firebase console within the storage folder, providing a convenient way for instructors to share relevant information with the administrator.

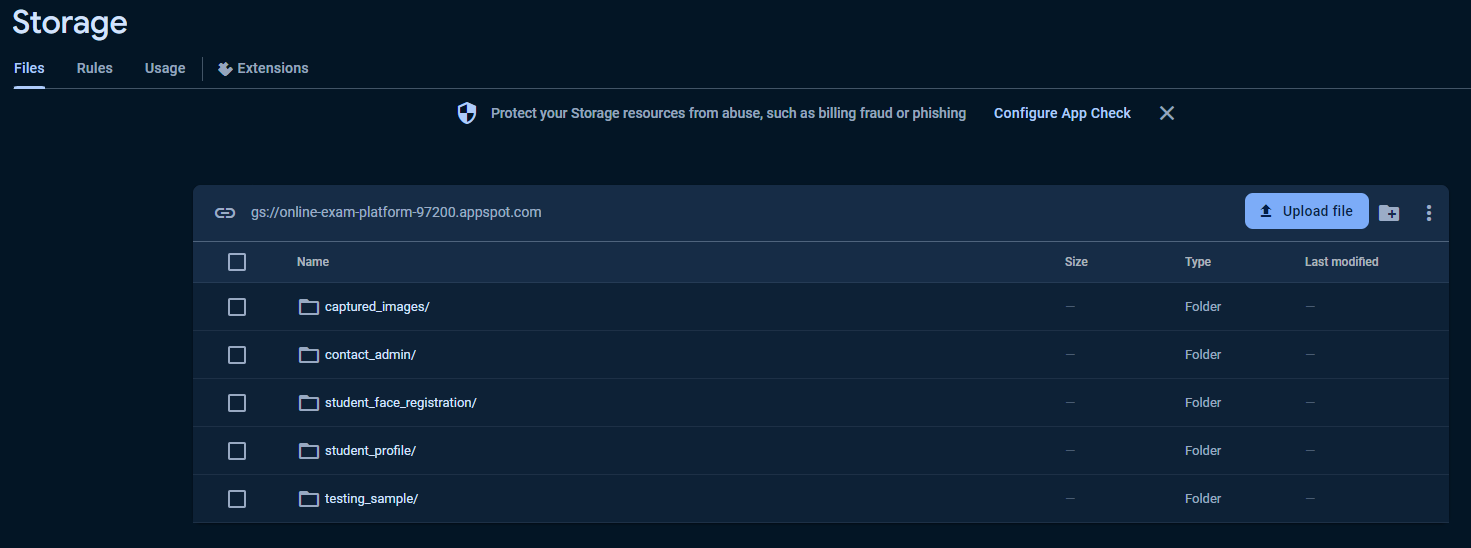


Figure 58: Firebase Storage

The Firebase Storage includes various folders for different purposes. In this case, the “Contact Admin” folder will be used to store all the attachments.

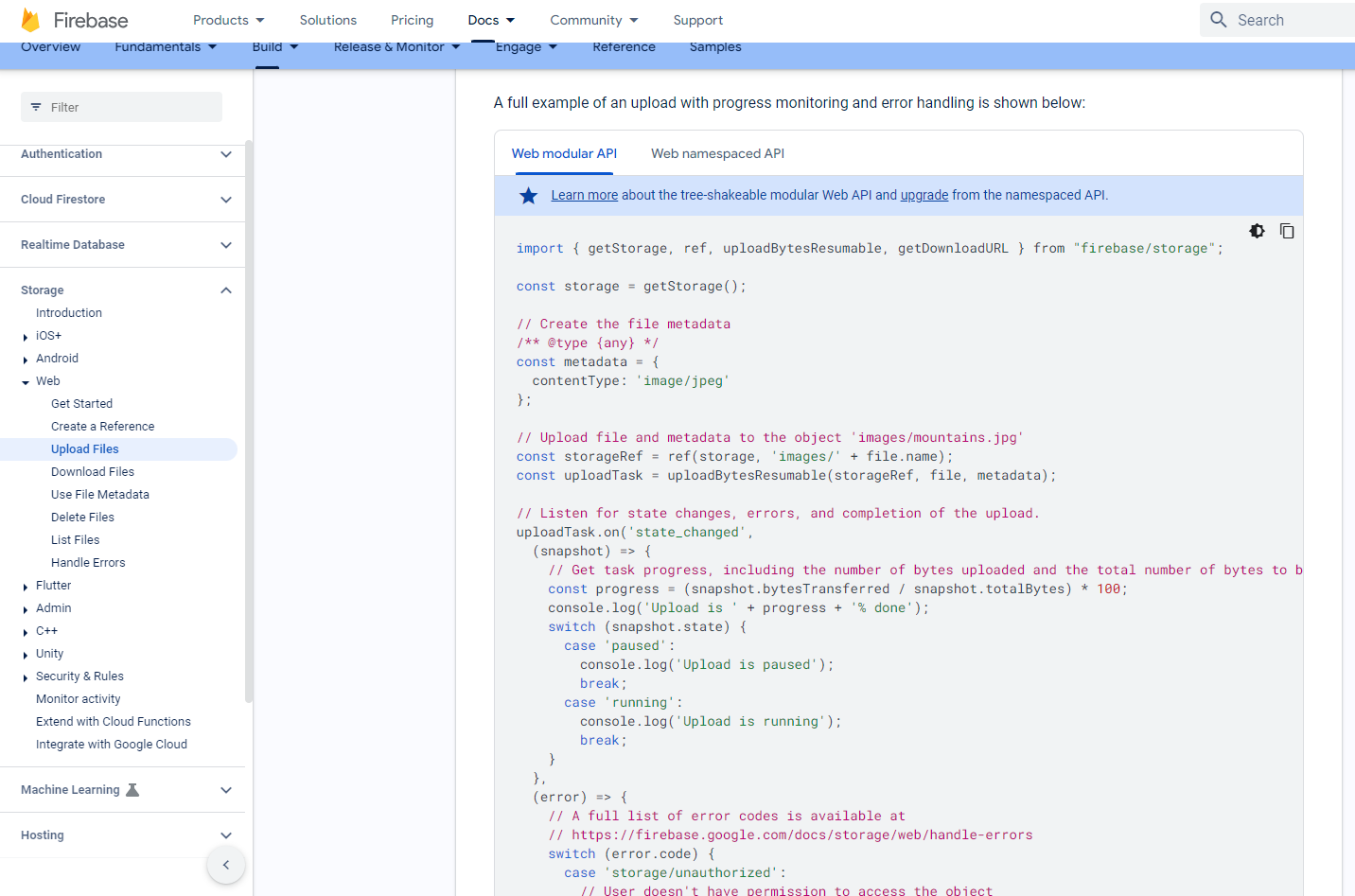
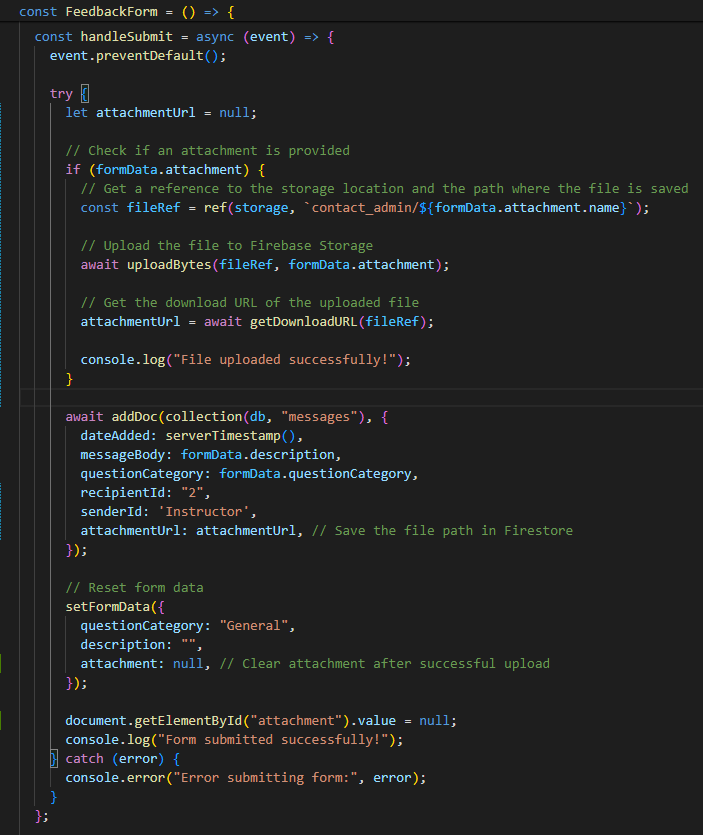


Figure 59: Firebase Documentation on Uploading Files

The Firebase documentation provides valuable information such as guides and code samples on how to implement the function (shown in figure 59) [20].



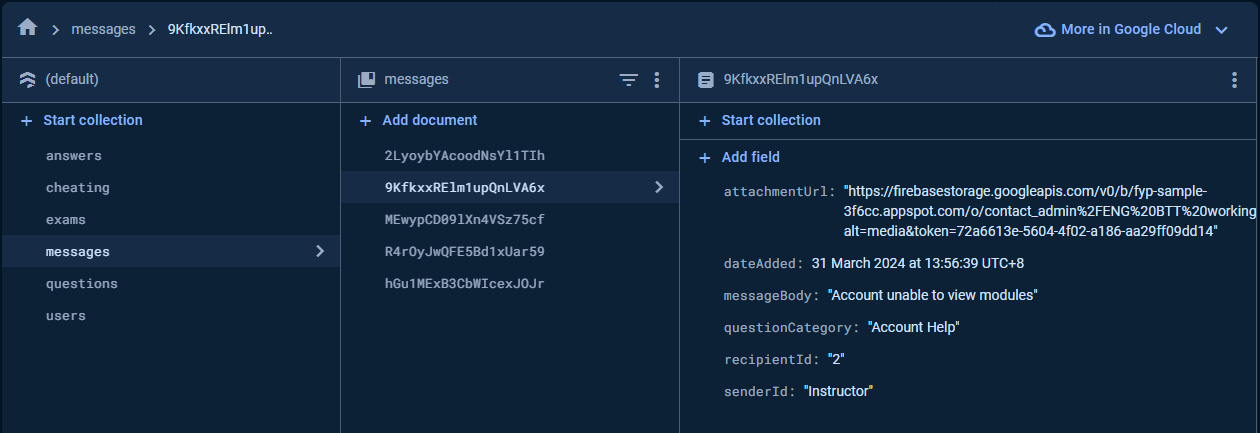


Figure 60: “Message” collection displaying message from Instructor

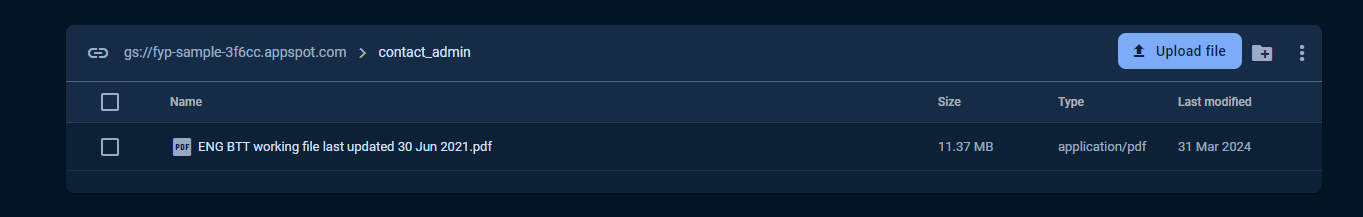


Figure 61: A File Uploaded to Firebase Storage

# “Chapter 5” Conclusions and Future Work

# 5.1 Conclusions

The development of a Secure Online Examination System provides flexibility and accessibility for students to take examination online. The integration of live proctoring features ensures the integrity of the examination process, enabling instructor to remotely monitor students and report any suspicious behavior in real-time.

Instructors play an important role in the system by managing examinations and setting up of examination sessions. Through the instructor interface, they can efficiently create and administer exams, thereby reducing the likelihood of human errors and ensuring a smooth examination for both instructors and students.

Furthermore, the instructor interface includes a communication platform between instructors and administrators, facilitating rapid responses to any technical issues or concerns that may arise during the examination process or any other aspect of the system.

Overall, the efficient exam setup and live proctoring features of the Secure Online Examination System enhance reliability and minimize the risk of academic dishonesty and human errors, making it a more secure environment for online examinations.

# 5.2 Recommendation for Future Work

There are several areas where future developments can further improve the Secure Online Examination System:

1. **Enhanced Proctoring Technologies:** Implement AI for behavior analysis and facial recognition to enhance the detection of suspicious activities during examinations. Abnormalities will be flagged immediately for review by instructors or administrators.
2. **Customizable Exam Settings:** Allow instructors to further customize settings, such as randomizing question order and timing restrictions set for each question or sections. This ensures that each student receives the exam in a unique sequence, thereby increasing the difficulty of cheating and promoting fairness in assessment.
3. **Auto Grading Function:** Introduce an auto-grading function to automatically evaluate students' answers, streamlining the grading process for instructors and provides immediate feedback to students. With this feature, students can promptly identify areas for improvement and gauge their understanding of the material.

# “Chapter 6” Reflection on Learning Outcome Attainment

**Problem Analysis:**

Throughout the project, conducting thorough problem analysis played a crucial role in identifying the challenges and requirements for establishing a secure platform for online examinations. This involved studying existing systems, understanding potential security threats, and analyzing user requirements.

With the implementation of the instructor interface, the system addresses the identified issues of ensuring academic integrity during online examinations. It provides instructors with the functionality to create, manage, and proctor examinations securely, including features such as live proctoring via webcam and a reporting function, which demonstrate a proactive approach to mitigating cheating and maintaining the credibility of online exams.

Thus, highlighting the importance of problem analysis in driving the development process, the system was designed to meet the needs of instructors while minimizing potential dishonest acts.

**Investigation:**

Conducting research and investigations into existing platforms was a stepping stone in developing our online secure examination system. This process involved conducting case studies, exploring various design concepts, and testing different software technologies to determine their suitability for our project requirements.

During the investigation phase, I analyzed the strengths and weaknesses of each platform, enabling me to make informed decisions regarding their integration into our system. By evaluating these aspects, I could identify the most effective approach to implementing essential features such as the calendar overview, question management, and proctoring functionality. This ensured that instructors have easy access to a wide range of functions, enhancing usability and efficiency.

Furthermore, the investigation process played a vital role in identifying and mitigating potential risks to ensure the successful implementation of the system. For example, implementing safeguards such as requiring instructors to confirm their choice before deleting a question helped prevent inadvertent data loss due to human error.

**Lifelong Learning:**

This project served as a valuable platform for me to acquire new skills and gain practical experience in full-stack software development. Throughout the project, I actively pursued self-directed learning to become proficient in tools like Figma and React.js, which were instrumental in crafting wireframes and designing the user interface.

Additionally, the collaborative nature of our project created an environment of shared learning and growth. Collaborating closely with my team members allowed me to leverage their expertise in areas where I had less experience, such as backend development or data management. This collaborative learning accelerated my understanding of the project's complexities and expanded my skill set.

Reflecting on this experience, I now appreciate the importance of embracing a mindset of continuous improvement and adaptation. The skills and experiences gained from this project have instilled in me the confidence to tackle complex challenges in the future. Whether it involves collaborating in diverse teams or learning new technologies, I am well-prepared to embrace lifelong learning on my journey forward.

# References

[1] D.Singh, “Online examinations: Everything you need to know about going Digital,” mettl, https://blog.mettl.com/guide/online-examination/ Sep. 23, 2023).

[2] H. H. Gorgani and S. Shabani, “Online exams and the COVID-19 pandemic: A hybrid modified FMEA, QFD, and *k*-means approach to enhance fairness,” SN applied sciences, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8477634/

[3] Today, “Some undergrads admit to cheating in online exams during pandemic, despite safeguards and stiff penalties,” TODAY, https://www.todayonline.com/singapore/some-undergrads-admit-cheating-online-exams-during-pandemic-despite-safeguards-and-stiff-penalties-1884581 (accessed Sep. 23, 2023).

[4] P. M. Newton and K. Essex, “How common is cheating in online exams and did it increase during the COVID-19 pandemic? A systematic review - journal of academic ethics,” SpringerLink, https://link.springer.com/article/10.1007/s10805-023-09485-5 (accessed Sep. 24, 2023).

[5] T.Inc, “Online assessment platform software,” Talview,2019, https://www.talview.com/online-assessment-tool

[6] Talview Online Proctoring and Assessments Platform. (n.d). Microsoft Talview https://appsource.microsoft.com/en/product/web-apps/talview.talviewonlineassessments?tab=overview

[7] Synap, “Online exam & assessment platform,” Synap, https://synap.ac/ (accessed Aug. 29, 2023).

[8] Smowl, “SMOWL - Online exam proctoring software,” SMOWL Proctoring | Supervision system for online exams, https://smowl.net/en/ (accessed Aug. 29, 2023).

[9] Live Video Call SDK & API for web and mobile. ZEGOCLOUD (n.d) https://www.zegocloud.com/product/video-call

[10] Video Call SDK Quotas and Limits, ZEGOCLOUD (n.d). https://www.zegocloud.com/docs/video-call/quotas-limits?platform=ios&language=objective-c

[11] Web JavaScript Call Kit SDK Quick Start, ZEGOCLOUD (n.d). https://docs.zegocloud.com/article/14935

[12] React. (n.d.). React. https://react.dev/

[13] K.T. Hanna, Google Firebase. Mobile Computing. https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase#:~:text=The%20Firebase%20Realtime%20Database%20is,even%20when%20users%20are%20offline.

[14] Staff, C. What is Github and why should you use it? Coursera. https://www.coursera.org/articles/what-is-git

[15] Figma. (n.d). Figma: https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma-

[16] GfG. Use case diagrams Unified Modeling Language. GeeksforGeeks. https://www.geeksforgeeks.org/use-case

[17] Cloud Storage for Firebase (n.d) Firebase. https://firebase.google.com/docs/storage/web/upload-files#:~:text=Note%3A%20For%20Spark%20plan%20projects,for%20Firebase%20and%20Firebase%20Hosting.

[18] npm: uuid. (n.d) Npm. https://www.npmjs.com/package/uuid

[19] GfG. ReactJS useParams Hook. GeeksforGeeks. https://www.geeksforgeeks.org/reactjs-useparams-hook/

[20] Upload files with Cloud Storage on Web (n.d) Firebase. https://firebase.google.com/docs/storage/web/upload-files