****

**A SECURE ONLINE EXAMINATION SYSTEM**

**Submitted by: Yap Qi Long, Marcel**

**Supervisor: A/P Chua Hock Chuan**

School of Electrical & Electronic Engineering

A final year project report presented to Nanyang Technological University

in partial fulfilment of the requirements for the

Degree of Bachelor of Engineering

2024

# Table of Contents

[Table of Contents ii](#_Toc163848142)

[Abstract ii](#_Toc163848143)

[Acknowledgment iii](#_Toc163848144)

[Acronyms iv](#_Toc163848145)

[List of Figures v](#_Toc163848146)

[List of Tables vii](#_Toc163848147)

[Chapter 1 Introduction 1](#_Toc163848148)

[1.1 Motivations 1](#_Toc163848149)

[1.2 Objectives and Scope 2](#_Toc163848150)

[1.3 Organisation of Report 3](#_Toc163848151)

[Chapter 2 Literature Review 4](#_Toc163848152)

[2.1 About 4](#_Toc163848153)

[2.2 Existing Examination Platforms Comparison 4](#_Toc163848154)

[2.3 Software Project Management 8](#_Toc163848155)

[2.4 Git 10](#_Toc163848156)

[2.5 Stages of Software Development 12](#_Toc163848157)

[2.6 Database Research 14](#_Toc163848158)

[2.7 Design System and User Interface 20](#_Toc163848159)

[2.8 Front End Technologies 23](#_Toc163848160)

[Chapter 3: Design 25](#_Toc163848161)

[3.1 Features Required 25](#_Toc163848162)

[3.2 User Journey 25](#_Toc163848163)

[3.3 Design System 26](#_Toc163848164)

[3.4 Wireframing 30](#_Toc163848165)

[3.5 Components used for development 33](#_Toc163848166)

[Chapter 4: Implementation 34](#_Toc163848167)

[4.1 Folder Structure 34](#_Toc163848168)

[4.2 Database 34](#_Toc163848169)

[4.3 Back End Implementation 37](#_Toc163848170)

[4.4 Front End Implementation 44](#_Toc163848171)

[4.5 Component Creation 47](#_Toc163848172)

[Chapter 5 Results 48](#_Toc163848173)

[5.1 Exams Control 48](#_Toc163848174)

[5.2 Modules 49](#_Toc163848175)

[5.3 Personnel 50](#_Toc163848176)

[5.4 Messages 53](#_Toc163848177)

[Chapter 6 Conclusion and Future Work 56](#_Toc163848178)

[6.1 Conclusion 56](#_Toc163848179)

[6.2 Future Work 57](#_Toc163848180)

[Reflection on Learning Outcome Attainment 58](#_Toc163848181)

[Acknowledging/Declaring the Use of GAI 60](#_Toc163848182)

[References 62](#_Toc163848183)

# Abstract

Examinations are a key part of an individual’s education to assess and determine if they learnt the required skills from that educational institution. Traditional examinations are conducted physically with pen and paper with invigilators, however there is an increasing popularity for digital solutions to conduct examinations online after the COVID –19 pandemic. The elimination of a physical examination venue and finding manpower for invigilators which saves time and resources has contributed significantly to their popularity.

ExamPulse is a secure online examination platform which facilitates seamless participation, administration, and oversight of examinations for students, educators, and administrators alike. Its list of features includes live video feeds during examinations, automated marking, personnel management, and more, all aimed at optimizing the examination process.

The platform's development follows a structured industry-standard pipeline, encompassing requirement gathering, design, development, testing, and deployment phases. The platform was developed using currently popular technologies such as ReactJS and Firebase.

This paper explores the rationale behind the shift to online examinations and delves into the features and development approach of ExamPulse. It offers a secure and efficient solution for online assessments.

# Acknowledgment

First, I would like to thank my parents for their contribution to allowing me to have a university education. Next, I thank Prof Chua for his support and guidance throughout the project's development. I would like to thank my teammates for their patience and cooperation in developing this product. Lastly, a big thank you to all my friends, especially Tasha, for their moral and emotional support throughout the project’s development.

Marcel Yap

April 2024

# Acronyms

|  |  |
| --- | --- |
| FE  BE  DB  UI  UX  AI  App | Front End  Back End  Database  User Interface  User Experience  Artificial Intelligence  Application |

# List of Figures

[Figure 1: Student view of Socrative. 4](#_Toc163848718)

[Figure 2: Teacher view of Socrative. 5](#_Toc163848719)

[Figure 3: SpeedExam Student View: Question 6](#_Toc163848720)

[Figure 4: SpeedExam Instructor View: Dashboard 6](#_Toc163848721)

[Figure 5: Quizizz Landing Page 7](#_Toc163848722)

[Figure 6: Student System Check 8](#_Toc163848723)

[Figure 7: Example of a Kanban Board in Jira 10](#_Toc163848724)

[Figure 8: Git Branches Concept [19] 12](#_Toc163848725)

[Figure 9: Stages of Software Development 13](#_Toc163848726)

[Figure 10: Firebase console and examples of data stored from testing. 15](#_Toc163848727)

[Figure 11: Home page which allows you to post data with a certain format. 16](#_Toc163848728)

[Figure 12: GraphQL in action used for querying or mutation data in the database. 16](#_Toc163848729)

[Figure 13: AWS MySQL Database Home Page 17](#_Toc163848730)

[Figure 14: DBeaver - User Table view and sample data. 18](#_Toc163848731)

[Figure 15: DBeaver - Using SQL script to insert data into table. 19](#_Toc163848732)

[Figure 16: pgAdmin Home Page 19](#_Toc163848733)

[Figure 17: Site Map for Admin 26](#_Toc163848734)

[Figure 18: Colour Theme for App 27](#_Toc163848735)

[Figure 19: Selection of Shortlisted Fonts 28](#_Toc163848736)

[Figure 20: Logo and Logo with name. 29](#_Toc163848737)

[Figure 21: Figma Colour Library 30](#_Toc163848738)

[Figure 22: Partial list of components for wireframing. 31](#_Toc163848739)

[Figure 23: Admin pages wireframe. 32](#_Toc163848740)

[Figure 24: ER diagram. 35](#_Toc163848741)

[Figure 25: Firebase GUI for users’ collection after initalisation. 36](#_Toc163848742)

[Figure 26: Preview of user account in Authentication. 37](#_Toc163848743)

[Figure 27: Example of file in Storage. 37](#_Toc163848744)

[Figure 28: Configuration of Firebase.js file. 38](#_Toc163848745)

[Figure 29: Example of adding data to database. 39](#_Toc163848746)

[Figure 30: Example of reading data. 40](#_Toc163848747)

[Figure 31: Example of using query 40](#_Toc163848748)

[Figure 32: Example of updating data. 41](#_Toc163848749)

[Figure 33: Example of deleting data. 41](#_Toc163848750)

[Figure 34: Users collection with example user data. 43](#_Toc163848751)

[Figure 35: Same user in authentication table. 43](#_Toc163848752)

[Figure 36: Handle user logging in code. 44](#_Toc163848753)

[Figure 37: Using React Router to map out all possible routes. 45](#_Toc163848754)

[Figure 38: Example of styled components usage. 46](#_Toc163848755)

[Figure 39: Modal using styled components. 46](#_Toc163848756)

[Figure 40: Input array in parent component using Navbar. 47](#_Toc163848757)

[Figure 41: Navigation bar displaying pages in Navbar.js. 47](#_Toc163848758)

[Figure 42: Exams page 48](#_Toc163848759)

[Figure 43: Edit exam time modal 49](#_Toc163848760)

[Figure 44: Modules List Page 50](#_Toc163848761)

[Figure 45: Adding Modules Modal 50](#_Toc163848762)

[Figure 46: Personnel Page 51](#_Toc163848763)

[Figure 47: Filtering users in Personnel page. 51](#_Toc163848764)

[Figure 48: Delete personnel modal. 52](#_Toc163848765)

[Figure 49: Add new personnel page. 52](#_Toc163848766)

[Figure 50: Editing personnel page. 53](#_Toc163848767)

[Figure 51: Messages Page 54](#_Toc163848768)

[Figure 52: Messaging component in Instructor’s page. 55](#_Toc163848769)

# List of Tables

[Table 1: Development components 33](#_Toc163835604)

[Table 2: Project Folder Structure 34](#_Toc163835605)

# Chapter 1 Introduction

## Motivations

Examinations are essential to the education system in Singapore and the world to assess students and determine if they have the proper skills learnt from that institution. In 2020, the COVID-19 pandemic caused the closure of many schools and prompted the move to e-learning, thus examinations were also shifted to the virtual space. Digital solutions for online examinations had become increasingly popular due to rapid technological advances during the COVID-19 pandemic [1]. Also, they have become increasingly popular due to the elimination of the need to set up a physical examination venue and allocating invigilators for the examinations, saving considerable time and resources.

However, one big challenge for online examination platforms is the detection of cheating in an online examination due to the absence of genuine invigilation unlike face-to-face examinations. It is also extremely difficult to prove online cheating and thus threatening academic integrity [2]. 73.6% of a sample of undergraduate business students from a university in the South that took examinations held the view that that it was much easier to cheat in an online examination compared to a traditional one [3].

The propensity for students to engage in cheating is known to diminish when there is an increased level of surveillance or the perception of constant monitoring. Students exhibit reduced instances of academic dishonesty when they are aware that their actions are being scrutinized, as illustrated by the observed decrease in cheating behaviours among individuals participating in examinations conducted via Zoom, where the students were made known that the session was being recorded.

Thus, online proctoring on students is essential during examinations to catch any suspicious behaviours or attempts at cheating. Due to the vast number of student cameras to look at, the integration of Artificial Intelligence and Computer Vision would significantly help instructors perform proctoring duties by detecting suspicious eye or head movement and flag them out if spotted [4]. Currently, there are existing online examination platforms in the market, however most of them do not have surveillance features nor do they integrate AI to help with proctoring exams.

Therefore, this project aims to develop a “Secure Online Examination Platform” web application for universities. This application will attempt to emulate the experience of a real face-to-face examination for both student and instructor as much as possible. The application will also incorporate a control panel, enabling instructors to manage all aspects of the examination both prior to, during, and after its administration. Instructors will observe all students participating in the examination via their computer cameras and manually flag any cheating detected. The application will integrate AI and Computer Vision to help the institution proctor students and detect any attempt at cheating, maintaining academic integrity [5]. Furthermore, the incorporation of additional functions, such as automated grading and statistical analysis can be considered for implementation if needed. Moreover, the application presents an alternative should physical examinations become inaccessible again.

## Objectives and Scope

The primary objective of this research project is to design and implement a Secure Online Examination System that can be universally applied across universities and other academic institutions. This system is compartmentalized into three distinct components: Students, Instructors, and Administrators. The development of each component has been allocated to an individual student, with this report specifically delineating the component concerning the administrator.

**Scope of administrator:**

1. Manage users on the app.
2. Access all examinations.
3. Answer queries from Instructors.
4. Manage modules offered by the institution.

## Organisation of Report

Chapter 1: Introduction - This first chapter provides an overview of the project, its motivation and scope.

Chapter 2: Literature Review – The literature review chapter will offer a critical analysis of existing research and examine various potential technologies applicable to this application.

Chapter 3: Design – The design chapter will examine the methodology involved in designing the application, encompassing both the user interface and database architecture.

Chapter 4: Implementation – The implementation chapter will show the methodologies employed in the development of various features within the application.

Chapter 5: Results – The Results chapter will present the final features of the developed application and describe its functionality.

Chapter 6: Conclusion and Future Work – The conclusion chapter will summarize the app's development process and propose enhancements for future iterations.

# Chapter 2 Literature Review

## 2.1 About

This chapter aims to present research on current technologies used in online examinations and explore various software development technologies that could be applied in this project.

## 2.2 Existing Examination Platforms Comparison

To gain a deeper insight into the execution of online examinations, a critical analysis of various existing platforms was undertaken. This analysis facilitated the extraction of each platform’s features, strengths, and weaknesses, which will inform the design of this app. Comprehensive user journeys for each product were documented in Figma using screenshots at each stage of the user experience.

**2.2.1 Socrative**

Socrative is a quiz-like platform that offers various modes for quizzes, including "Quiz," "Space Race," "Multiple Choice," "True or False," "Short Answer," among others [6]. Students are provided with a user-friendly interface, enabling them to easily select and submit their answers before proceeding to the next question as shown in Figure 1.

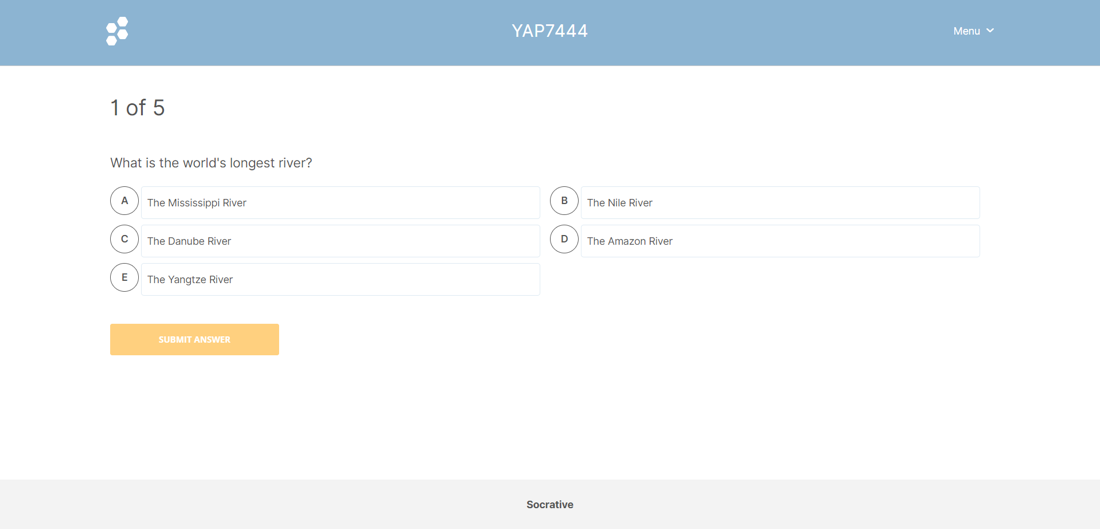


Figure 1: Student view of Socrative.

Instructors can effortlessly create quiz questions and are equipped with a dashboard that displays an analysis of how many students selected the correct answers, enabling instant feedback as illustrated in Figure 2.

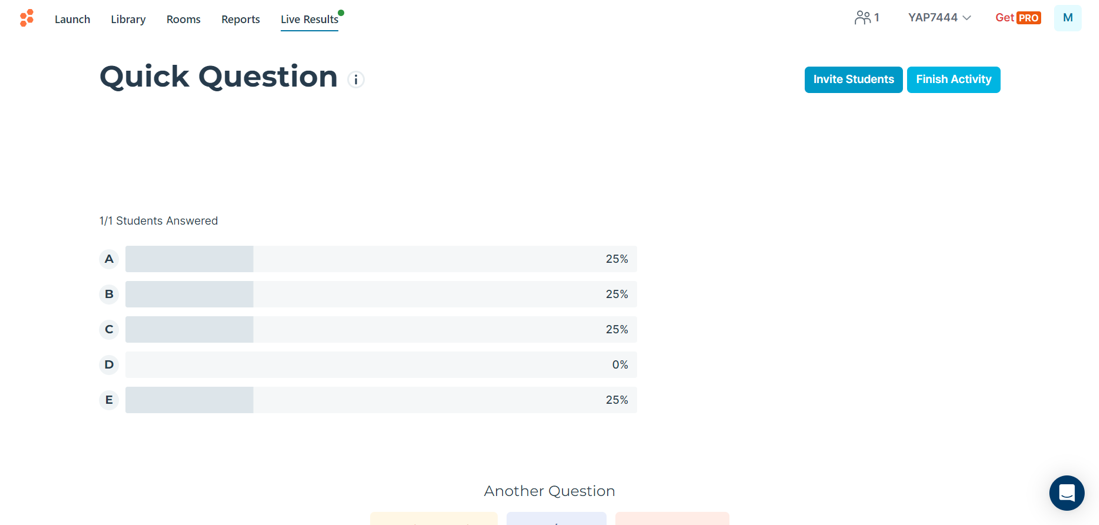


Figure 2: Teacher view of Socrative.

Summary of analysis of the platform:

* The user interface is adequate and clear for teachers but could benefit from enhancements such as smoother transitions.
* The font size of the questions for students can be increased for easier viewing and better UX.
* The design system exhibits inconsistency, with varying fonts and design styles employed throughout the platform.
* Certain terminology is occasionally unclear; providing descriptions or tooltips upon hovering over different quiz types would enhance the user experience.
* The functionality to create a room, invite students, and display results operates effectively.

**2.2.2 SpeedExam**

SpeedExam is a platform that employs AI to detect cheating or any suspicious activity in students' backgrounds. It offers a comprehensive dashboard for both instructors and students, providing numerous functionalities such as the ability to add new personnel and set up examination questions with ease [7] as shown in the figures below.



Figure 3: SpeedExam Student View: Question

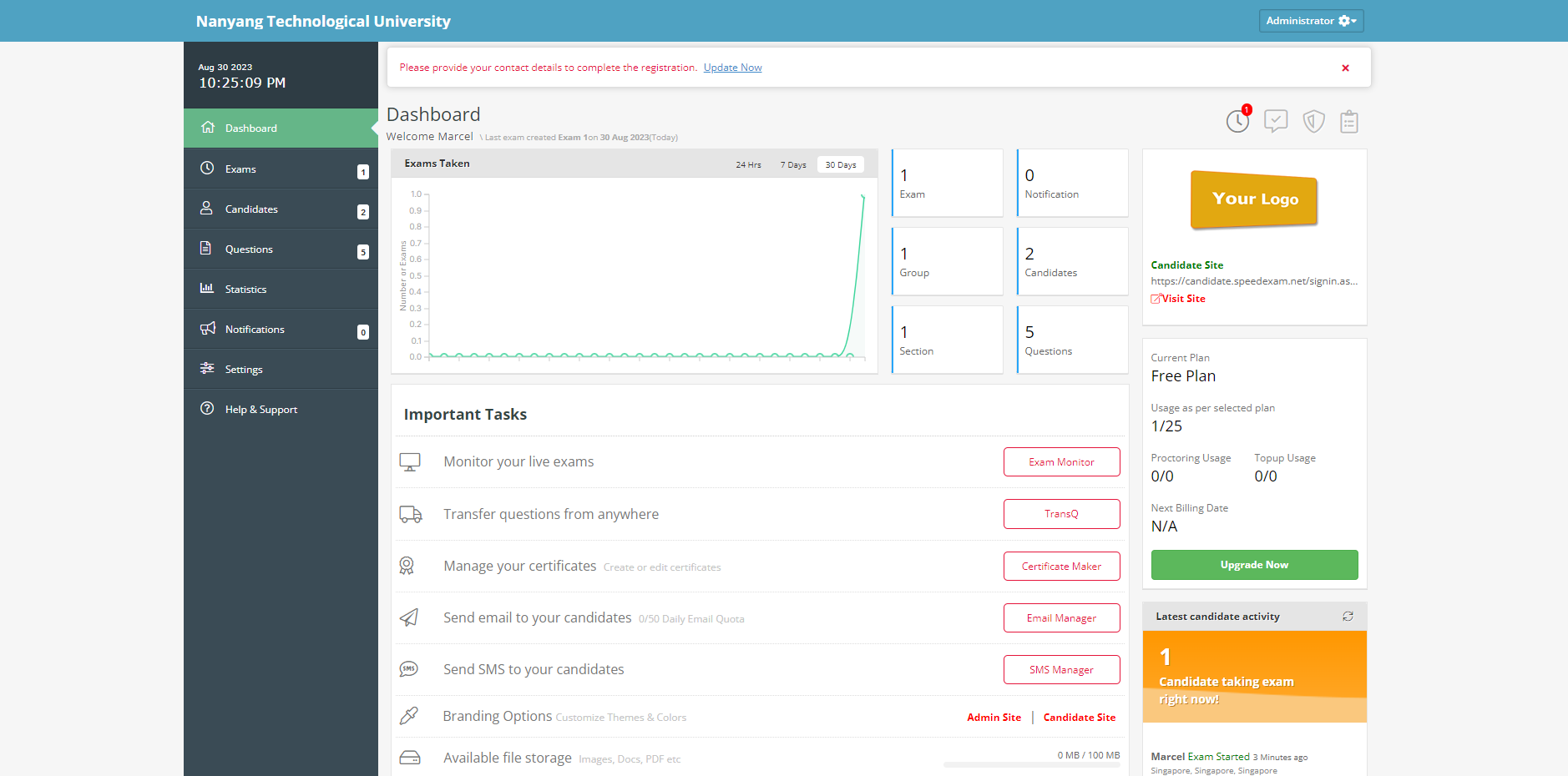


Figure 4: SpeedExam Instructor View: Dashboard

Summary of analysis of the platform:

* Comprehensive dashboard with many useful features.
* Font family for student can be improved with a clearer font.
* Platform is secure.

**2.2.3 Quizizz**

Quizizz offers gamified quizzes, assessments, instruction, and practice designed to motivate students toward mastery [8]. The platform also facilitates quiz creation with features such as "Create quiz from your own content" and "Create with YouTube," simplifying the process for users as shown in the figure below.

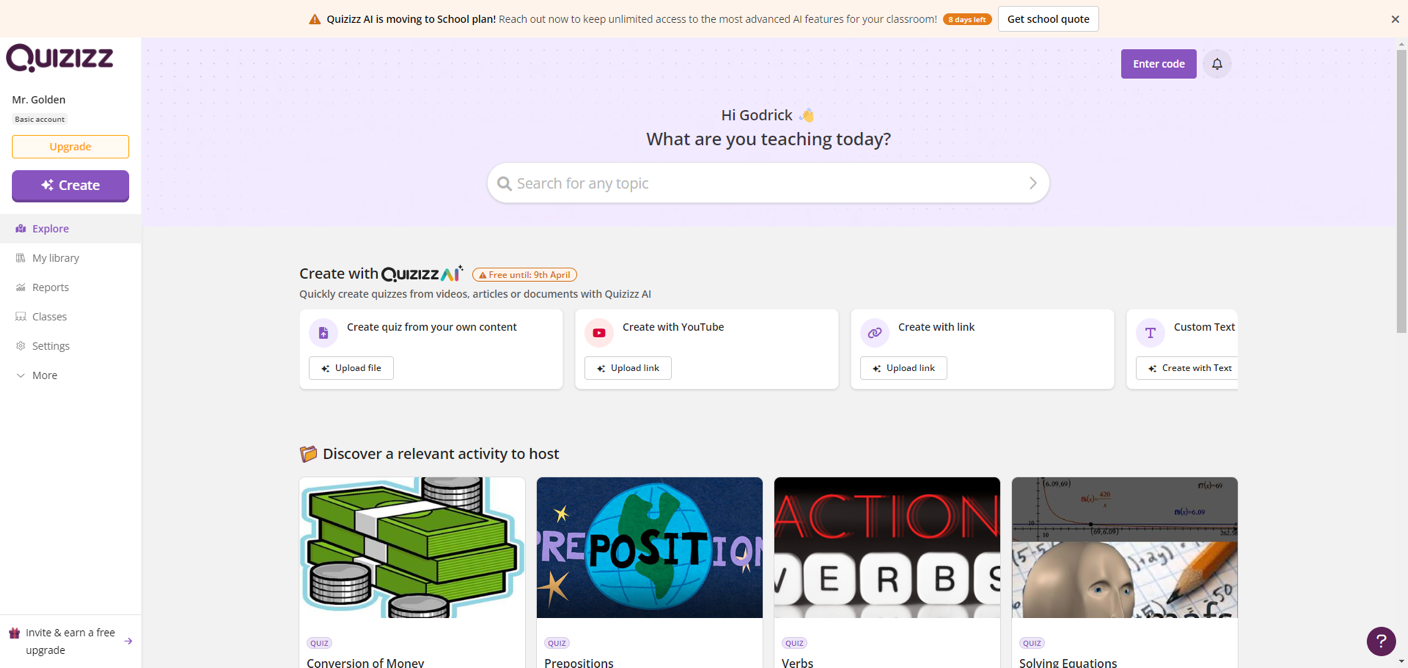


Figure 5: Quizizz Landing Page

Summary of analysis of the platform:

* Although not an examination platform, their UI and UX design has impressive qualities that offer valuable insights.
* Functions to ease the process of creating quizzes or questions, saving time.

**2.2.4 ProctorExam**

ProctorExam is an examination platform with full video proctoring. To facilitate examination proctoring, students must activate both the camera on their computer and on their mobile device. The ProctorExam App must be installed on their mobile device and logged into the identical account and examination session. Additionally, API documentation is also available for use by the platform.

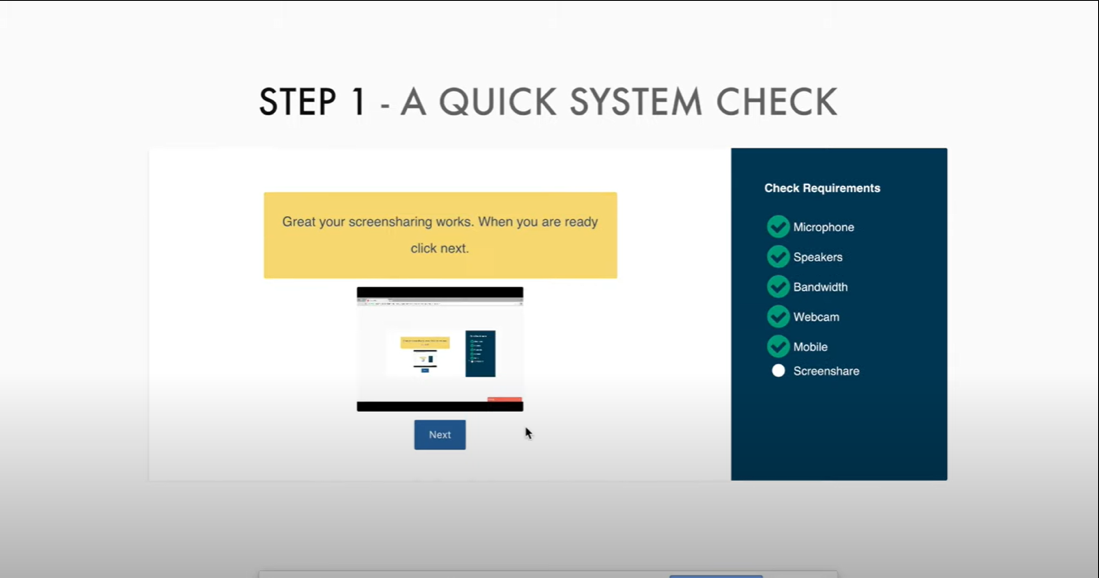


Figure 6: Student System Check

Summary of analysis of the platform:

* The platform requires an additional camera positioned behind the student to monitor their activities and the surrounding environment, necessitating the use of an app on the phone.
* Monitors the student and their background automatically.
* Monitors everything on the computer screen
* Fully automated solutions

## 2.3 Software Project Management

**2.3.1 Definition and Importance**

Software project management aims to streamline the complex process of software development which has many different aspects. This is accomplished by enabling the coordination, communication, and collaboration among team members, stakeholders, and resources involved in the project.

One of the reasons for the necessity of software project management is the efficient allocation of resources. Effective management of resources, including personnel, finances, time, and technology, boosts their utilization efficiency, thereby facilitating cost-effective and timely completion of projects.

Another significant reason is clear communication: Software project management cultivates transparent communication channels among team members, stakeholders, and clients. This clarity helps prevent misunderstandings, encourages alignment, and reduces the likelihood of errors and confusion among team members, ultimately leading to significant time and resource savings [9]. The subsequent sections will delve into various technologies employed to facilitate this goal.

**2.3.2 Agile Development**

An Agile Development workflow entails releasing features within a shorter timeframe to collect feedback and adapt the project's direction accordingly, adhering to the principle of continuous improvement [10]. This product utilizes an Agile workflow, providing flexibility and ensuring that the project meets its requirements effectively [11].

**2.3.3 Jira**

Jira, a software application developed by Atlassian, enables teams to track issues, manage projects, and automate workflows. Given that this project will employ an agile workflow, the use of a Kanban Board is recommended to monitor tasks and overall progress efficiently.

Jira provides Kanban Boards which are a visual representation of the team’s workflow using small cards that can represent either a bug or task. These cards can be expanded to show more details of the ticket. They can be assigned to different team members, allowing for task delegation.

Upon completion of a ticket, it can be transitioned to various columns denoting the stage at which the ticket currently resides, such as "in progress" or "completed" [12].

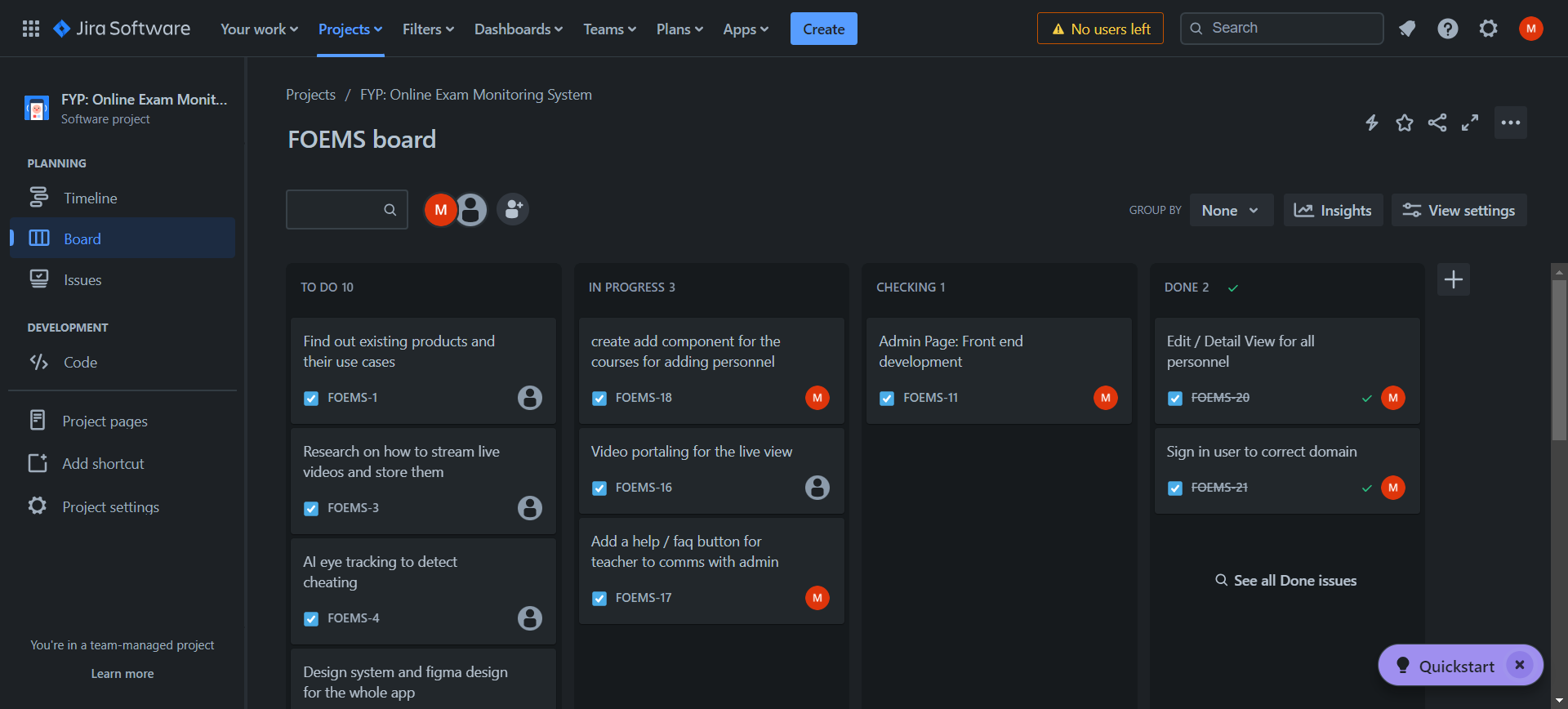


Figure 7: Example of a Kanban Board in Jira

**2.3.4 Confluence**

Confluence is a software application developed by the software company Atlassian and it is a team workspace which allows members to easily collaborate with one another. It can be used to take notes or write documentation for code written or findings from research conducted. This allows the collective knowledge of the project to be compiled and it can also be linked to any Jira tickets allowing for a smoother workflow.

## 2.4 Git

Git is a free and open-source distributed version control system which can handle a project of any size. Git was created in 2005 as an alternative to passing around changes in code via patches and archived files. It works by preserving the whole history of the codebase and everyone can access it [13].

**2.4.1 Importance of Version Control and Collaboration**

Version control is crucial as it enhances collaboration and teamwork during the software development process. Version control systems enable simultaneous collaboration among developers on a shared codebase, thereby reducing the risk of conflicts or data loss. This not only improves teamwork but also provides a systematic approach to monitoring code changes and facilitates effective tracking [14].

**2.4.2 Using Git For Version Control**

There are various methods to implement version control in your project using Git, such as through the Command Line Interface (CLI) or by utilizing applications like GitHub Desktop, which offers an intuitive user interface for managing versions. There are some important terminologies associated with using Git such as [15][16][17]:

* **Repository**: A basic element of Github, it is a place where the project’s code is stored on Github’s servers and each file’s version history can be seen.
* **Branch**: A parallel version of the project’s code that is contained within the repository but does not affect the primary or main branch.
* **Commit**: A commit is like saving changes to the project’s files and Git assigns a unique ID or hash that allows it to be identified.
* **Merge**: Merging is the action of putting a forked history back together if branches have diverged. It allows you to take independent lines of development and merge them back into a single branch, usually the main or master branch.

The combination of these different aspects of Git allows for development of multiple different features or parts of the project at once with multiple collaborators.

Commits in Git capture a "snapshot" of the most recent changes a developer has made, allowing others to pull these changes to their local devices to test or review the code. This process aids in identifying areas for improvement.

When a developer completes a feature and merges their branch with the main branch, other developers can easily pull the updated version and rebase their own branches to incorporate the latest changes. This process ensures a seamless continuation of development [18].

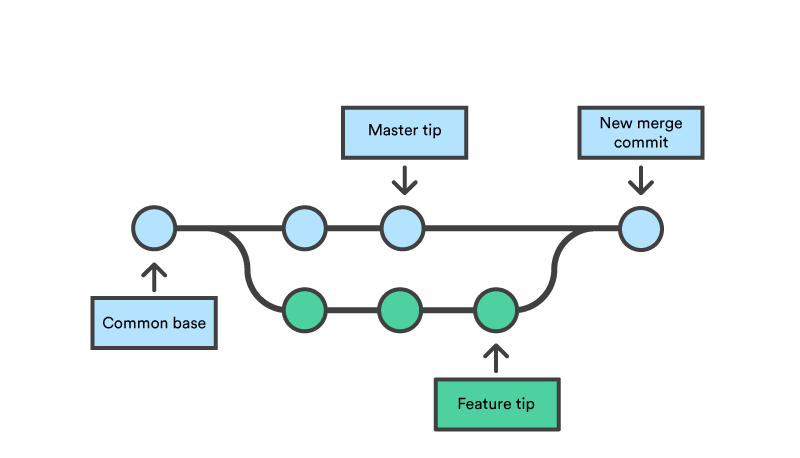


Figure 8: Git Branches Concept [19]

However, there is still a need for clear communication between team members when working on the same part of a file as conflicts will arise when merging code. The team would have to resolve the conflict by deciding which commit to use and updating the main branch appropriately. Lack of transparent communication and thorough discussion within the team would give rise to numerous errors and impede the pace of development.

## 2.5 Stages of Software Development

The 7 stages of software development are planning and analysis, defining requirements, design, development, testing, deployment and maintenance. The is just a general framework of development and can vary based on the type of project and the style of development such as agile or waterfall. For this project, only the first 5 stages will be focused on [20].

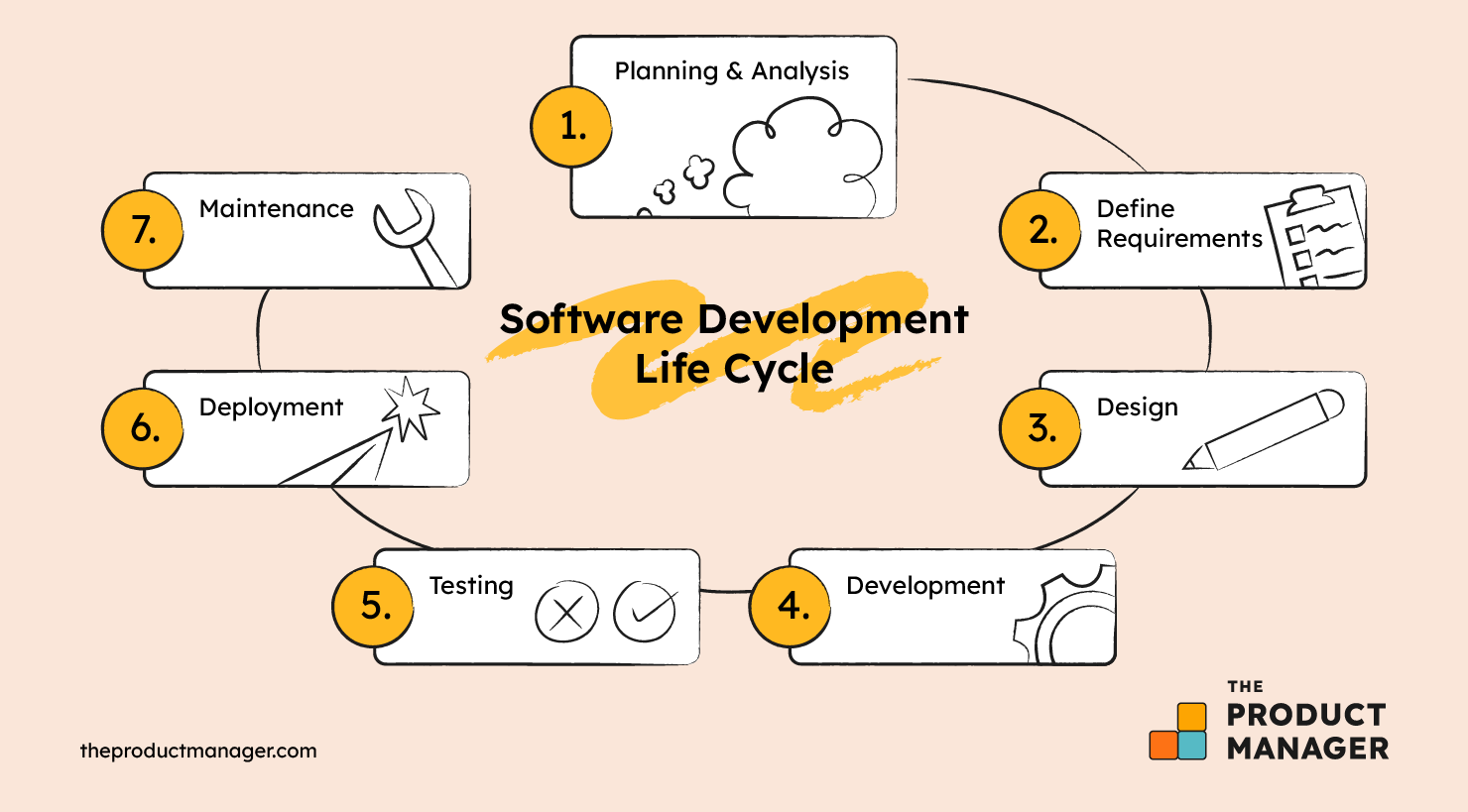


Figure 9: Stages of Software Development

**2.5.1 Planning and Analysis**

This initial stage is to identify the problem that the software is aiming to solve. Planning during this stage helps to ensure alignment to resources and requirements of the project. Extra information is also gathered for the project such as data on target audiences or initial research into the problem [21].

**2.5.2 Define Requirements**

During this stage, the information collected in the initial phase is transformed into precise and detailed requirements for the development team, delineating key features and additional functionalities. This process results in the creation of critical documents such as the Software Requirement Specification (SRS), Requirement Traceability Matrix, and Use Case Document.

**2.5.3 Design**

Teams will decide on what platforms to use to tackle problems such as what programming language or what collaboration to use. Development of a prototype is done at this stage to give the team and client a visualisaton of the product to be developed. A prototype holds importance as it facilitates rapid modifications without needing alterations to the underlying codebase, thereby conserving time and resources. Colour themes and a design system are developed here to ensure consistency throughout all the different parts of the product.

**2.5.4 Development**

Actual development of the product begins here, and members of the team will be assigned different parts of the product to develop. This phase can take the longest time as there may be unexpected issues or bugs to be fixed. Hence, it is important to establish a timeline to indicate the expectations for the product among developers, while concurrently setting milestones to monitor the progress of the product. This stage can run concurrently with the testing stage to ensure there are no critical errors or bugs to save time.

**2.5.5 Testing**

An important stage before shipping the product, vigorous testing is needed by the quality assurance team. The product must have met most, or all the requirements set out in the beginning and is functioning as expected. This stage also can reveal any UI, functionality or security issues with the product and can be sent for improvement. There are many different types of testing such as Functional Testing, Security Testing, Usability Testing, etc.

## 2.6 Database Research

A database is required for this project to store different types of data such as user data, exam data, results data and many more.

The subsequent section shall show the attributes for an exemplary database. Thereafter, an exploration of existing databases, comprehensively evaluating their strengths and weaknesses. This thorough examination is conducted to ascertain the optimal database solution for this project.

**2.6.1 Design Principles and considerations**

Well-performing databases are ACID Databases: ACID stands for Atomicity, Consistency, Isolation and Durability. Examples of this include MySQL, PostgresSQL and Oracle [22].

* **Atomicity**: It means that in a group of queries, if one fails then the whole transaction will fail, the request will not be processed, and the database will return to its previous state. This allows quick debugging and identification of the queries with the problem.
* **Consistency**: In the context of ACID means data consistency, called referential integrity. It is a way of enforcing the consistency between tables through foreign keys.
* **Isolation**: Ensures that simultaneously executing transactions does not disrupt one another. Concurrency in this context pertains to multiple transactions attempting to alter or access the same database record(s) concurrently.
* **Durability**: Ensures that alterations enacted by a committed transaction remain intact and are not subject to loss. All committed transactions must be stored persistently on non-volatile, durable storage. This safeguards the integrity of committed transactions, even in the event of a database crash.

**2.6.2 Firebase**

Firebase is a database hosting service by google and it is easy to use and setup. It also has a built–in authentication and session feature which would save us a lot of time. The data is stored in ‘documents’ under ‘collections’ and each document is like an object with different properties. Even though this database provides a quick and easy start, it does not offer relational data to be stored, it is also a NoSQL database.

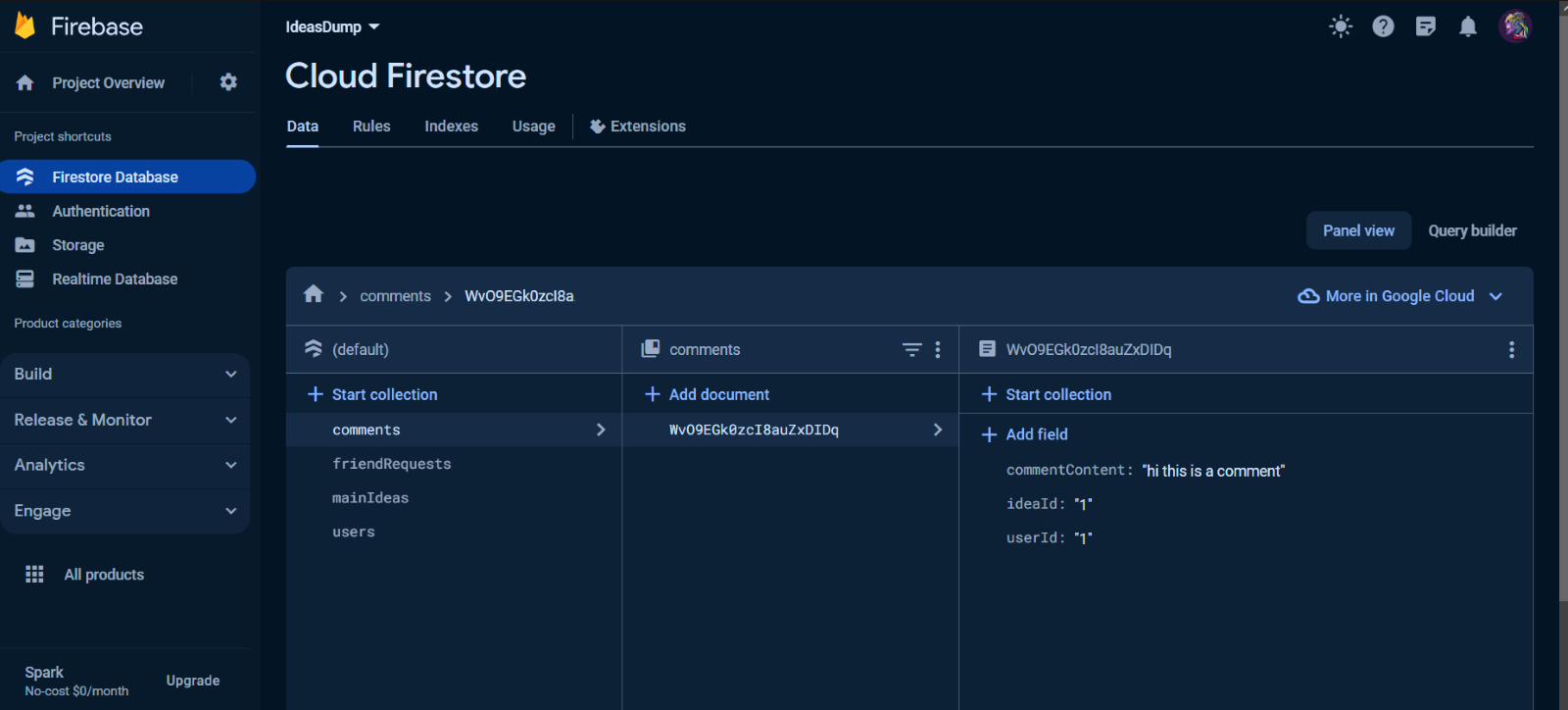


Figure 10: Firebase console and examples of data stored from testing.

**2.6.3 Hygraph**

Hygraph functions primarily as a content management tool but also offers capabilities for data storage and can host a database. It utilizes GraphQL to query and mutate data, classifying it as a NoSQL database. GraphQL is an efficient method for data retrieval, thanks to its straightforward syntax and the structuring of returned data in objects, which can be advantageous for specific applications. However, it is important to note that this setup does not provide the features of a relational database.

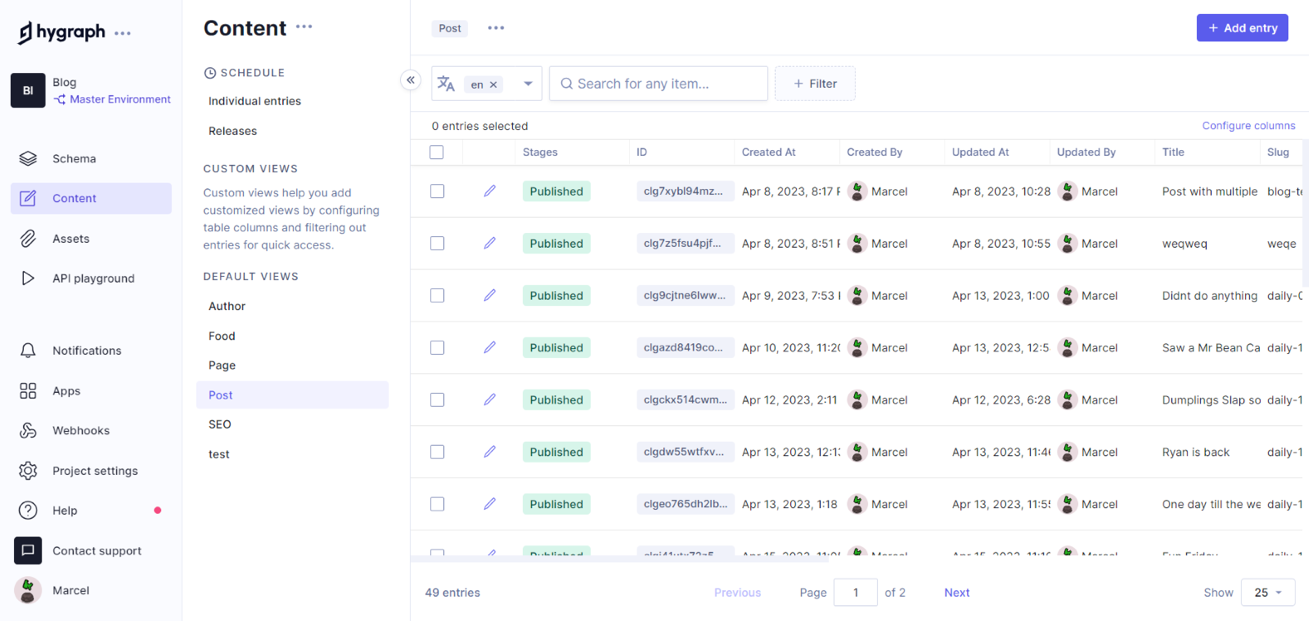


Figure 11: Home page which allows you to post data with a certain format.

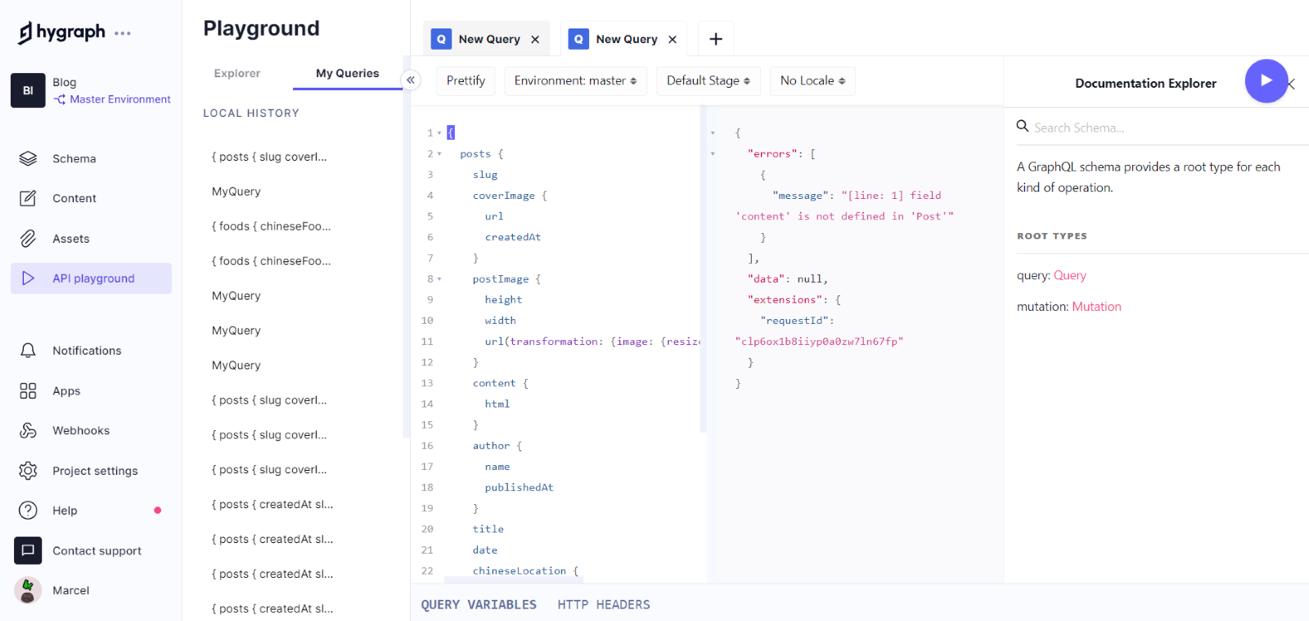


Figure 12: GraphQL in action used for querying or mutation data in the database.

**2.6.4 PostgreSQL**

PostgreSQL is a powerful, open-source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. It is ideal for us as it is an SQL database and it allows for relational data, and it is open source and free. Another alternative considered was MySQL, however it has many user complaints, and one critical difference is that PostgreSQL can store arrays, which is critical to the project.

**2.6.5 Amazon Web Service (AWS)**

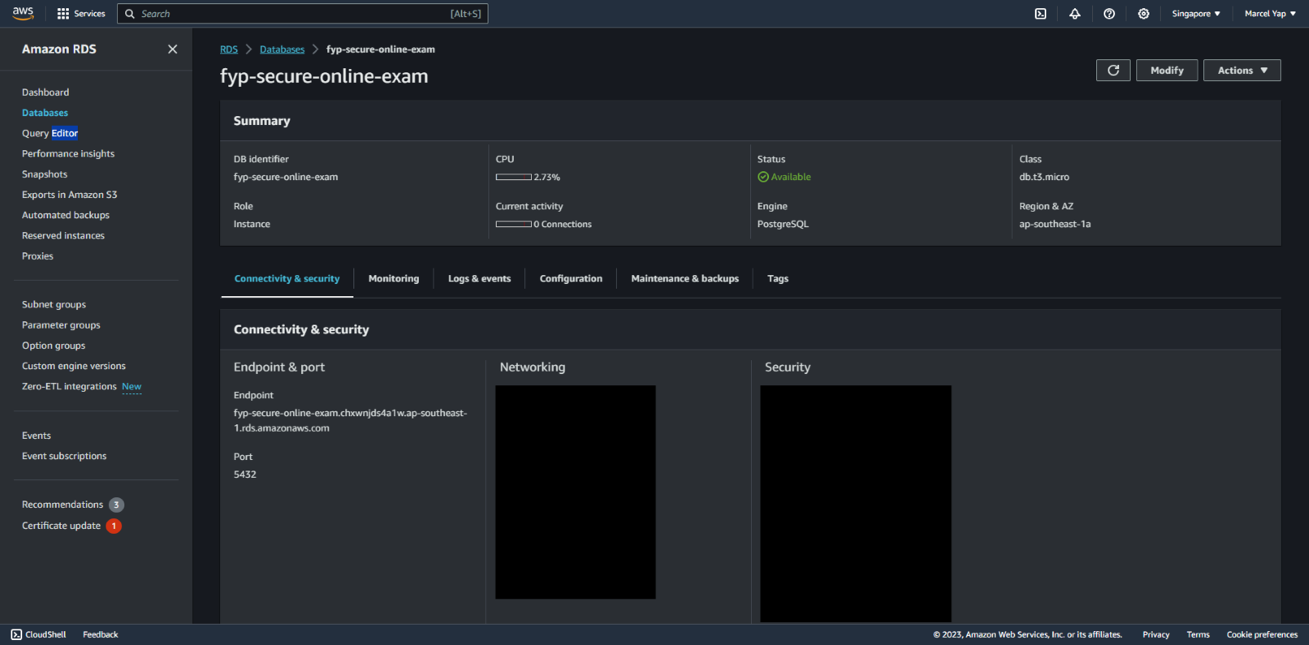
Amazon Relational Database Service (Amazon RDS) is a collection of managed services that allows for operating and scaling databases in the cloud. It supports Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server and provides hosting services with Amazon RDS on AWS Outposts [23].

Figure 13: AWS MySQL Database Home Page

AWS is known to have a reliable and stable service, but it is important to note the bandwidth and usage limit for the free tier of the database. Overuse could very easily incur charges. However, it is a flexible service that provides many different options for databases and hosting.

**2.6.6 Database Management System**

A database management system (DBMS) is software used to define, manipulate, retrieve, store and manage data in databases, it also helps the user to visualise the data in the database. Some popular database management systems are DBeaver and phAdmin4 [24].

DBeaver Community is a free cross-platform database tool for developers, database administrators, analysts, and everyone working with data. It supports all popular SQL databases like MySQL, MariaDB, PostgreSQL, SQLite, Apache Family, and more [25].

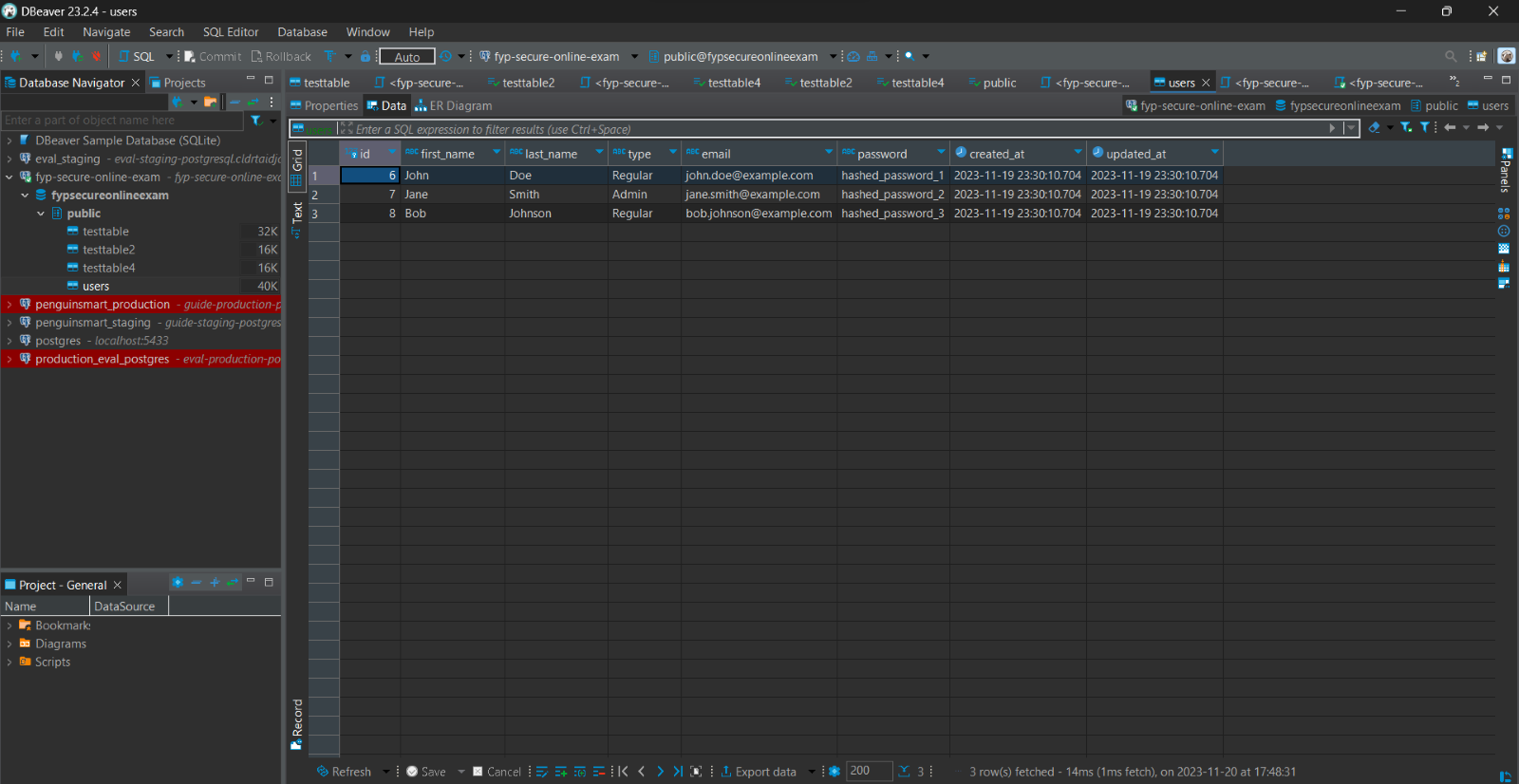


Figure 14: DBeaver - User Table view and sample data.

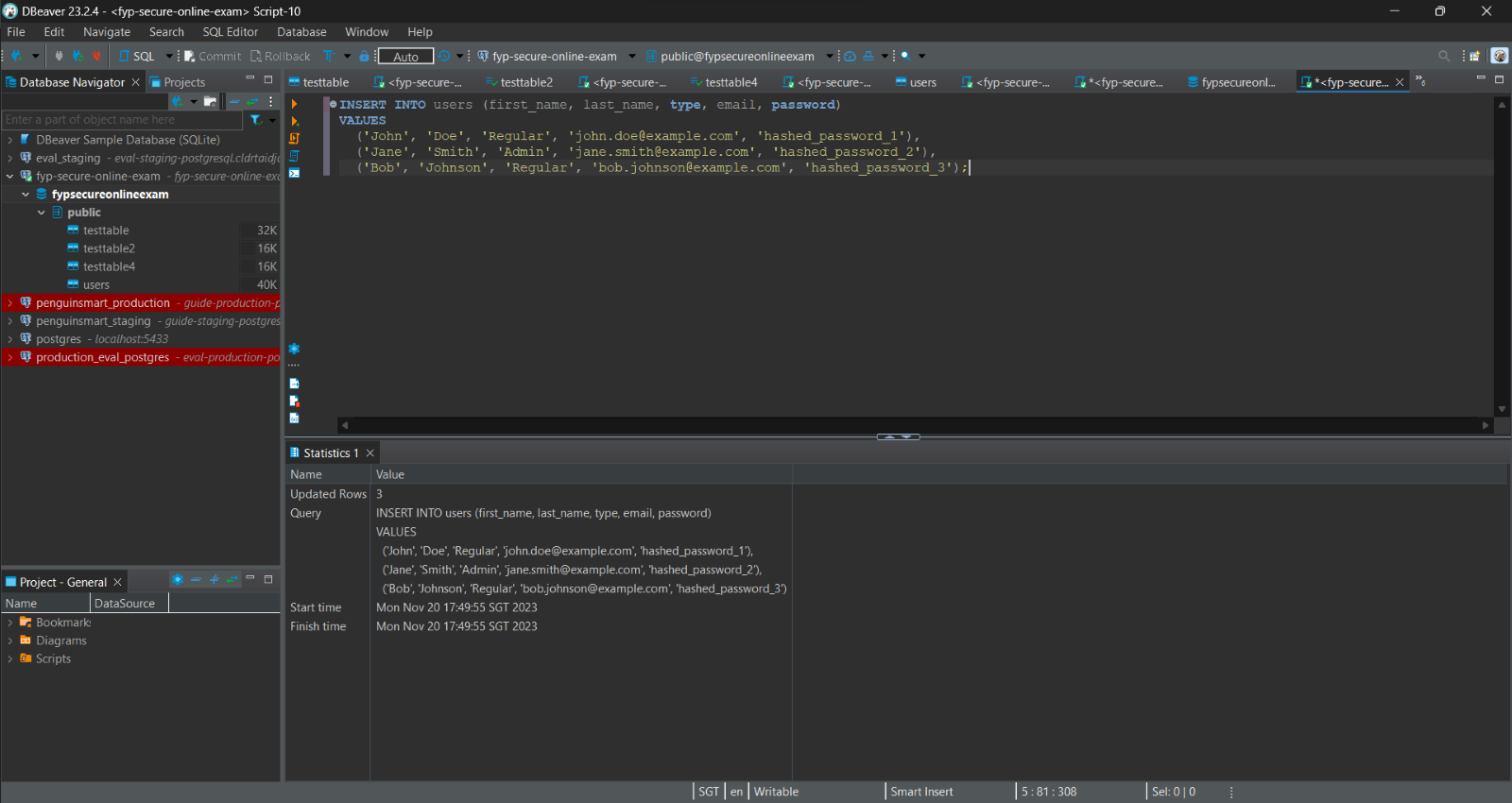


Figure 15: DBeaver - Using SQL script to insert data into table.

pgAdmin is a free software project released under the PostgreSQL licence. The software is available in source and binary format from the PostgreSQL mirror network [26].

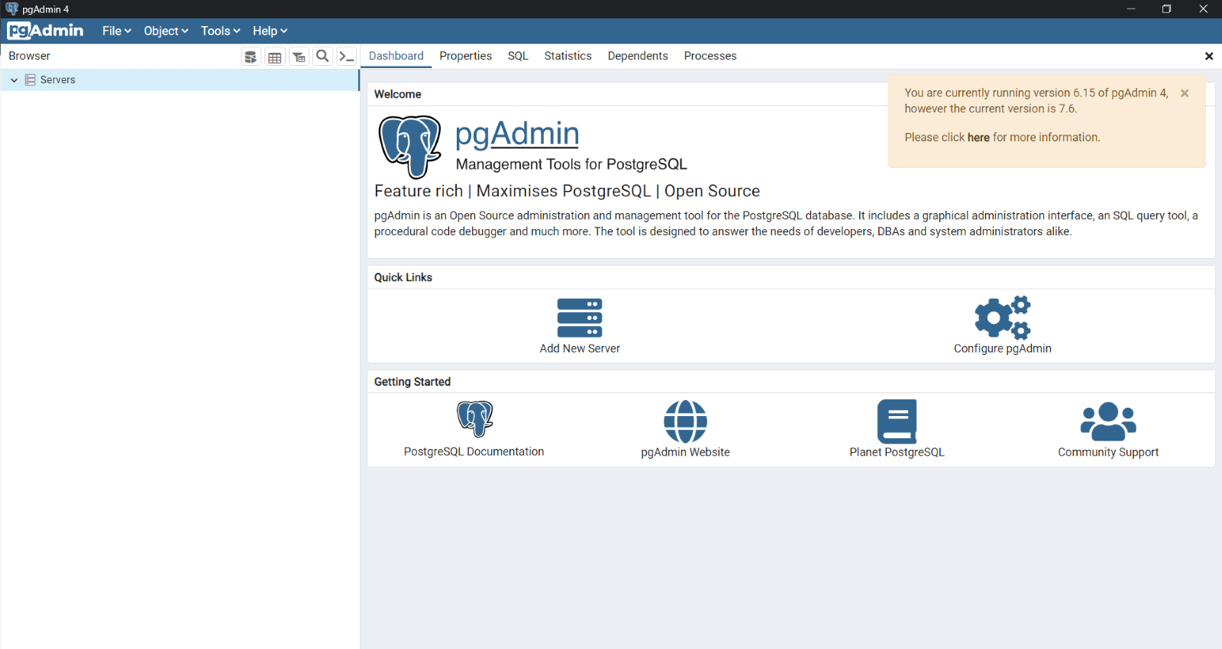


Figure 16: pgAdmin Home Page

**2.6.7 APIs**

An API (Application Programming Interface) functions as a software intermediary facilitating communication between separate applications. APIs provide a convenient mechanism for extraction and dissemination of data, facilitating internal operations within organizations and external interactions across diverse entities. [27]

## 2.7 Design System and User Interface

Students are easily affected by the environment around them during times of high stress such as during examinations. Various factors including psychological, physical, socio-economic, and educational aspects significantly impact their performance in examinations at the university level. Alterations in the format of examination papers just before exams can also influence students' academic performance [28].

A pleasing and aesthetic design of a software can trigger positive emotions in a user. An interface which is easy to use and requires minimal effort creates satisfaction and calmness [29]. Thus, this will give students peace of mind during a stressful examination, allowing them to answer to the best of their ability.

**2.7.1 Colour psychology**

When determining the colour scheme for the app, extensive research was conducted on various colours and their respective impacts on individuals. [30]

* Blue has calming properties and is widely regarded as the most soothing hue. Within therapeutic environments, the utilization of primary blue promotes meditation and relaxation.
* Purple induces a sense of calm and relaxation like blue. Lighter variations of purple often create feelings of hopefulness, while deeper, more saturated tones create sensations of power or strength.

**2.7.2 Usability heuristics**

Usability heuristics are 10 guidelines created by the Nielsen Norman Group to ensure user interface design is straightforward, user-friendly and comprehensive. These guidelines are used when building new programs, mobile apps or websites to ensure a smooth user experience. [31]

1. **Visibility of System Status:** Users should be constantly informed, or able to get the system status within an appropriate amount of time so that users can know the outcome of their interactions.
2. **Match Between the System and the Real World:** The design should use simple words and phrases which are familiar to the user, parallel to physical designs. Ensure information appears in a natural and logical order.
3. **User Control and Freedom:** Ensure there is an easy way to undo an action as users will often do an unwanted action. This will instill a sense of freedom and confidence in the design.
4. **Consistency and Standards:** Due to Jakob’s Law, which states that people use other platforms more than yours [32], the design should follow industry standards and the design should met users’ expectations.
5. **Error Prevention:** Prevent problems from occurring in the first place, by either eliminating error-prone conditions or creating a form of confirmation for the user.
6. **Recognition Rather than Recall:** Users should not have to memorise information from one part of the interface to another as humans generally have short term memory. Information required by the user should be visible and easily retrievable.
7. **Flexibility and Efficiency of Use:** The design should cater to experienced and inexperienced users. An example of this would be to create shortcuts that only experienced users would know about. This allows flexibility and users can use their own methods to use the interface.
8. **Aesthetic and Minimalist Design:** The design should not contain unnecessary or irrelevant information as any bit of information displayed will compete with other more important bits of information. The visual elements should still support the user’s primary goals.
9. **Help Users Recognize, Diagnose, and Recover from Errors:** Error messages presented to the user should be precise and suggest solutions for the user to solve the error. The elements of the error message should be clear so the user will notice them.
10. **Help and Documentation:** The best designs do not need any additional explanation, however if documentation is needed to help the user complete their task, then it should be easily accessible and list concrete steps on how to do so.

These guidelines are fundamental to the design of a new app and will be instrumental in the creation of the app's design system. [33]

**2.7.3 Figma**

Figma is a widely-used cloud-based design tool that facilitates team collaboration. It offers an intuitive interface for rapid prototyping and also includes features for creating high-fidelity designs. Additionally, Figma features a developer mode that converts designs into CSS code, simplifying the development process. [34]

## 2.8 Front End Technologies

To implement the design of the app, front end technologies are needed, which are frameworks and tools used to develop the user interface of web pages and apps. Some popular front-end technologies include React.js, Angular.js , Vue.js, Flutter, Bootstrap, HTML and CSS and many more.

**2.8.1 HTML**

HyperText Markup Language (HTML) is code which is used to develop the basic structure of a website or app. It creates elements such as paragraphs, input boxes, labels and many more. An element in HTML consists of an opening tag and a closing tag, the content to be displayed is put in between the 2 tags. Depending on the type of element selected, the content will be displayed differently based on the needs of the developer. [35]

**2.8.2 CSS**

Cascading Style Sheets (CSS) is used in combination with HTML to style the elements created. CSS can be used to change the text colour, background colour, font size, font family, create animations and many more. Elements in HTML are given a class, which can be referenced in CSS to edit their styling.CSS is vital to achieve the design of a website. [36]

**2.8.3 JavaScript**

JavaScript is a popular language that can be used for various purposes including web development. It lets you add interactivity and dynamic data loading when used together with HTML and CSS. JavaScript is also used for backend applications, used to send and receive data from databases and communicating with servers. [37]

**2.8.4 ReactJS**

ReactJS is a component-based library deployed for the development of interactive user interfaces using Javascript. It is supported by Facebook, Instagram and a community of individual developers and organisations. React essentially empowers the creation of expansive and complex web-based applications capable of dynamically changing their data without needing subsequent page refreshes. Creating and exporting components with React is easy and the power of reusing premade components for different pages saves time and resources. For these reasons it is one of the most popular front-end frameworks. [38]

**2.8.4 React Router**

React Router is a JavaScript framework facilitating the management of client and server-side routing within React applications. It facilitates the development of single-page web or mobile applications, permitting seamless navigation without the need for page refreshing. [39]

# Chapter 3: Design

This chapter will explore the design and implementation of the app with reference to the research conducted in literature review.

## 3.1 Features Required

Since this report focuses on the admin aspect of the project, the features required are as follows:

1. Control over examinations in progress.
2. Management of personnel in the app, such as account registration and editing details of personnel
3. Help with questions or issues via messaging.
4. Management of modules / classes that student or instructors are a part of.

Standardised components are also part of the requirements to make it easier for future use to create more pages and functions.

## 3.2 User Journey

A user journey is a sequence of steps that the user takes in order to accomplish a task or a goal on the app or website. This can be mapped out to better understand user behaviour and thus design an app which can help the user achieve their needs as quickly and painless as possible.

**3.2.1 Site Map**

A site map is a visualisation of all the possible pages the user can access on the website, it is helpful in the planning phase to ensure that all requirements are met.

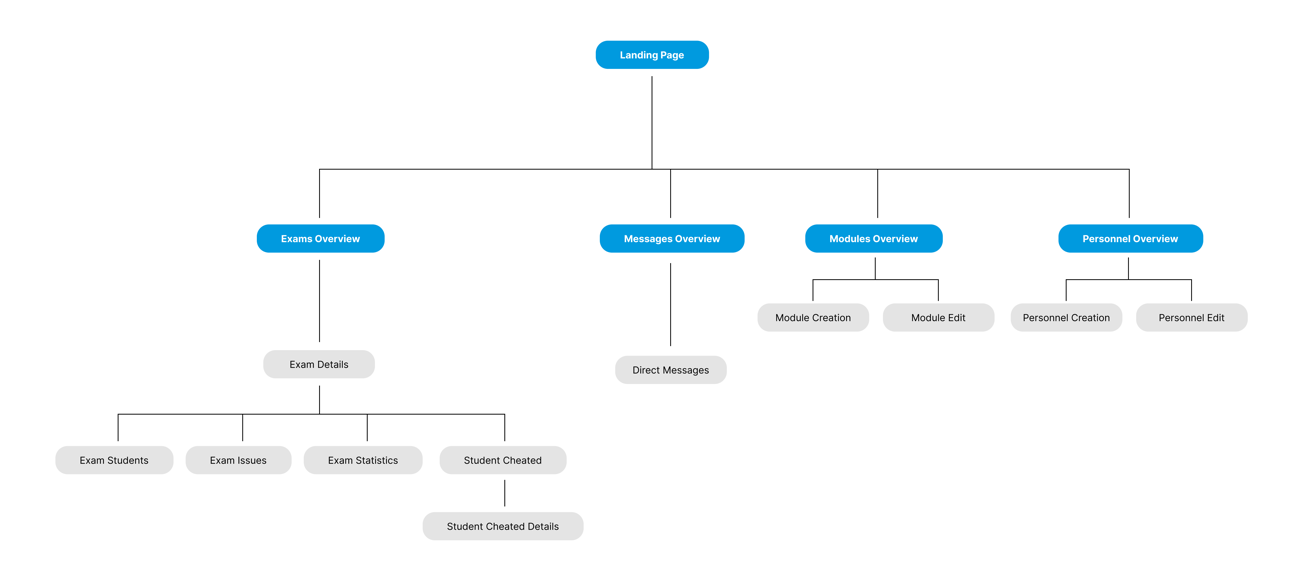


Figure 17: Site Map for Admin

The user will still be able to access all the different pages with a navigation bar at the left side of the screen. Each page will also have a “back” button to ensure that the user can easily navigate around the different pages.

## 3.3 Design System

This section will elaborate on the design of the app itself based on research conducted in literature review. The objective is to create a minimalistic, clean and modern design for ease of use.

**3.3.1 Colour Theme**

Deciding on the colour theme is very important as it determines the emotions evoked when the user is using the app.

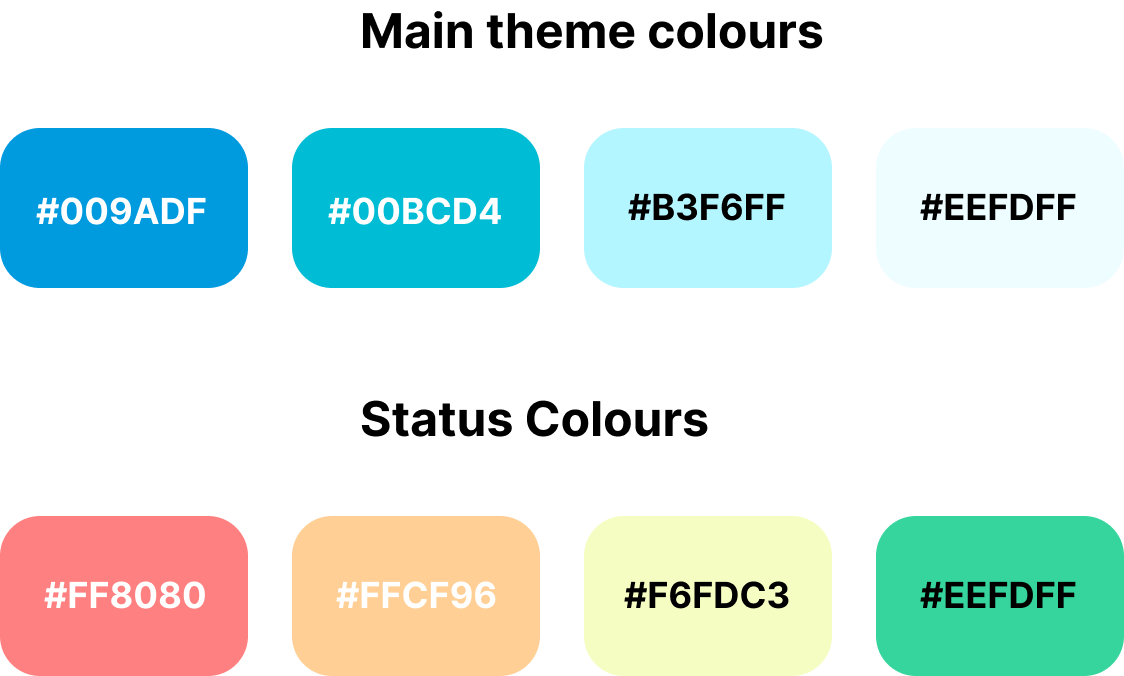


Figure 18: Colour Theme for App

Blue was decided as the main theme colour for the app as it has calming properties and it is not an attention seeking colour, thus it would prevent distractions during examinations. The status colours were not sharp colours and were instead muted to prevent further distraction. These colours also follow a minimalist scheme.

**3.3.2 Typography**

To ensure an optimal user experience, it is important that users can readily and quickly access information needed. Clear typography plays a pivotal role in facilitating this. San serif fonts can create a modern, clean, and minimalist look in the design. Sans-serif fonts have the capacity to imbue designs with a contemporary, sleek, and minimalist appearance. Their inherent readability makes them a preferred choice over serif fonts for facilitating comprehension and aesthetic coherence.

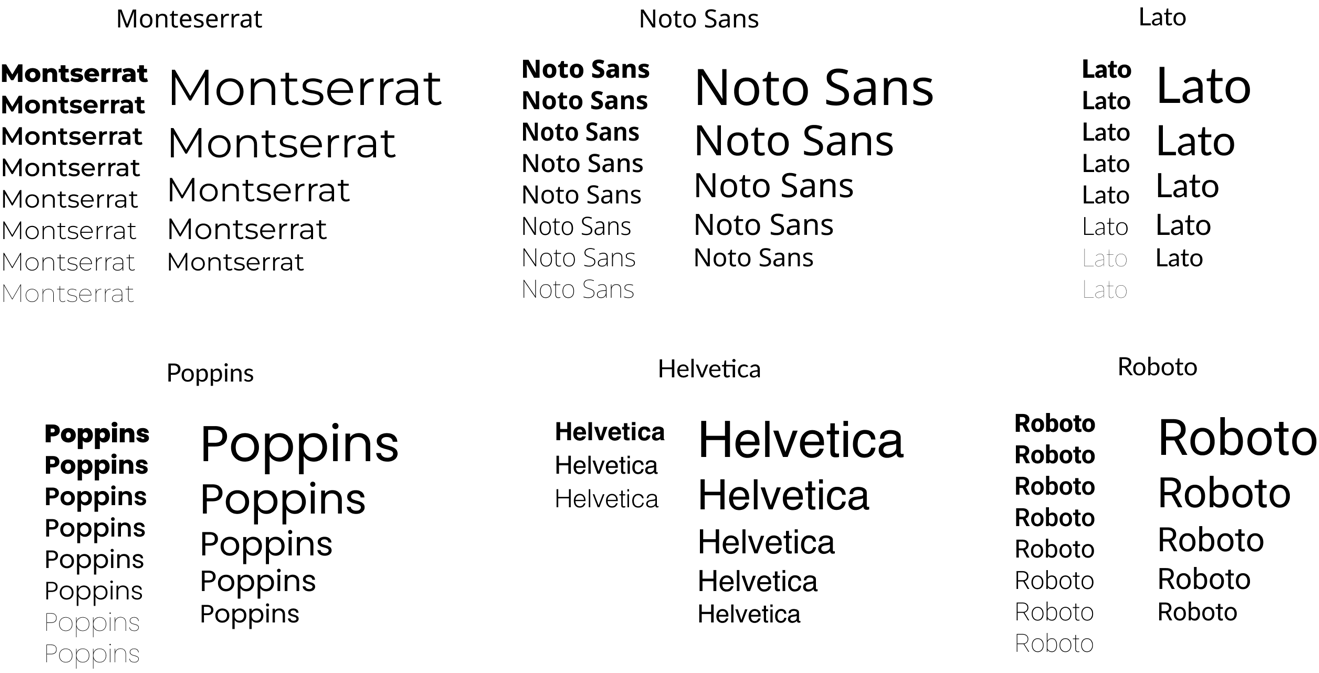


Figure 19: Selection of Shortlisted Fonts

Figure 19 shows the different shortlisted fonts found online, the left column shows the font in different font weights while the right column shows the font in their different font sizes for comparison. Poppins was chosen as the final font choice due to its uniqueness yet clear readability and will be used for the whole app.

**3.3.3 Logo and Name**

The logo and name of the app are key parts of creating a new brand or app, it helps users easily identify the app when needed. The name decided upon is “ExamPulse” to give an impression of a fast, efficient and effective service. The logo is a graduation cap due to its symbolism with achievement, credibility and education.

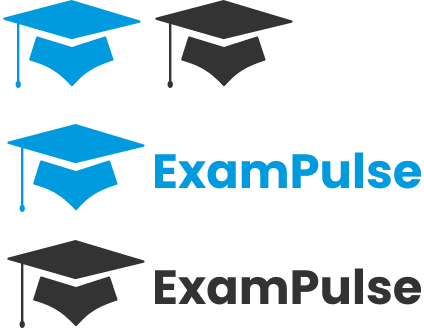


Figure 20: Logo and Logo with name.

The logo is made with the primary theme colour of the app and a full black version is made to for potential use across various sections within the app.

**3.3.4 Implementation in Figma**

A design system is needed in any new app design to ensure consistency throughout the app. Different styling throughout the app would cause distractions and confusion to the user, which has more impact on the student taking an exam. Figma has features to help with the implementation such a colour library where the colour themes can be added and easily accessed to ensure all designers use the correct colours.

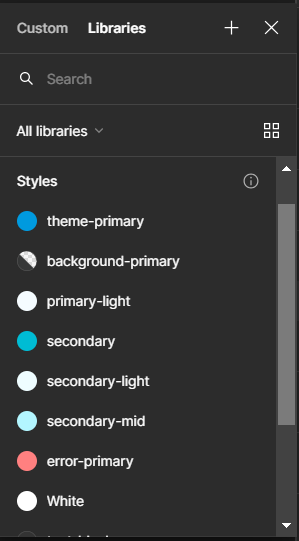


Figure 21: Figma Colour Library

## 3.4 Wireframing

Wireframing is creating and designing low fidelity designs to visualise the layout of elements of the app. Wireframing is important as changes can be made quickly and effortlessly since the elements do not have much detail on them, this was used during the initial design stages of the app.

**3.4.1 Components**

Individual components of UI elements were created to make creation of pages easier and ensure consistency.

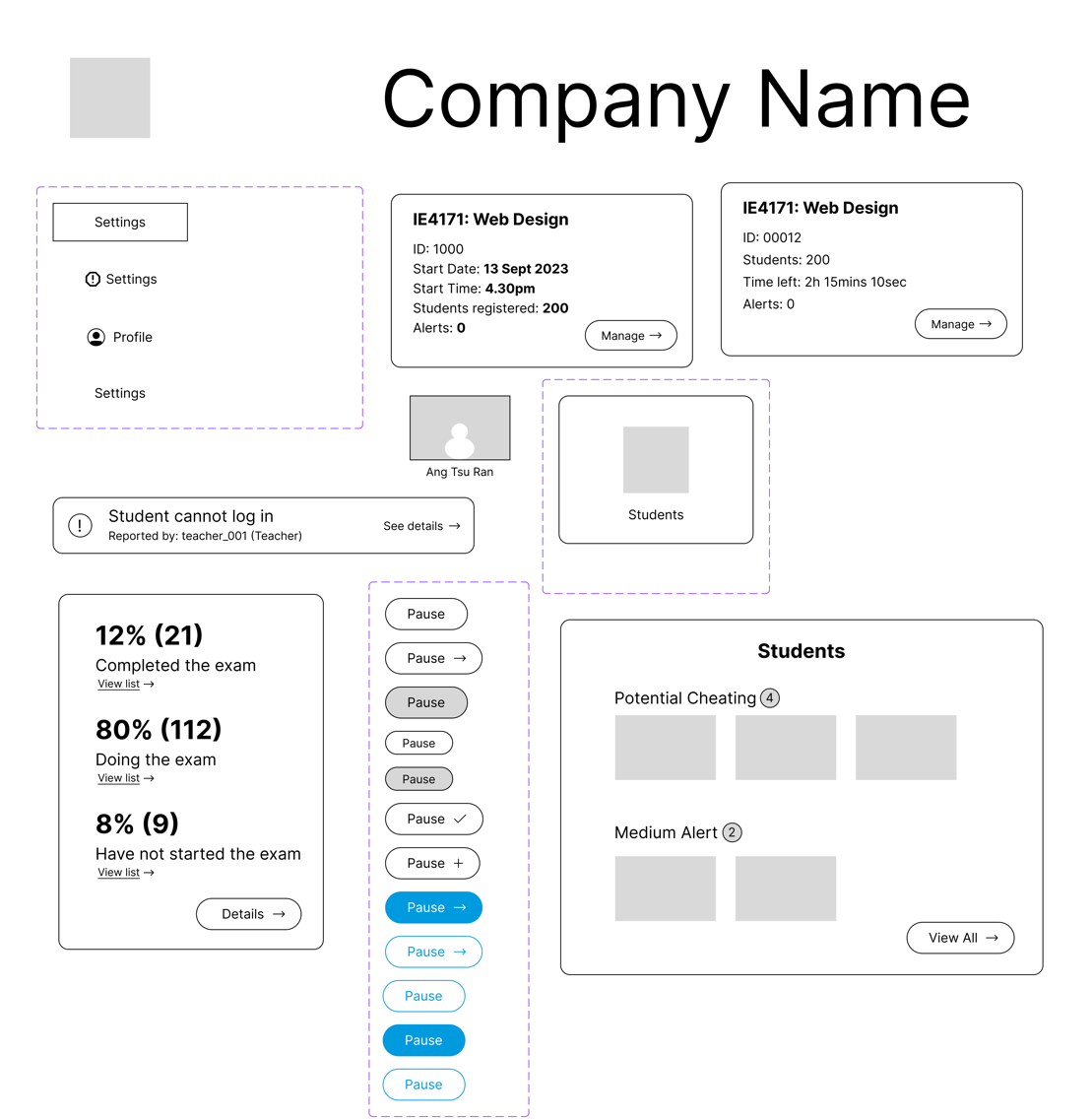


Figure 22: Partial list of components for wireframing.

Variations can be created from components such as different coloured buttons for different use cases throughout the app. A single change to the main component will trigger the same change for all the instances of that component, making it extremely easy commit changes.

**3.4.2 Pages Wireframe**

Using the components created, the layout and basic structure of the different pages of the app can be created.

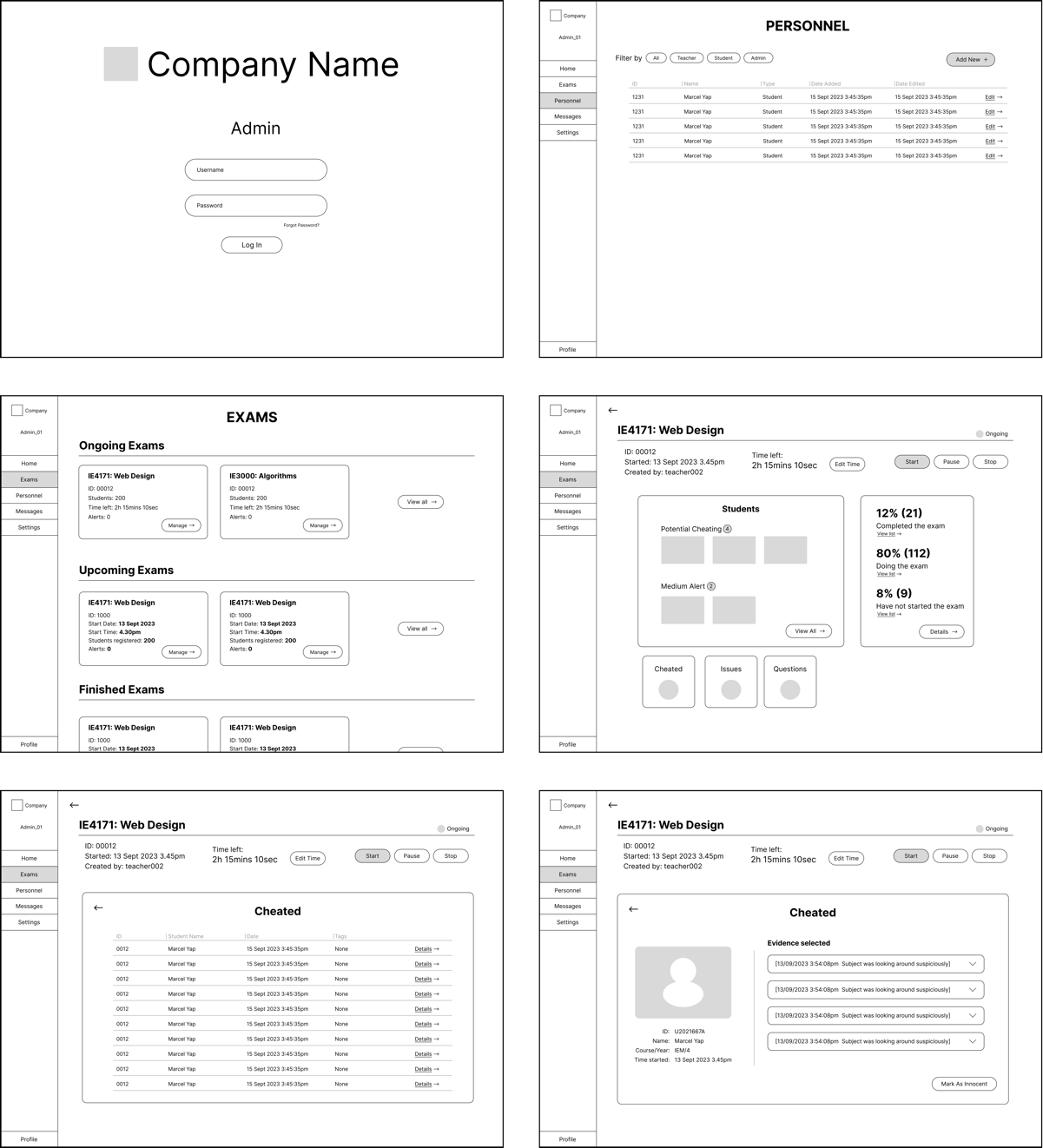


Figure 23: Admin pages wireframe.

After the wireframe is completed, a high-fidelity design can be created, which includes adding more detail, colours, typography and design style to the wireframe. Once the high-fidelity design is completed, it can be referenced to while developed by the front-end developer.

## 3.5 Components used for development

This table outlines the overall framework and technologies employed in the development of the app.

|  |  |
| --- | --- |
| **Component** | **Description** |
| Front-end Framework | React |
| Styling | Styled Components |
| Routing | React Router |
| Icon Library | React Icons |
| Back-end | Firebase |

Table 1: Development components

# Chapter 4: Implementation

## 4.1 Folder Structure

The folder structure of this project is outlined in Table 2. This framework is designed to segregate distinct facets of the project, distinguishing the backend from the frontend components.

|  |  |  |
| --- | --- | --- |
| **No.** | **Folder Name** | **Description** |
| 1 | src/pages | All pages, further spilt into faculties |
| 3 | src/components | Common components |
| 4 | src/backend/firebase | Configuration, queries and functions |
| 5 | public | Icon and index.html |

Table 2: Project Folder Structure

## 4.2 Database

This section will show the details of the database design, including its configuration and implementation using Firebase.

**4.2.1 Database Design**

Database design is important at the start of a full stack project to properly map out the different data structures needed in the database. Based on the administrator aspect of this project, 4 main collections are needed to represent the data needed for the features. The relationship between the different collections can be illustrated with an Entity-Relationship (ER) diagram.

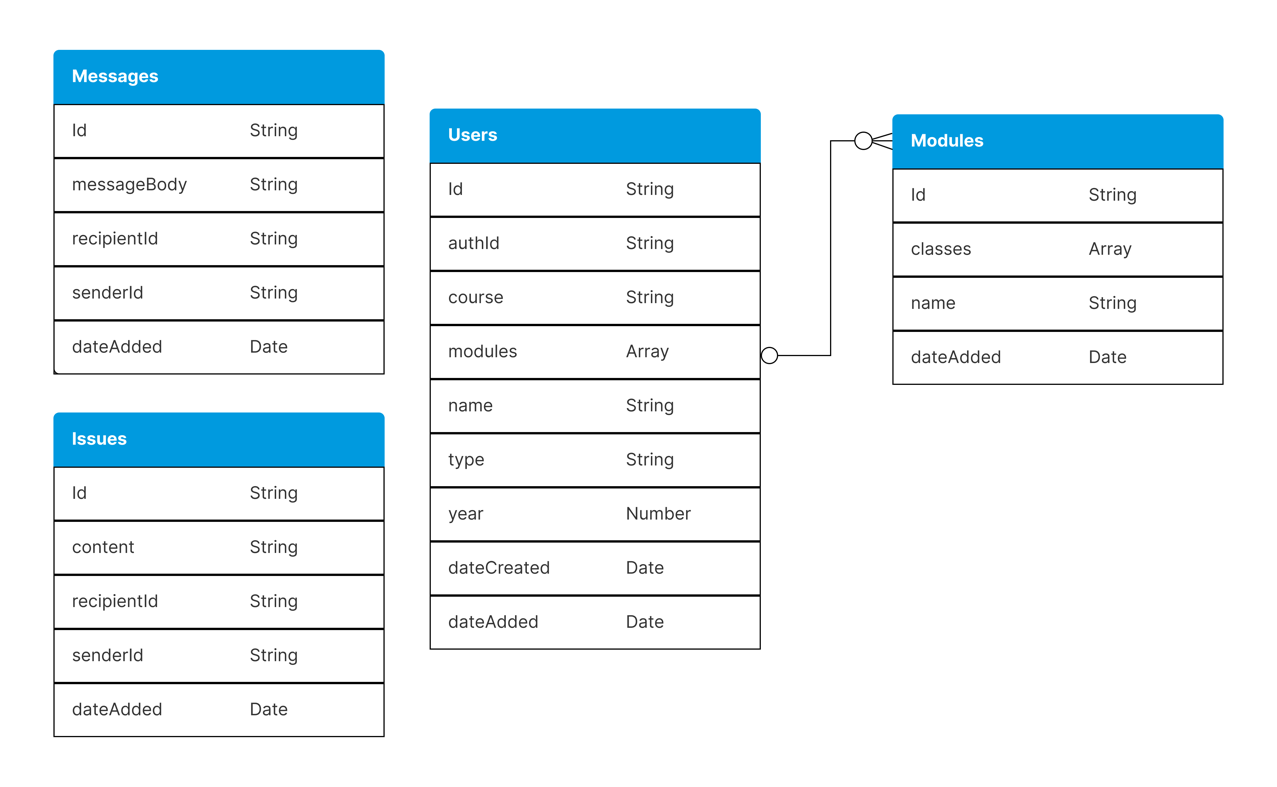


Figure 24: ER diagram.

In the user collection, authId acts as a foreign key to the authentication table provided by Firebase, which stores user account details.

**4.2.2 Firebase Implementation: Firestore Database**

Firebase consists of many different parts, however only 3 will be used for this project:

1. Firestore Database: The main database where collections of data are stored and accessed.
2. Firebase Authentication: A system which manages user accounts and stores their login information.
3. Firebase Storage: A storage system for images, videos and files which can be accessed by the other parts of Firebase.

The creation of collections can be done in code or using the Firebase GUI, the latter was used for the initial set up of the collections. If a collection referenced in code does not exist in the database, then a collection of the same name will be created. Seed data was added to initialise the database for testing to check if data can be pulled and sent.

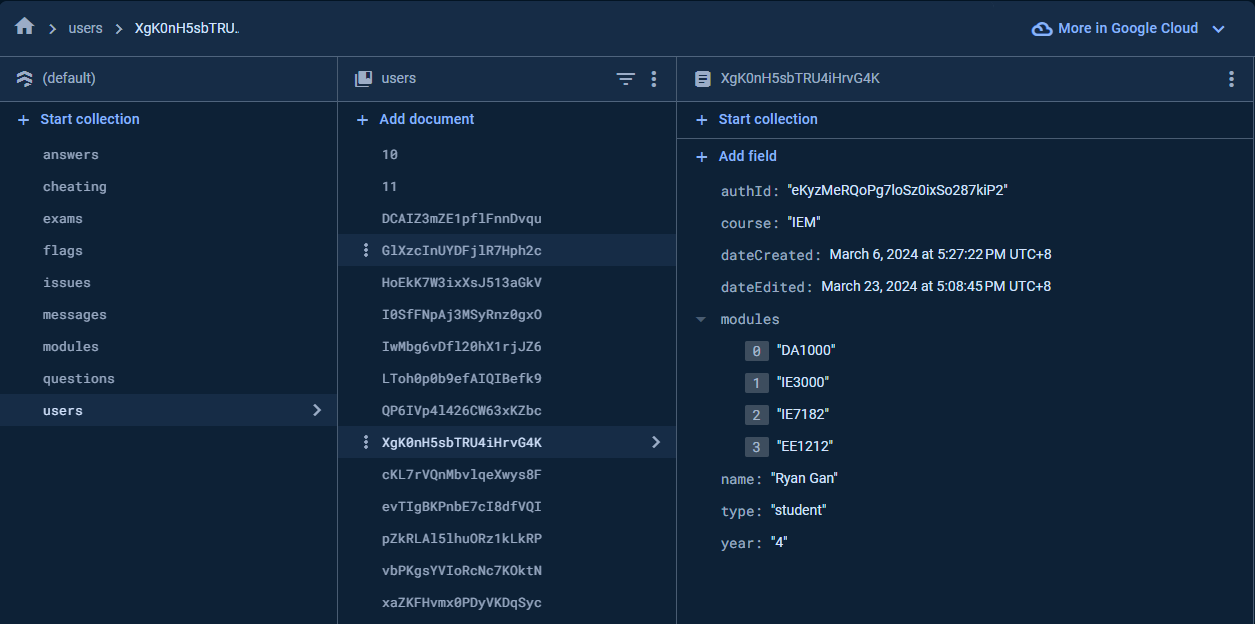


Figure 25: Firebase GUI for users’ collection after initalisation.

Each entry in a collection, or row for SQL, is called a document in Firebase and each document can have multiple properties, or columns in SQL. Likewise with collections, this can be achieved either in code or using the Firebase GUI. Each document can be edited to have multiple properties of different data types such as array, string, etc, according to the database design.

**4.2.3 Firebase Implementation: Authentication**

Authentication within Firebase comprises a user-friendly and resilient system which, once initialized, furnishes an account infrastructure to any application. It offers a variety of signup and sign in methods, including email authentication, Google Single Sign-On (SSO), among others. The provided methods, exemplified by createUserWithEmailAndPassword or signInWithEmailAndPassword, are easily implementable and facilitate seamless user interaction.



Figure 26: Preview of user account in Authentication.

When a user opts for email signup, while the associated account email can be seen in the database, the user's password undergoes encryption, rendering it inaccessible to any party. Additionally, each user account is assigned a unique User UID, herein referred to as 'authId' within the user collection.

**4.2.4 Firebase Implementation: Storage**

Firebase Storage constitutes another facet of the Firebase platform dedicated to the storage of media and files. Upon storage, each file is automatically allocated a unique Uniform Resource Locator (URL), facilitating its retrieval from the storage repository and subsequent display.

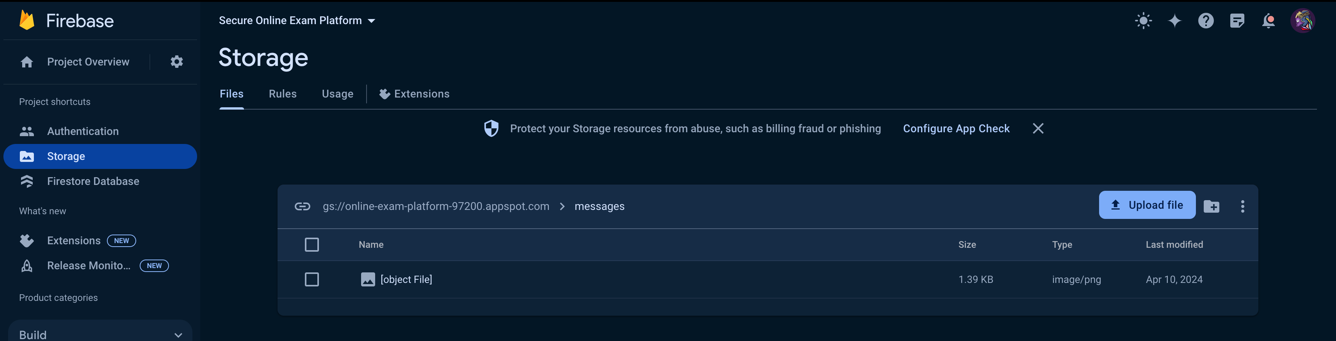


Figure 27: Example of file in Storage.

## 4.3 Back End Implementation

Prior to commencing the front-end development phase of the application, it is imperative to initiate the back-end implementation to validate the functionality of the application before embarking on the aesthetic aspects. This segment will delve into the utilization of React to establish connections with different components of Firebase, facilitating the execution of fundamental CRUD operations (Create, Read, Update, and Delete)

**4.3.1 Configuration**

The initial step involves creating a file named `firebase.js` to house all database configurations, which are then exported for easy integration across the app. This configuration file includes essential parameters such as the database’s API Key, authDomain, projectId, and more, which facilitate connection to the database. Following the configuration setup, the Firebase app is initialized, and various services such as Firestore, Storage, and Auth are exported for use in other parts of the application.

The initial procedure entails the creation of a file named firebase.js, designated for the storage of all database configurations, thereby enabling their integration throughout the application via straightforward imports. These configurations include essential parameters such as the database's API Key, authDomain, projectId, among others, for establishing connectivity with the database. Subsequently, the Firebase application is initialized, and its components including firestore, storage, and authentication are exported for utilization in other sections of the app.

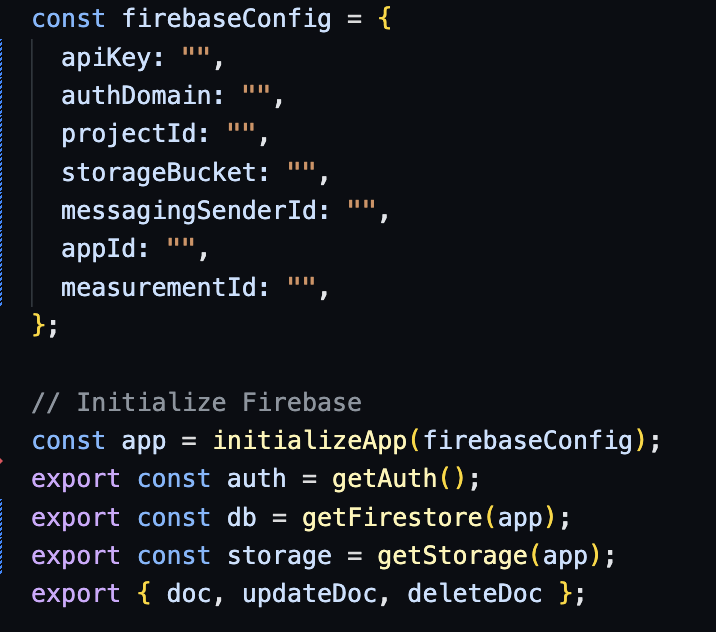


Figure 28: Configuration of Firebase.js file.

**4.3.2 CRUD Operations**

These constants can be imported to query the database and perform CRUD actions.

**4.3.2.1 Create**

Initially, it is important to establish a reference to the collection where the data is to be stored; in this instance, the data is being appended to the "messages" collection. Subsequently, the function "addDoc" is employed, facilitating the adding of the data to the database, typically encapsulated within an object featuring multiple properties.



Figure 29: Example of adding data to database.

As depicted in the figure above, db is imported for initialization purposes, serving to direct operations towards the designated database. 'MessageDocRef' serves as a receipt upon the transaction's culmination, granting access to its properties, notably including the ID of the appended document. Encapsulating the transaction within a 'try' statement ensures comprehensive error handling, guaranteeing that any encountered errors are captured and appropriately displayed.

**4.3.2.2 Read**

Reading data is arguably as crucial as creating data. A frequently utilized function in the app involves retrieving user data from the database. This is achieved by creating a reference to the relevant collection, which is then used in a manner similar to how data is created.



Figure 30: Example of reading data.

Await is needed to ensure the query is completed before setting the state of currentUserData, which is a variable in this page. Specific queries can be created to query for a user with a specific authId or users with type “admin”. These queries are constructed similarly to an SQL query.



Figure 31: Example of using query

The results of the query are saved into a querySnapshot, which stores all the data in an array, which can be assigned to another variable for use.

**4.3.2.3 Update**

The process of updating documents involves the utilization of the 'updateDoc' function provided by the Firebase API. A reference to the collection is also needed. The input parameters for this function include either a new object intended to replace the existing one entirely, or alternatively, the utilization of a spread operator to selectively update one or more properties within the original object.

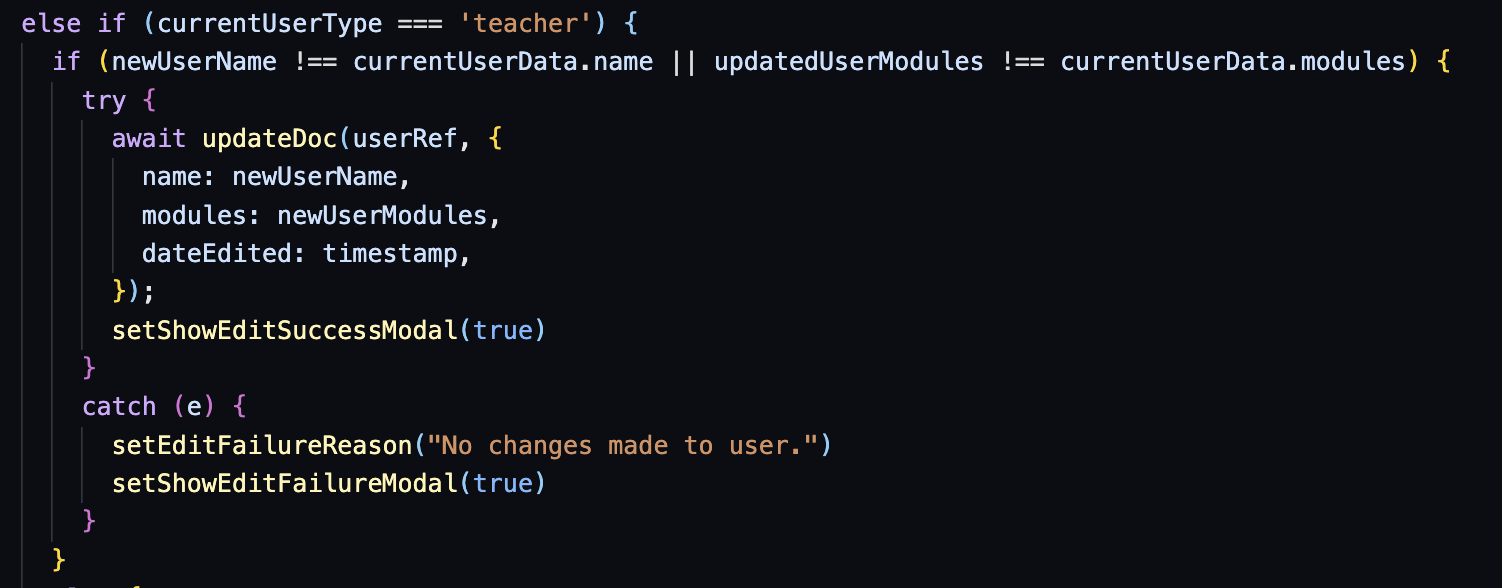


Figure 32: Example of updating data.

**4.3.2.4 Delete**

To delete a document, a reference to the document’s id is needed followed by using the deleteDoc function.

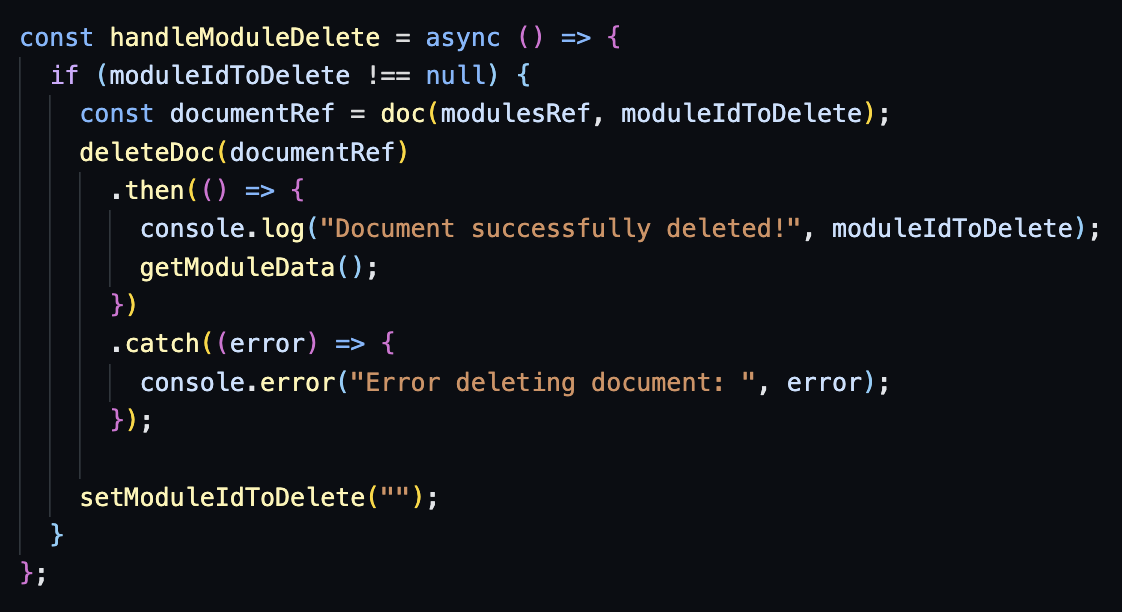


Figure 33: Example of deleting data.

Initially, the process involves verifying the existence of a valid 'moduleIdToDelete'. Should such a moduleId exist, a document reference is established utilizing the 'doc' function available within the Firebase API, incorporating both the 'modulesRef' and the 'moduleIdToDelete'. Subsequently, the 'deleteDoc' function is invoked, with the acquired document reference passed as an argument to commence the deletion operation.

**4.3.3 Accounts and Log In System**

To facilitate the creation and use of user accounts across all three faculty roles—student, teacher, and administrator—a systematic approach leveraging Firestore and an authentication table is employed. The authentication table stores essential user credentials, including email addresses and associated passwords, while uniquely identifying each user through an assigned authentication identifier, termed as authId within this project.

Consequently, a dedicated "users" collection is established to accommodate comprehensive user data, adhering to the prescribed database schema. This collection encompasses vital user attributes such as name, role classification (type), enrolled modules, among others. Notably, the authId serves a dual purpose by being embedded within the "users" collection, functioning as a foreign key to establish a linkage with the authentication table.

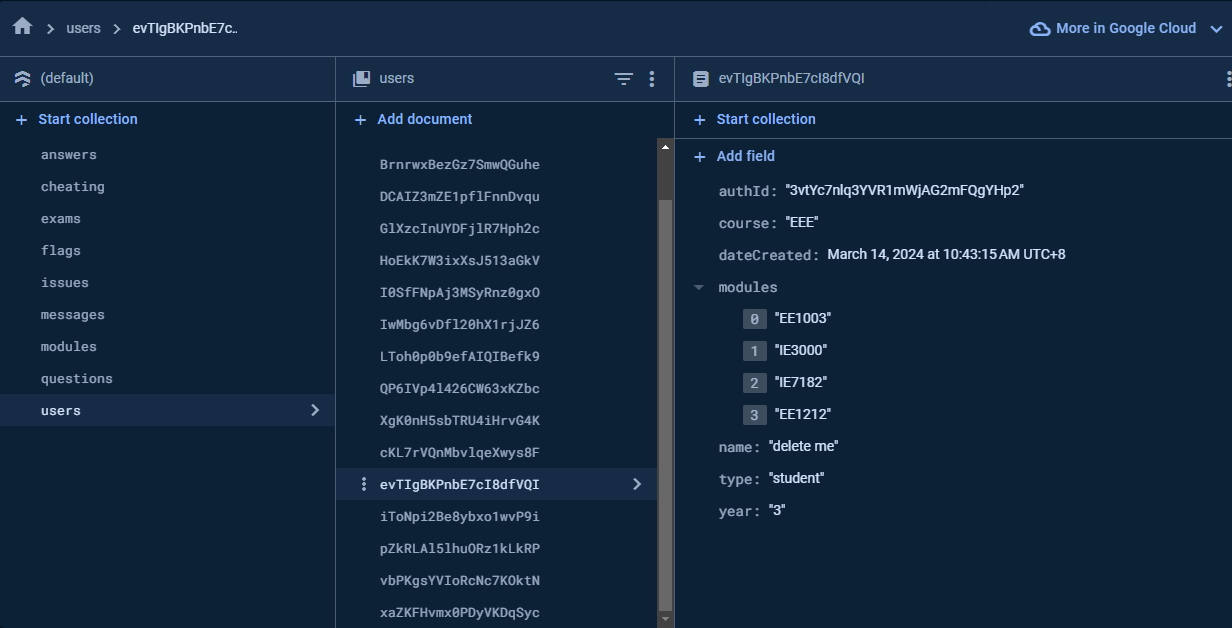


Figure 34: Users collection with example user data.



Figure 35: Same user in authentication table.

Upon initiating login via the designated login page, user authentication is conducted through the integrated function signInWithEmailAndPassword. This function verifies the provided email and password against stored credentials within the authentication table. Upon successful verification, the user gains access to the application, and their authentication identifier “authId” becomes accessible throughout the application via the useAuth() function.

Furthermore, the user is directed to the page corresponding to their role classification, whether student, teacher, or administrator. In the event of an unsuccessful login attempt, either due to non-existence of the user or incorrect input, a modal interface is triggered, notifying the user of the encountered error and guiding them towards appropriate actions.

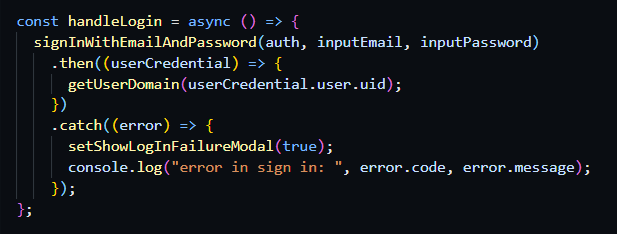


Figure 36: Handle user logging in code.

## 4.4 Front End Implementation

This section is dedicated to exploring the utilization of React for developing the front-end segment of the application, focusing on routing, styling, and the implementation of custom functions to present dynamic data to the user. React facilitates the creation of reusable components that can be employed across various segments of the application.

**4.4.1 Routing**

React router is an open-source package that helps with mapping all the different pages in the app. It works by linking a path such as to an element to rendered, when the link is loaded, the respective element is loaded and shown to the user. Each faculty role has an array of objects that contain all the pages relating to that faculty, allowing for easy implementation of new pages.



Figure 37: Using React Router to map out all possible routes.

**4.4.2 Styled Components**

Styled components represent a method for incorporating styles within React applications. This approach entails the creation of a named component to which styles are applied. Subsequently, this styled component is exported and then imported into the primary page for utilization.



Figure 38: Example of styled components usage.

Styled components employ conventional CSS; however, a significant advantage of this methodology is its capacity to accommodate input properties, or "props," which enable dynamic alterations to the component's styles contingent upon variables defined in the main page. For instance, as illustrated in Figure 39, a modal might be configured to display based on a Boolean property or be rendered conditionally.

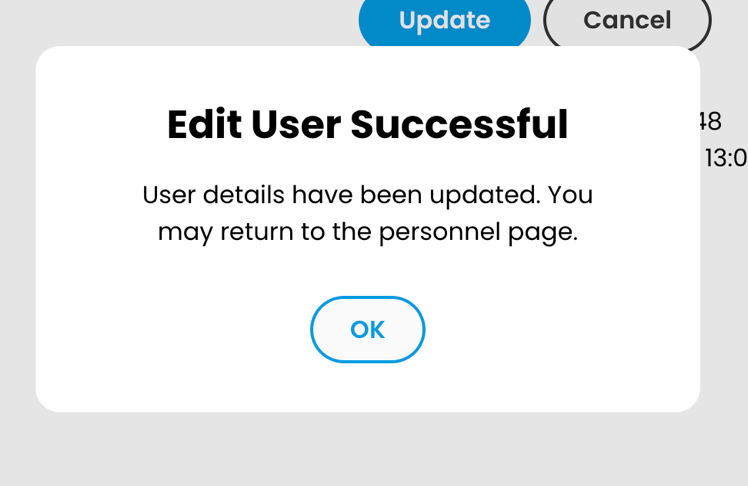


Figure 39: Modal using styled components.

## 4.5 Component Creation

Components are designed for reusability across the application, encompassing elements like navigation bars or buttons. Such components are engineered to be flexible, accepting inputs from parent components. This design philosophy ensures that they can dynamically display data based on the inputs received, thereby enhancing the application's modular architecture and user experience.

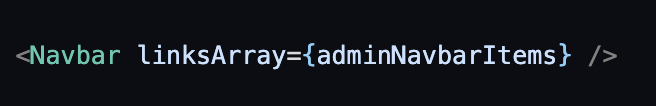


Figure 40: Input array in parent component using Navbar.

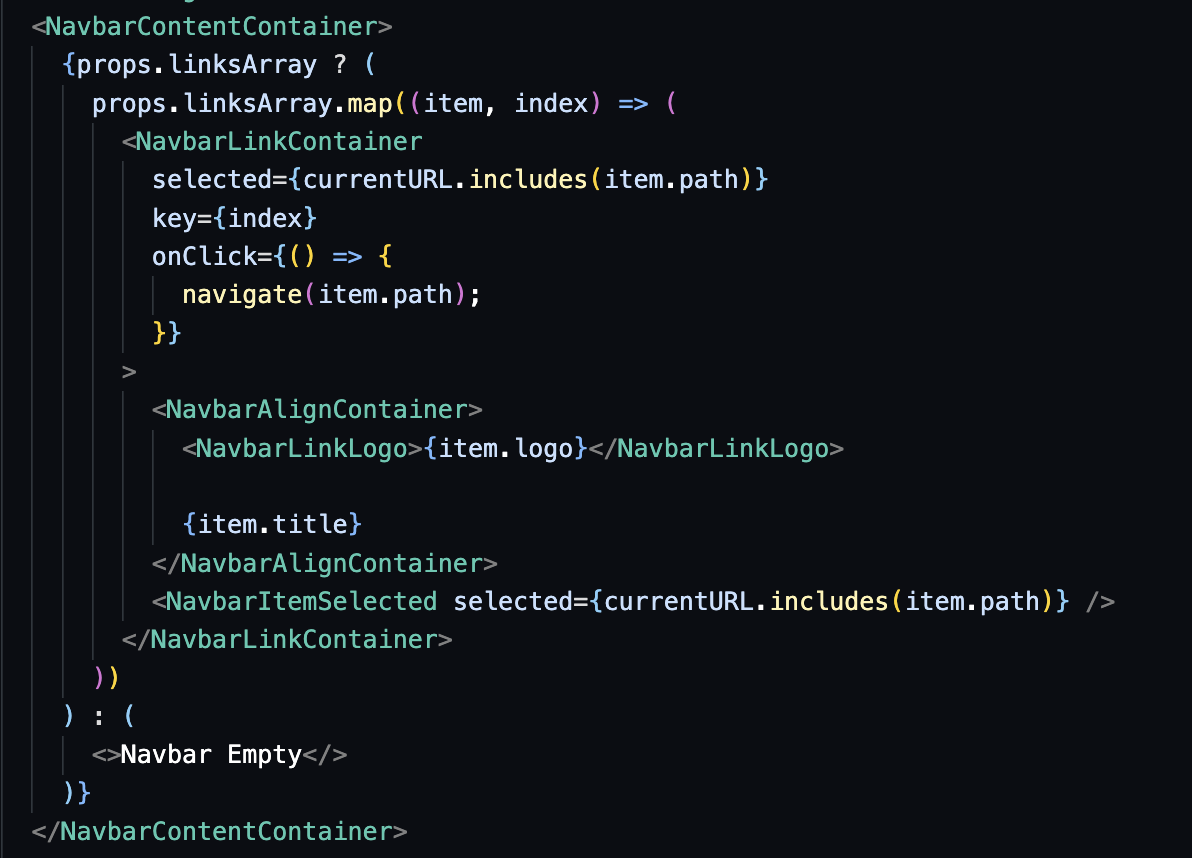


Figure 41: Navigation bar displaying pages in Navbar.js.

Prior to the display of data, a verification process is conducted to ascertain that the input data is not devoid of content, with the provision of an alternative user interface in instances where no data is present. Additionally, the stylings are tailored in accordance with the data input, which are governed by the definitions within the styled component.

# Chapter 5 Results

## 5.1 Exams Control

The administrator has the capability to view all exams that the instructor has scheduled, including upcoming, ongoing, and past exams. This is achieved by retrieving data from the exams collection, sorting it according to the current date, calculating the duration of each exam, and then mapping them out accordingly.A screenshot of a computer exam

Description automatically generated

Figure 42: Exams page

When the user needs to adjust the timing of exams, they can click the “manage” button, which will prompt a modal to appear. This modal allows the user to alter the start and end dates of the exam. The updated duration is also displayed to the user. This functionality is particularly useful in situations involving sudden changes in plans or when an extension of time is required.

A screenshot of a computer

Description automatically generated

Figure 43: Edit exam time modal

## 5.2 Modules

This page displays all the modules available within the application. Each module is identified by a name and is associated with an array of classes. This arrangement enables instructors to tailor quizzes for specific classes as required. Users can modify or remove modules by utilizing the designated "edit" and "delete" icons associated with each module as shown in Figure 44.

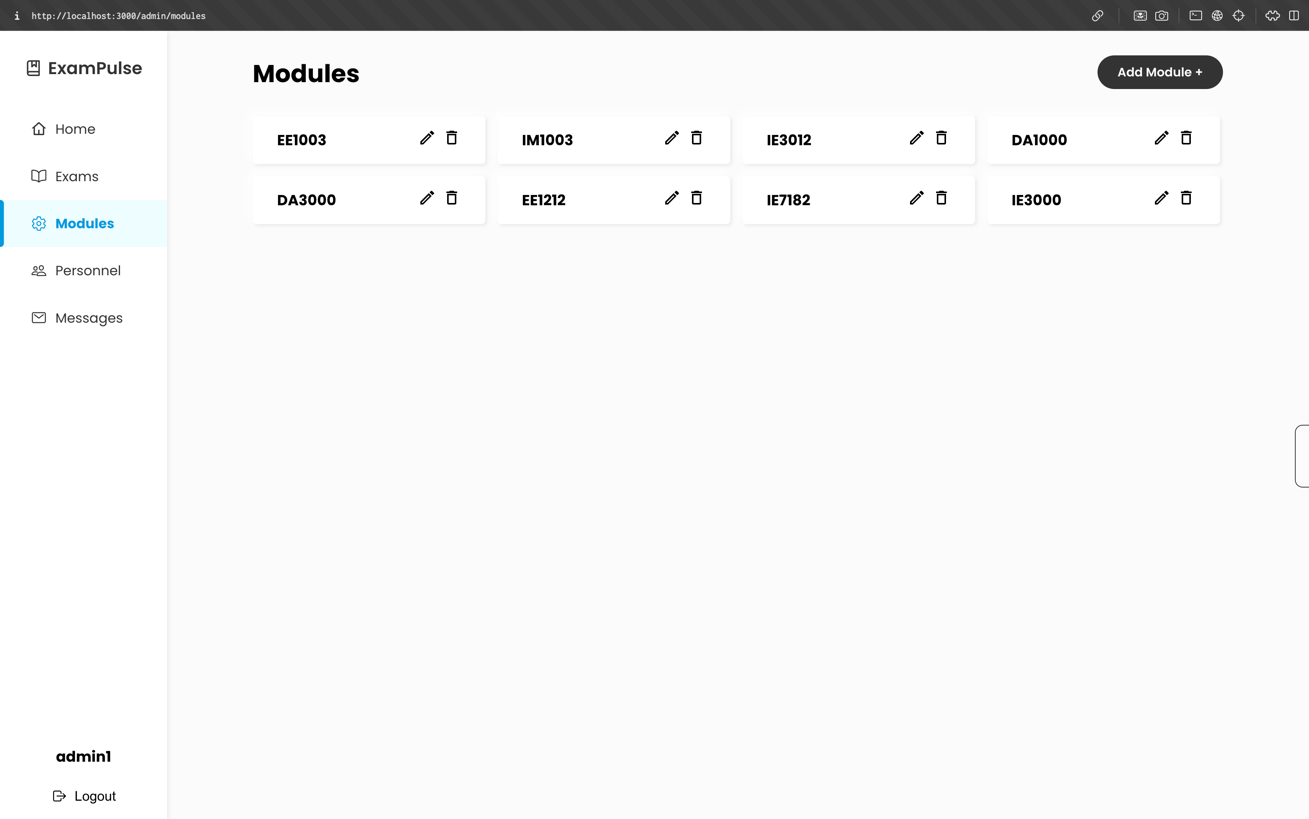


Figure 44: Modules List Page

The user can append new modules to the page by pressing the "add module" button. Upon activation, the user is prompted to input the name of the module and the classes associated with it shown in Figure 45. Upon submission, this addition will prompt an update to the database to reflect the newly incorporated module and a modal will appear to inform the user if the actions was successful or not.

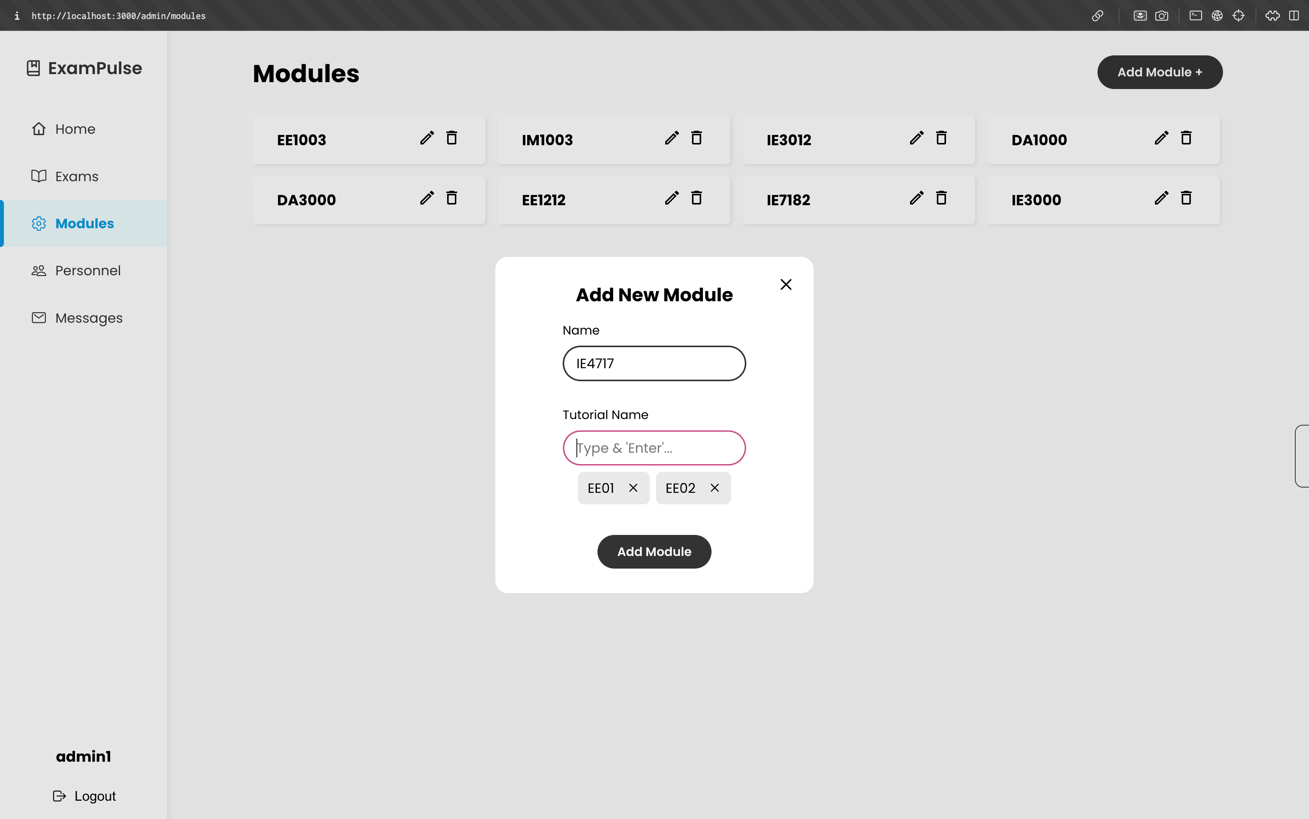


Figure 45: Adding Modules Modal

## 5.3 Personnel

This page provides the user with the functionality to manage the details of other users within the application. From this dashboard, the user possesses the capability to communicate via messaging, modify user details, or delete users, as depicted in Figure 46. Additionally, the page features a filter bar positioned at the top, designed to assist in categorizing users for enhanced visibility, as illustrated in Figure 47. The arrows adjacent to the column titles serve as sorting mechanisms, enabling information organization in ascending or descending order. Messaging a user will redirect the user to the Messaging page with a new conversation with that user.

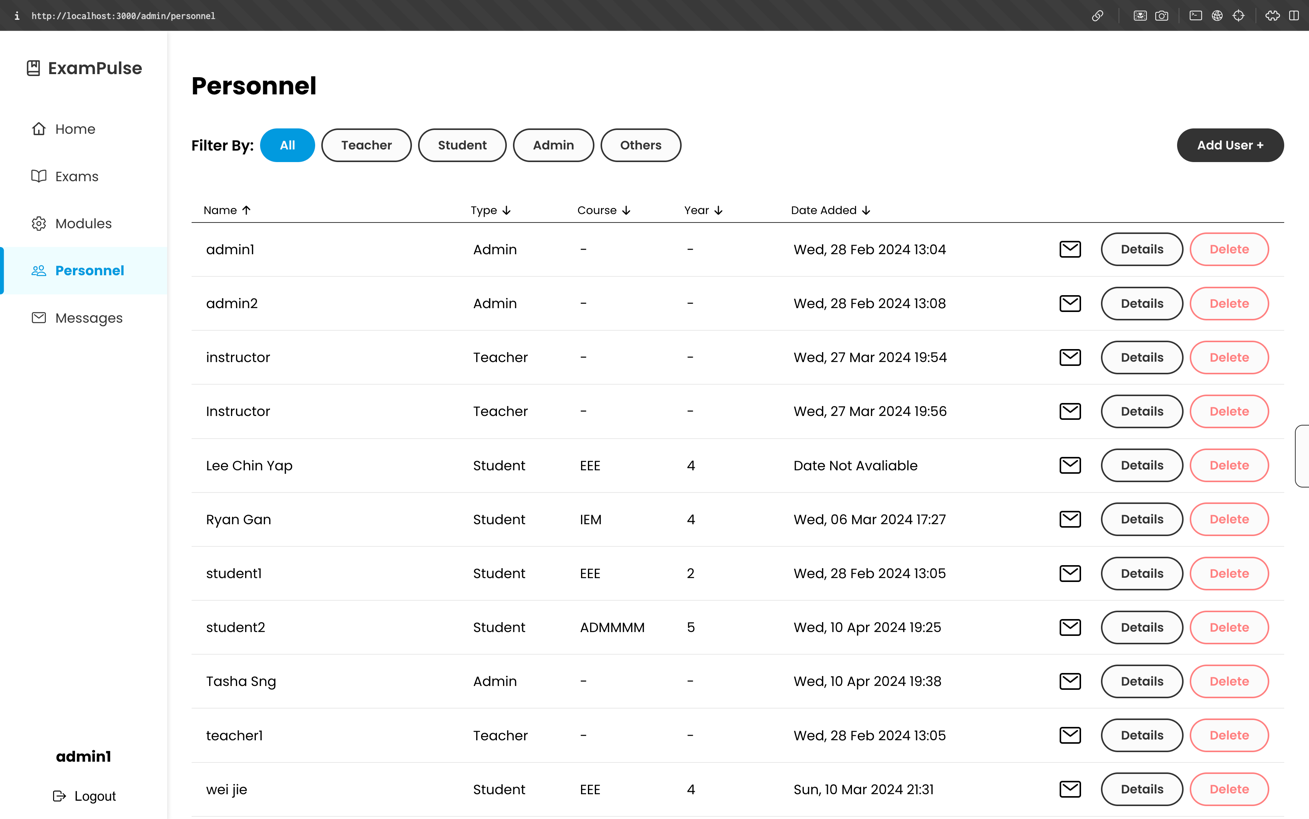


Figure 46: Personnel Page

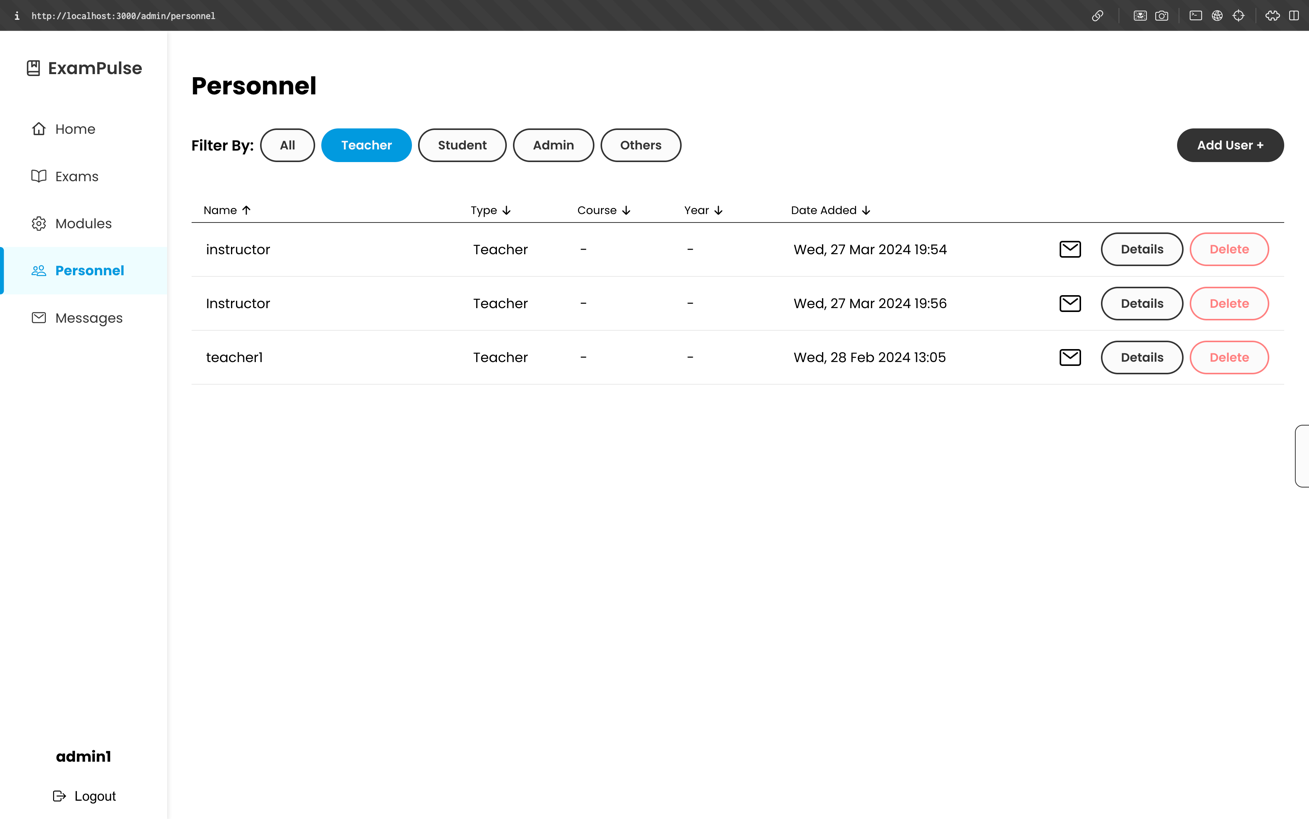


Figure 47: Filtering users in Personnel page.

Since deletion of a user is an important action, a confirmation modal will appear when the user wants to delete another user in case of an accidental click as shown in Figure 48.

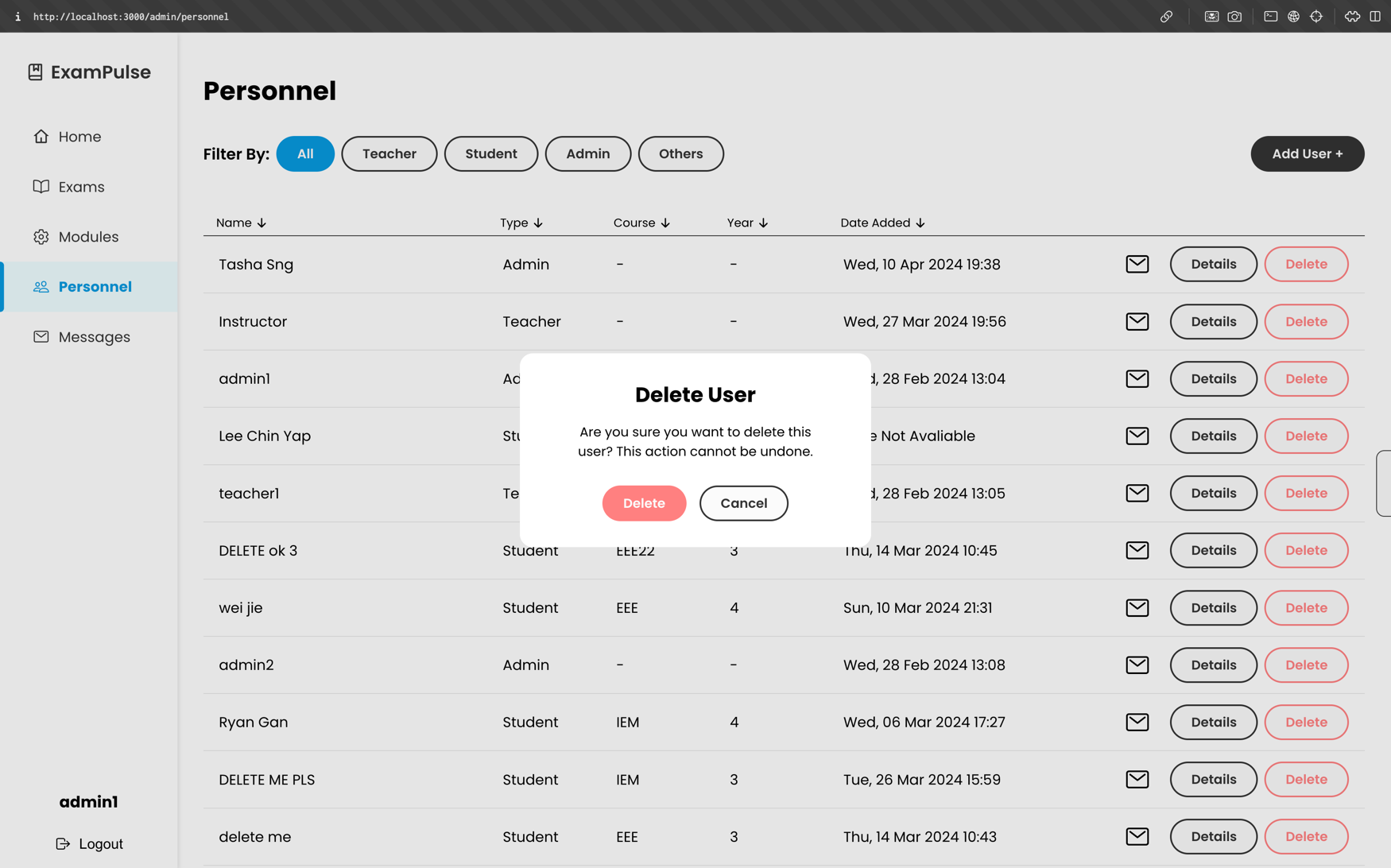


Figure 48: Delete personnel modal.

When adding a new user, the user is presented with the option to select from one of three categories. The selection of a category triggers the display of a distinct screen for the specific user type, as different categories of users necessitate differing input fields as shown in Figure 49. Successful creation of a new user will prompt a modal to show informing the user that the action is successful.



Figure 49: Add new personnel page.

Users can also be edited to update their details such as a change in name or a change in the modules taken as shown in Figure 50.

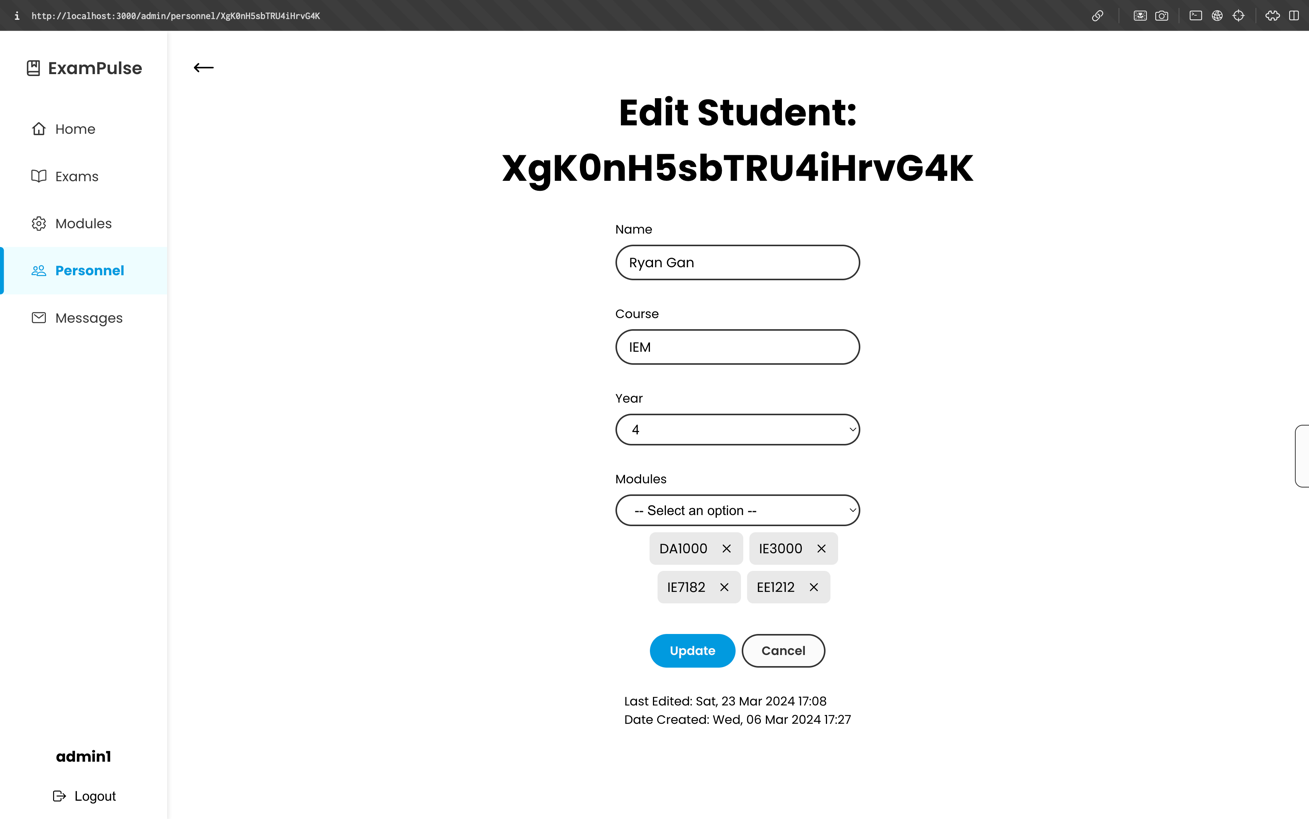


Figure 50: Editing personnel page.

## 5.4 Messages

The messages page serves as a communication channel that facilitates interaction between instructors and administrators for discussing issues or queries. Upon selection of a chat, a function is invoked to retrieve all pertinent messages, searching the "messages" collection in the database for exchanges between the two users, as demonstrated in Figure 51. When a message is sent, an object is added to the database; this object includes the ids for the two users involved and the content of the message.

The sending or reception of a message triggers the same retrieval function to refresh the locally stored array of messages, thus ensuring that the chat interface is updated instantaneously. Additionally, the chat functionality supports the sending of multimedia content such as pictures and videos, enhancing the communicative capabilities of the platform.

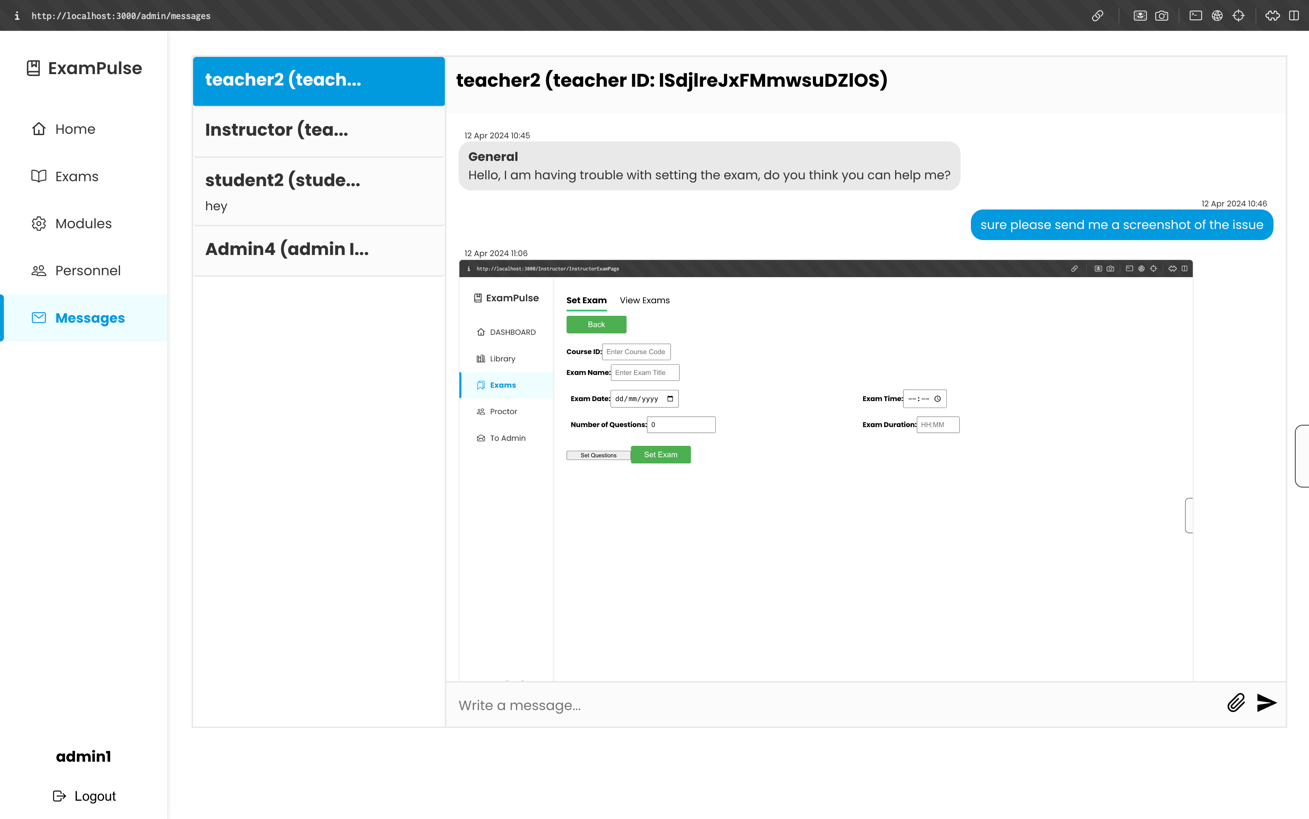


Figure 51: Messages Page

On the Instructor interface, there is a feature that enables instructors to initiate a conversation with an administrator. This is facilitated through a dropdown list, from which instructors can select an administrator of their choice, as depicted in Figure 52. In addition, instructors can select the category of the issue they wish to discuss. Upon submission, if no prior chat exists, this action will initiate a new chat session with the selected administrator. This specific component of the project was developed by my teammate, Cheong Chin Kai who oversees the Instructors portion of the project.

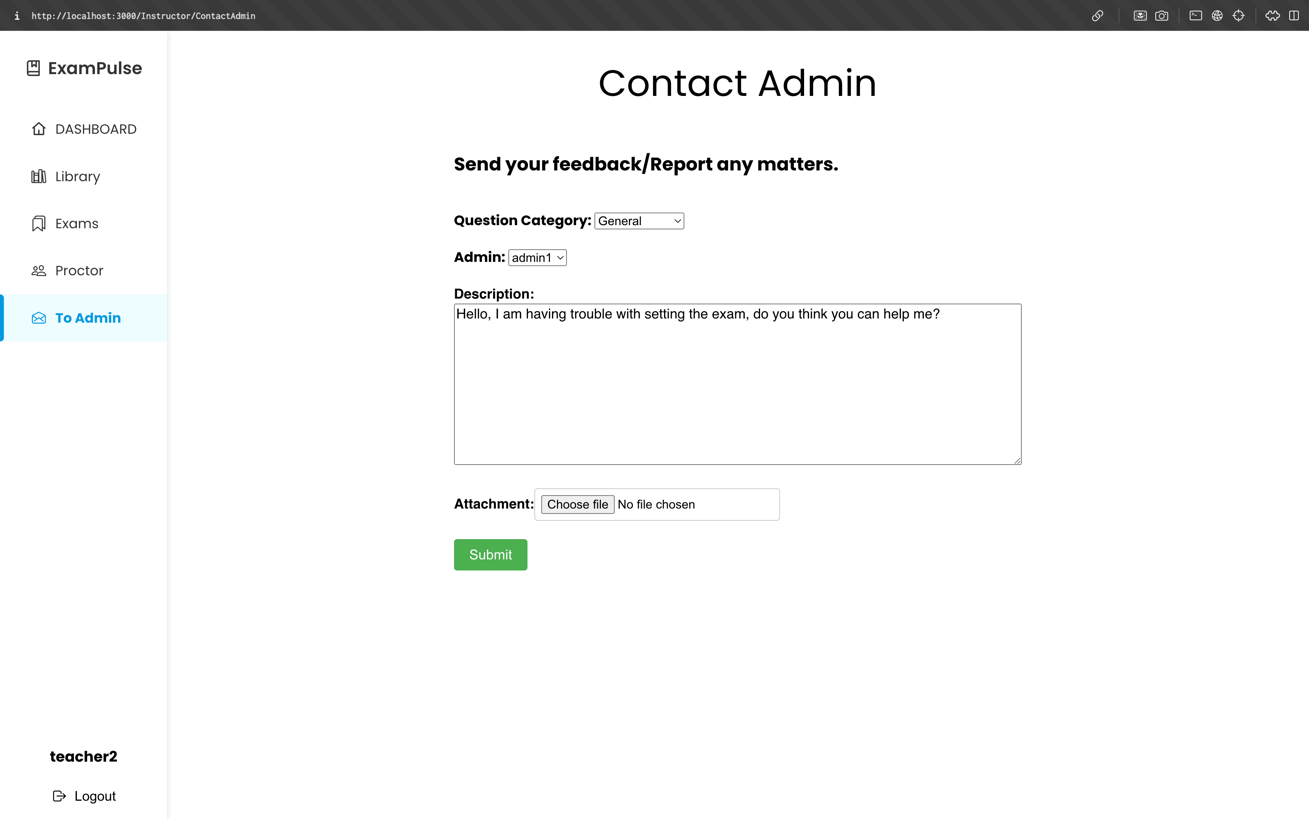


Figure 52: Messaging component in Instructor’s page.

# Chapter 6 Conclusion and Future Work

## 6.1 Conclusion

In conclusion, through the integration of React, Firebase, and various libraries, we have successfully developed ExamPulse, a fully functional and secure online examination platform. This system offers substantial benefits to educational institutions by facilitating the conservation of time and resources.

Implementing Firebase, a database system previously unfamiliar to our team, presented several challenges. However, through dedicated effort, we managed to comprehend the Firebase APIs. The use of the APIs allowed us to perform CRUD functions crucial to any full stack app.

Moreover, the use of React significantly enhanced our project's efficiency and maintainability. By designing reusable components, we established a highly efficient coding structure, which not only streamlined the development process but also ensured greater consistency across the application. This approach has proven to be extremely effective in supporting scalable and dynamic educational technology solutions.

Styled components facilitated the creation of custom styling, which we could seamlessly apply to the components developed within our project. This capability significantly enhanced the aesthetic flexibility and consistency across the user interface, allowing for a more tailored and cohesive visual experience.

Overall, this project has provided valuable insights into the development of digital solutions for educational contexts. It is our hope that the results of our efforts will offer substantial benefits to educational institutions in the future. The experience gained from this project not only enhances our technical expertise but also contributes to our understanding of the practical applications of such technologies in real-world scenarios.

## 6.2 Future Work

**6.2.1 Different modes of examination**

Currently, the examination platform is limited to Multiple Choice Questions (MCQs) due to time constraints in development. However, it would be beneficial to expand the types of available examination modes to include options such as written responses or diagram drawing. These additional modes could greatly enhance the comprehensiveness of the assessments, allowing for a more thorough evaluation of student knowledge and skills across different subjects.

**6.2.2 AI Proctoring**

Currently, during examinations, students are monitored by instructors observing their camera feeds, a process that consumes significant manpower and is challenging for a limited number of instructors to effectively oversee numerous students' behaviour. Consequently, the integration of an AI model to detect eye movements could significantly enhance the efficiency and effectiveness of the app.

**6.2.3 UX Improvements**

To enhance the user experience within the application, the development of a feedback form is recommended. This form should be presented to users upon examination completion to gather their insights and suggestions. The feedback obtained would be instrumental in guiding further refinements and enhancements to the application's functionality and interface.

Additionally, the implementation of a tutorial for new users could significantly improve usability. This tutorial would guide users through the app's features and navigation, ensuring they can effectively utilize all functionalities provided. Such an introduction is particularly beneficial in reducing the initial learning curve and enhancing user engagement from the outset.

# Reflection on Learning Outcome Attainment

During the development of the ExamPulse project, I have obtained valuable insights and a wide range of skills that have helped me improve personally and professionally. This section will provide detailed elaboration regarding the learning outcomes derived from the project.

Firstly, this project presented an opportunity for me to expand my technical skills in coding and software development. Building a full stack application with modern development technologies such as ReactJS has helped me to gain hands on experience in building robust apps. This project also introduced me to Firebase database and how to use it effectively and gain a better understanding of its potential so I can use it future projects,

Next, since this project was group based, where each member took one aspect, we had to work in a team like software teams in the industry. This is something relatively new to me as I had always developed software projects on my own and did not need to communicate with team members or use collaborative tools. However, through the development of this project, I have learnt the importance of delegating work and communicating with my teammates. For example, since we were working on project code at the same time, it was very important to inform one another when we had merged code to the master branch to prevent conflicts. I also learnt to have frequent discussions with my teammates to gather updates and plan on next steps for the project while still maintaining focus on the object of the project.

Moreover, I used this opportunity to practise product management, which includes decision-making processes concerning design elements, styling choices, and the overall trajectory of the app’s development. I was able enhance my proficiency in UI design, through the creation of components and pages from scratch, and the creation of a comprehensive design system to fit the project's requirements. I also improved my time management skills to set and meet deadlines for features to be completed. The acquisition of these project management skills can greatly aid me in future undertakings, empowering me to approach complex projects with assurance and effectiveness.

In conclusion, the development journey of the ExamPulse project has been extremely rewarding, fostering my personal and professional growth. I have expanded my skillset in technical skills, UI design, teamwork and product management, and gained valuable hands-on experience in software development. With these skills, I would be able to tackle complex problems in the future

# Acknowledging/Declaring the Use of GAI

Please refer to NTU's Current Policy & Guidelines on the Use of Generative AI available in NTUlearn home page and the link:

<https://entuedu.sharepoint.com/sites/Student/dept/ctlp/SitePages/Exploring-the-Impact-of-Generative-Artificial-Intelligence-(GAI)-Tools-on-Education.aspx>

1. Complete the following declaration if applicable.
2. Create a Paper Trail to document the input prompt, output obtained, and how you have used it.

I, Yap Qi Long, Marcel, yapq0006[@e.ntu.edu.sg](mailto:______________@e.ntu.edu.sg) (NTU email) honestly and sincerely make the following declaration in relation to the following course submission.

1. Name of course: Information Engineering and Media
2. Course Code: IEM
3. Instructor: A/P Chua Hock Chuan
4. Title of Assignment/Project Submission: FYP Report

In relation to the foregoing, I hereby declare that, fully and properly in accordance with the Assignment/Project Instructions I have (check where appropriate):

|  |  |
| --- | --- |
| 1. Used GAI as permitted to assist in generating key ideas only |  |
| 1. Used GAI as permitted to assist in generating a first text only. |  |
| And/or |  |
| 1. Used GAI to refine syntax and grammar for correct language submission only. |  |
| Or |  |
| 1. As it is not permitted: Not used GAI assistance in any way in the development or generation of this assignment or project. |  |

I also declare that I have:

1. Fully and honestly submitted the digital paper trail required under the assignment/project instructions; and that.
2. Wherever GAI assistance has been employed in the submission in word or paraphrase or inclusion of a significant idea or fact suggested by the GAI assistant, I have acknowledged this by a footnote; and that,
3. Apart from the foregoing notices, the submission is wholly my own work.

|  |  |
| --- | --- |
| Yap Qi Long, Marcel | 12 April 2024 |
| Student Name & Signature | Date |

# References

[1] Awad Ahmed, Fatima Rayan, et al. “Analysis and Challenges of Robust E-Exams Performance under COVID-19.” Results in Physics, vol. 23, Apr. 2021, p. 103987. ScienceDirect, <https://doi.org/10.1016/j.rinp.2021.103987>.

[2] Kasinathan, Vinothini, et al. “ProctorEx: An Automated Online Exam Proctoring System.” Mathematical Statistician and Engineering Applications, vol. 71, no. 3s2, Aug. 2022, pp. 876–89. [www.philstat.org](http://www.philstat.org), <https://www.philstat.org/special_issue/index.php/MSEA/article/view/320>. [Accessed 20 September 2023].

[3] King, Chula G., et al. “Online Exams and Cheating: An Empirical Analysis of Business Students’ Views.” Journal of Educators Online, vol. 6, no. 1, Jan. 2009. ERIC, <https://eric.ed.gov/?id=EJ904058>.

[4] Lee, Kyungmee, and Mik Fanguy. “Online Exam Proctoring Technologies: Educational Innovation or Deterioration?” *British Journal of Educational Technology*, vol. 53, no. 3, May 2022, pp. 475–90. *DOI.org (Crossref)*, <https://doi.org/10.1111/bjet.13182>.

[5] Yağci, Mustafa, and Menderes Ünal. “Designing and Implementing an Adaptive Online Examination System.” *Procedia - Social and Behavioral Sciences*, vol. 116, Feb. 2014, pp. 3079–83. *ScienceDirect*, <https://doi.org/10.1016/j.sbspro.2014.01.711>.

[6] Socrative, “Socrative,” *Socrative*, 2022. <https://www.socrative.com/>

[7] Speedexam, “Online Exam Software | Free Test, Assessment System | Quiz

Maker,” *www.speedexam.net*. <https://www.speedexam.net/>

[8] Quizizz, “About | Quizizz,” *quizizz.com*. https://quizizz.com/home/about?lng=en

[9] “Guide to Software Project Management | Smartsheet,” *www.smartsheet.com*. https://www.smartsheet.com/content/software-project-management

[10] M. Fowler and J. Highsmith, “The Agile Manifesto,” 2001. Available: https://www.nitrix-reloaded.com/wp-content/uploads/2010/05/The\_Agile\_Manifesto\_SDMagazine.pdf

[11]J. TERRY, “Benefits of Agile Development | Planview LeanKit,” *Planview*, 2023. https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/benefits-of-agile-development/

[12] “Jira | Definition and Overview,” *www.productplan.com*. https://www.productplan.com/glossary/jira/

[13] A. Gerber and C. Craig, “Introducing Git,” *Apress eBooks*, pp. 145–187, Jan. 2015, doi: https://doi.org/10.1007/978-1-4302-6602-0\_7.

[14] “Importance of version control in software development | MoldStud.com,” *moldstud.com*, Mar. 11, 2024. https://moldstud.com/articles/p-importance-of-version-control-in-software-development (accessed Apr. 12, 2024).

[15] GitHub, “About Repositories,” *GitHub Docs*. https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories

[16] GitHub, “Committing and reviewing changes to your project in GitHub Desktop,” *GitHub Docs*. https://docs.github.com/en/desktop/making-changes-in-a-branch/committing-and-reviewing-changes-to-your-project-in-github-desktop

[17] Atlassian, “Git Merge | Atlassian Git Tutorial,” *Atlassian*. https://www.atlassian.com/git/tutorials/using-branches/git-merge#:~:text=Merging%20is%20Git

[18] “Git - First-Time Git Setup,” *git-scm.com*. https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

[19] J. Losch, “How git works,” *Johannes Losch*, Jul. 26, 2020. https://blog.jlosch.de/how-git-works (accessed Apr. 12, 2024).

[20] H. Clark, “6 Stages Of The Software Development Life Cycle (SDLC),” *The Product Manager*, May 20, 2022. https://theproductmanager.com/topics/software-development-life-cycle/

[21] “Software Development Planning: Process, Importance, Tools,” *www.knowledgehut.com*, Sep. 05, 2023. https://www.knowledgehut.com/blog/web-development/software-development-planning

[22] “ACID Databases – Atomicity, Consistency, Isolation & Durability Explained,” *freeCodeCamp.org*, Jan. 17, 2024. https://www.freecodecamp.org/news/acid-databases-explained/#:~:text=ACID%20stands%20for%20Atomicity%2C%20Consistency (accessed Apr. 12, 2024).

[23] AWS, “Amazon Relational Database Service (RDS) – AWS,” *Amazon Web Services, Inc.*, 2019. https://aws.amazon.com/rds/

[24] L. Carrero, “10 popular database management systems (DBMS) [List],” *Stackscale*, Mar. 17, 2022. https://www.stackscale.com/blog/popular-database-management-systems/

[25] DBeaver, “About | DBeaver Community,” *DBeaver*. https://dbeaver.io/about/

[26] pgAdmin, “pgAdmin - PostgreSQL Tools,” *www.pgadmin.org*. https://www.pgadmin.org/

[27] S. Preibisch, *API Development*. Berkeley, CA: Apress, 2018. doi: https://doi.org/10.1007/978-1-4842-4140-0.

[28] S. Rasul and Q. Bukhsh, “A study of factors affecting students’ performance in examination at university level,” *Procedia - Social and Behavioral Sciences*, vol. 15, pp. 2042–2047, 2011, doi: https://doi.org/10.1016/j.sbspro.2011.04.050.

[29] E. Feo, “The Emotional Connection in UX: Bridging Users and Software 🙂,” *Medium*, Dec. 02, 2023. https://bootcamp.uxdesign.cc/the-emotional-connection-in-ux-bridging-users-and-software-1da3f9154666 (accessed Apr. 12, 2024).

[30] C. Lengen, “The effects of colours, shapes and boundaries of landscapes on perception, emotion and mentalising processes promoting health and well-being,” *Health & Place*, vol. 35, pp. 166–177, Sep. 2015, doi: https://doi.org/10.1016/j.healthplace.2015.05.016.

[31] J. Choma, Luciana, and D. Beraldo, “Communication of Design Decisions and Usability Issues: A Protocol Based on Personas and Nielsen’s Heuristics,” *Lecture Notes in Computer Science*, pp. 163–174, Jan. 2015, doi: https://doi.org/10.1007/978-3-319-20901-2\_15.

[32] J. Yablonski, *Laws of UX*. “O’Reilly Media, Inc.,” 2024. Accessed: Apr. 12, 2024. [Online]. Available: https://books.google.com.sg/books?hl=en&lr=&id=3RHwEAAAQBAJ&oi=fnd&pg=PT17&dq=Jakob%E2%80%99s+Law+design&ots=TEdkddl1Ax&sig=taqe8aSF-ynxzVZYoWElX6yQOCc&redir\_esc=y#v=onepage&q=Jakob%E2%80%99s%20Law%20design&f=false

[33] J. Nielsen, “10 Heuristics for User Interface Design,” *Nielsen Norman Group*, Apr. 24, 1994. https://www.nngroup.com/articles/ten-usability-heuristics/

[34] B. Kopf, “The Power of Figma as a Design Tool,” *Toptal Design Blog*, 2018. https://www.toptal.com/designers/ui/figma-design-tool

[35] E. Castro, *HTML for the World Wide Web*. Peachpit Press, 2003. Accessed: Apr. 12, 2024. [Online]. Available: https://books.google.com.sg/books?hl=en&lr=&id=Dq9L3KkDOFIC&oi=fnd&pg=PA13&dq=what+is+html&ots=3CkRKxCYEP&sig=8q0ALtpObrr5SJSNRrMC4RqkgXM&redir\_esc=y#v=onepage&q=what%20is%20html&f=false

[36] W3schools, “CSS Introduction,” *W3schools.com*, 2019. https://www.w3schools.com/Css/css\_intro.asp

[37] A. Ranjan, A. Sinha, and R. Battewad, *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript*. BPB Publications, 2020. Accessed: Apr. 12, 2024. [Online]. Available: https://books.google.com.sg/books?hl=en&lr=&id=b2bdDwAAQBAJ&oi=fnd&pg=PT24&dq=what+is+javascript+used+for+in+web+development&ots=6gcB1DvrI0&sig=-k4gGwQj-iqzCC2Gtm2A6LamTYE&redir\_esc=y#v=onepage&q=what%20is%20javascript%20used%20for%20in%20web%20development&f=false

[38] S. Aggarwal, “Modern Web-Development using ReactJS,” *International Journal of Recent Research Aspects*, vol. 5, pp. 133–137, 2018, Available: http://ijrra.net/Vol5issue1/IJRRA-05-01-27.pdf

[39] “Why Should You Use a Router in React.js? | Simplilearn,” *Simplilearn.com*, Sep. 27, 2022. https://www.simplilearn.com/tutorials/reactjs-tutorial/routing-in-reactjs#:~:text=React%20Router%20is%20a%20JavaScript