



THE PROJECT REPORT IS PREPARED  
FOR  
FACULTY OF COMPUTING AND INFORMATICS  
FOR  
FOUNDATION IN INFORMATION TECHNOLOGY  
TRIMESTER 3, 2020/2021

**PSP0201**

**MINI IT PROJECT**

**MCQ QUIZ MAKER**

**MR. NEOH KEE LIN**

CHEAH VEE VIEN	1201100314
LINGKESWARI PONNUSAMY	1201101292
SRIHARI NAIDU A/L VENKATASH	1201100182
SUKANYA A/P THINAKARAN	1201101315

15 / 6 / 2021

# **ABSTRACT**

This project focuses on solving disorganised quizzes by creating a paperless quiz system that manages quizzes for users which includes both teachers and students and tracks quiz results using Python version 3.9.2 with Tkinter user library. The scope of the system can be defined by looking at the type of user account interacting with it, each with different functional authorities over quizzes and users. Firstly, the teacher has the largest functional authority to manage quizzes for students and release questions to retrieve insightful statistical information. The second type of user which are students are able to perform quizzes and view their results.

# TABLE OF CONTENTS

<b>TITLE</b>	<b>PAGE</b>
<b>CHAPTER 1: INTRODUCTION</b> 1.1. Problem Statement 1.2. Project Objective	<u>4 - 5</u> <u>5</u>
<b>CHAPTER 2: BACKGROUND STUDY</b> 2.1. Case Study 1- Quizizz app 2.2. Case Study 2 - Kahoot app	<u>6 - 11</u> <u>12 - 15</u>
<b>CHAPTER 3 :DESIGN</b> 3. Functional Design 3.1. Storyboard 3.2 Flow Chart 3.3 Data Dictionary 3.4 User Manual	<u>16 - 17</u> <u>18 - 22</u> <u>23</u> <u>24 - 63</u>
<b>CHAPTER 4 : IMPLEMENTATION</b> 4.1. System overview 4.2. Code Results	<u>64</u> <u>65 - 70</u>
<b>CHAPTER 5 : CONCLUSION</b>	<u>71</u>
<b>CHAPTER 6 : REFERENCES</b>	<u>72</u>

# CHAPTER 1: INTRODUCTION

## 1.1 PROBLEM STATEMENT

---

This section defines the problems and situations faced by using existing paper-based quizzes.

### a) High Cost

Paper-based tests require a great deal of printing work and papers. Each time a test is given, it needs to go through the paper-and-ink printing measure. The printing process not only involves monetary cost but it is also time-consuming. For instance, assuming there are 1000 students taking the test, the printing job must use up at least 1000 pieces of paper. In addition, a mistake or correction in the quiz paper will result in a massive reprinting work. After a quiz session, the answered quiz papers are gathered and cannot be utilized once more. Accordingly, even with a similar arrangement of inquiries, another quiz session will in any case require printing another new set of quiz papers.

### b) Cumbersome marking process

After a quiz session, the instructors would have to painstakingly hand-mark each collected quiz paper, calculate the scores and enter it into a list for students to view their results. This is time-consuming and reduces the efficiency of the marking process. It may also result in human error such as wrong calculation of scores.

### c) Difficult to manage

The quantity of quizzes generated increases over time and deliberate stockpiling of these papers will be troublesome and disorganised. It is likewise hard to search and trace back previous quizzes for reference.

**d) Inconvenience and Low Security**

Paper-based quizzes can be easily exposed, leaked and compromised. A huge pile of paper-based quizzes are difficult to store and check on constantly. Therefore, they are usually kept in lockers which might not be a safe option if someone steals the key. Various problems may also arise when it comes to paper-based quizzes such as inadequate sets of quizzes printed, lost pages within quizzes and lost keys to locker.

## **1.2 PROJECT OBJECTIVE**

---

The objective of fostering this quiz management system is to establish a paperless environment aimed to overcome the limitations of the paper based quiz system.

- a) To provide a content-rich paperless quiz system
- b) To obtain instantaneous quiz results and speed up the assessment of quizzes.
- c) To provide useful statistical information on user details and scores.
- d) To provide a systematic, efficient and user-friendly environment for creating and managing quizzes.

# **CHAPTER 2: BACKGROUND STUDY**

A quiz is a game or contest where the participant attempts to answer the given questions effectively. It is likewise a short casual evaluation utilized in education and similar fields to gauge development in knowledge, abilities, and skills. Quizzes are normally created by a party to collect feedback and perform evaluations on an individual or a group of individuals. This chapter discusses the advantages and disadvantages of the computer-based quiz which we made. Other than that, this chapter also includes an analysis on two different quiz management systems found on the Internet.

## **2.1 CASE STUDY 1 : QUIZIZZ APP**

---

Quizizz is an online-based system which allows teachers to conduct student-paced formative assessments in a fun and engaging way for students of all ages. The system is designed in such a way where the questions will appear on each of the student's screens, so that they can answer questions at their own pace, and review their answers at the end of the quiz.

Apart from that, this quiz can be played by students using any kind of device with a browser, including personal computers, laptops, tablets, or smartphones. The system also authorizes teachers to pluck questions from any other quizzes, add images from the internet with no trouble and auto-save their progress. Teachers from around the world create thousands of great questions on Quizizz every day. This community effort helps generate great content that can be used by everyone.

Generally, there are multiple settings that teachers can control before starting a quiz. Teachers can see the settings menu after selecting their preferred game type - Live Quiz or Assign Homework on the quiz page. Quizizz also gives teachers detailed class-level and student-level insights for every quiz that is conducted. Teachers can also download the reports as an Excel spreadsheet.

The screenshot shows the Quizizz platform interface. On the left, there's a sidebar with user information (Mr. Shamill, Plan: Super), navigation links (Create, Explore, My library, Reports, Classes, Settings, More), and a Help button. The main area displays a quiz titled "A Modern Cinderella". It includes a thumbnail image, statistics (67% average accuracy, 1680 plays), and details (5th grade, English). Below this are two tabs: "INSTRUCTOR-LED SESSION" (selected) with "Start a live quiz" and "ASYNCHRONOUS LEARNING" with "Assign homework". The quiz itself has 15 questions. Question 1 asks about the perspective of the modern version, with options: Mystery Girl, Prince, Queen, and Audience and Judges. Question 2 asks what does NOT make it modern, with options: A TV, A horse and carriage, and A castle. There are buttons for Show Answers, Preview, and a timer set at 120 seconds.

FIGURE 2.1.1: QUIZZIZZ GAME TYPE PAGE (LIVE OR HOMEWORK)

The screenshot shows the "General settings" menu. It contains four sections: 1) "Participant attempts" with a dropdown set to "Unlimited". 2) "Name factory" with a toggle switch turned off. 3) "Show answers during activity" with a dropdown set to "On". 4) "Show answers after activity" with a dropdown set to "On".

Figure 2.1.2: Quizziz general setting menu

Quizizz: Movie Trivia										
Quiz started on: Thu 09, Feb 06:59 PM Total Attendance: 6 Average Score: 2833										
Questions	Class Level			Player Level						
	# Correct	# Incorrect	# Unattempted	James	Ange	Myrie	Full Will			
Which movie series always has the quote, "I have a bad feeling about this"	3	2	1	star wars	star wars	star wars	Indiana Jones			
Which Disney Pixar movie has the quote, " I'm happier than a tornado in a trailer park!"	4	1	1	Cars	Cars	Cars	Cars			
In which 80's movie does Tom Cruise say, "I have the need, the need for speed!"	4	1	1	Top Gun	Top Gun	Days of Thunder	Top Gun			
Who said, "They may take our lives, but they'll never take our freedom!!"	5	0	1	Mel Gibson in Braveheart						
Which Rocky movie has Mr. T in it?	3	2	1	Rocky 3	Rocky 3	Rocky 3	Rocky 4			
<b>Total</b>	<b>19</b>	<b>6</b>	<b>5</b>	4740	4340	3540	2590			
<b>Accuracy</b>	<b>63%</b>			100%	100%	80%	60%			
<a href="#">Player level data in next Sheet!!</a>										
Time is represented in GMT										

**Figure 2.1.3: Quizziz Results As Excel Spreadsheet**

## STRENGTHS

### 1. Engagement of students in class

Students complete the homework assigned at their own pace and therefore it helps them to remain highly engaged throughout the quiz. Although there is a problem where the students will finish their homeworks at different timings, but Quizizz has a couple of features that may solve this problem. Teachers can easily see the progress of all students so the teacher will know who has finished and can easily direct them to the next activity.

### 2. Homework

Quizizz can be assigned for homework. All the students will only need the game code and they can complete their work at anytime and anywhere. Homework quizzes can also be assigned a deadline time to automatically end, preventing any further attempts.

### 3. Blog and teacher resource section

Quizizz also has a Blog Section that highlights new features or techniques for using the platform. Besides, Quizizz has a Teacher Resource Section that includes a 1-page get-started guide, training presentation, and help center.

#### **4. Questions and Memes**

You can also add existing questions from other's Quizizz to your own. Fun Memes can also be added to questions once they are answered. These can be customized to fit your classroom community, school environment, or classroom content and provide positive feedback for students.

## **WEAKNESSES**

#### **1. Assignment creation**

Quizzes cannot be created asynchronously. Every student needs to be available at the same time. For example students cannot retake the quiz if they did it poorly. The quizzes also usually have a specific deadline, which makes it less flexible.

#### **2. Question visibility**

Students need to switch between the teacher's screen and theirs during an activity to view the questions which is time-consuming because they have to keep switching tabs to view the questions.

#### **3. Join code for the game**

Students need to type in the code (example: 02273952) into the space provided to join the game. Sometimes these codes are invalid and create some technical errors which takes time for the students to actually join the game.

## **SOLUTION TO THESE PROBLEMS**

Although there are many problems in Quizizz, nevertheless all these technical problems have been solved by Quizizz themselves in March 2020. They have been working with teachers to expand from a multiple-choice quiz game into a learning platform to meet more of our needs.

Firstly, teachers can now create as many assignments as they want. They can use Teleport to quickly add in new questions . Students can also retake the quiz by default and teachers can see how students progress in their reports. Quizizz has also added the ability to have no deadline, extended deadlines, and to reopen expired assignments with their new Super plan.

Secondly, Quizizz has enabled teachers to host a live quiz outside of the classroom. So from now onwards, students can always see questions on their own devices. Therefore students don't have to switch between the teacher's screen and their screen during the process. This allows students to engage directly with the question and answer options which helps reinforce the content and makes it easier to read, and allows students to enable device-level read-aloud if they need it. Finally, if a student has limited WiFi, they'll have enough time to view the question and won't be at a disadvantage due to a disconnection (self-pacing helps with this as well).

Thirdly, students can now use a join link to skip the code entry step which helps in saving time and sanity.

The screenshot shows the Quiz Editor interface. At the top, there are buttons for 'Quiz Editor', '+ New question', and 'Teleport'. The 'Teleport' button is highlighted with a red box. Below it, a question card is displayed for 'Question 1'. The question text is: 'At the beginning of the story, why does the author make you guess her pet?'. An image of a dog is shown next to the question. Below the question are three answer choices:

- to make the story fun to read
- because she does not know the answer
- to make the reader take a hard test

At the bottom of the card are buttons for '30 Seconds' and 'Tag topics'.

Figure 2.1.4 : Teleport Function

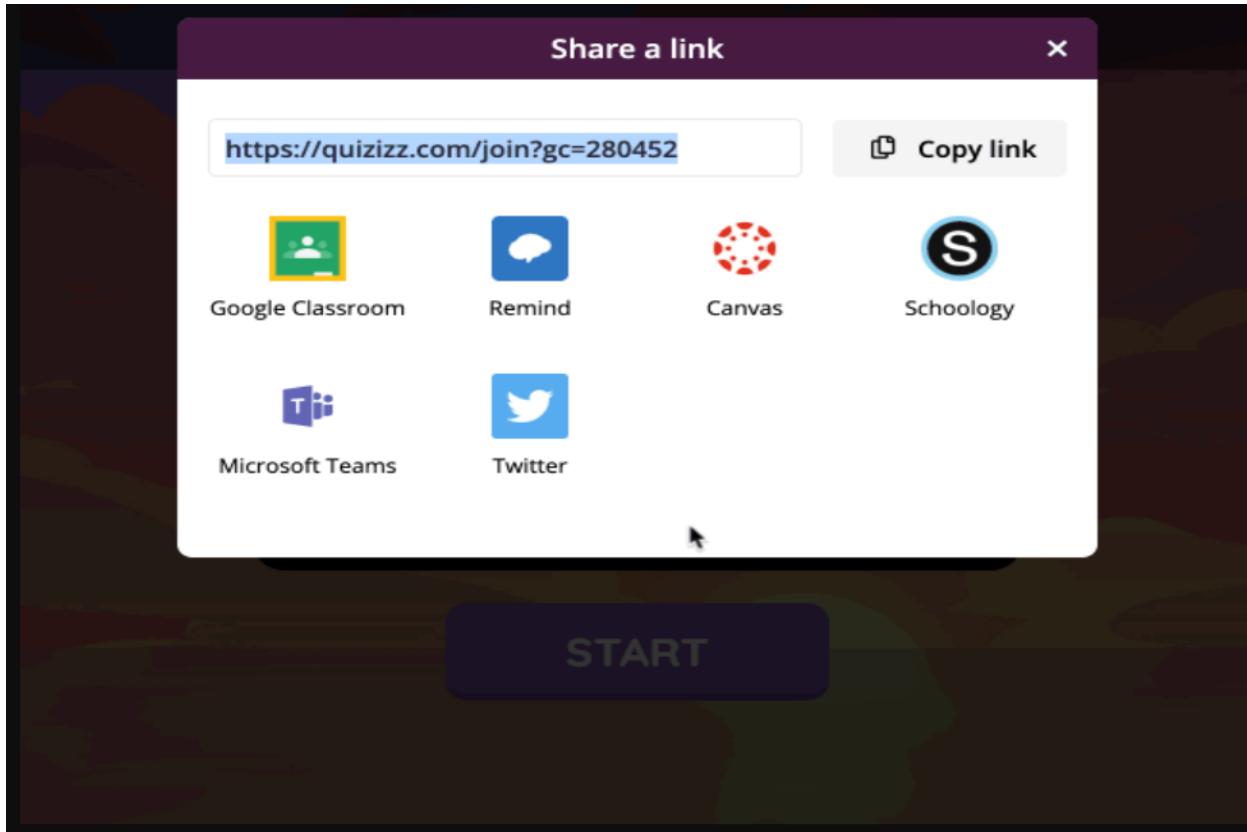


Figure 2.1.5 : Share Link For The Quiz Game

## 2.2 CASE STUDY 2 : KAHOOT APP

---

Kahoot! is a game-based learning system that is mainly used for educational purposes in schools and many other educational institutes. Kahoot! is also an interactive learning application that requires students' active responses. All the students will be connected to the game using a generated game PIN which will be shown on the teacher's screen. The questions created by the teacher will be presented on a device while students answer them using their own devices such as mobile phones, computers or laptops. Points will be given to correct and quick responses and students' names will be displayed at the leaderboard after each question.

As for the users, there are four options at the login page which are teacher, student, workplace and social where each contains its own set of features to the respective users. As for the questions setup, users need to add questions and indicate the correct answer for the answer choices they've added. Users are able to create different types of questions and enhance the questions with pre-timer, images, animation and sounds. The quiz is a basic type of question where it requires at least two options, one of which must be marked as the correct or wrong answer. There is an option for users to set-up the questions as single-select or multi-select. Kahoot! can be accessed through Kahoot! mobile application and web browser.

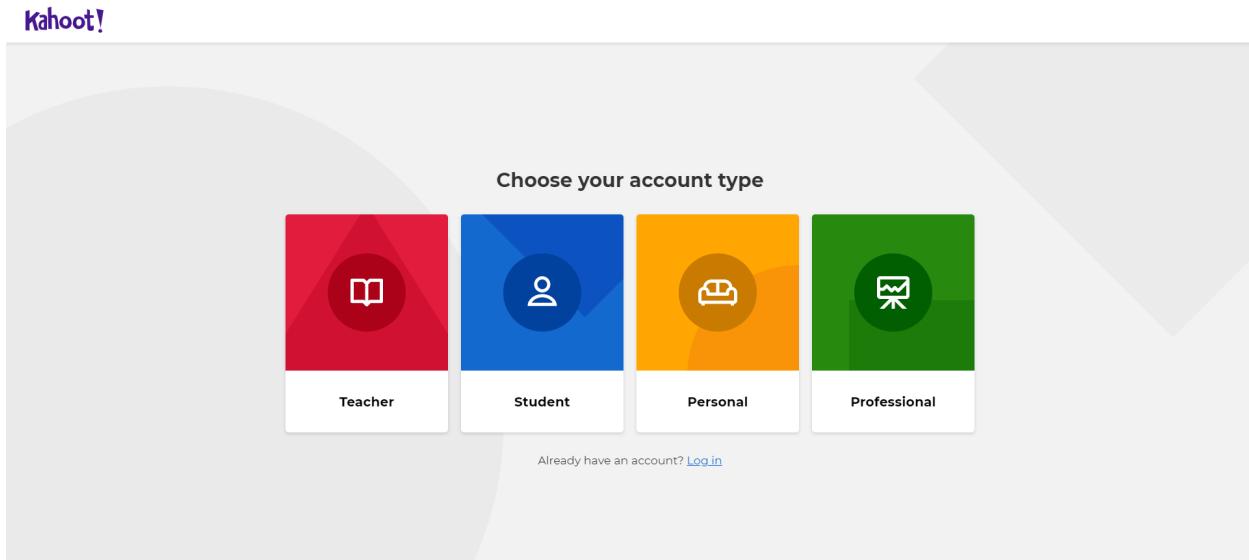
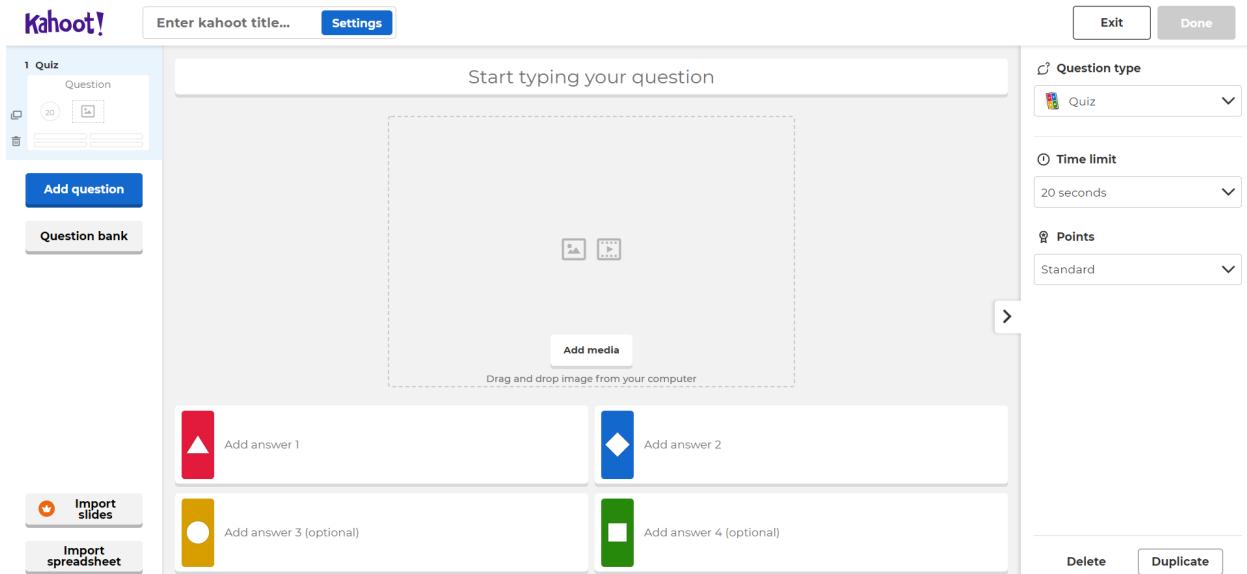


Figure 2.2.1 : Kahoot! Login Page



**Figure 2.2.2 : Kahoot! Create Question**

## STRENGTHS

### 1. Active engagement of the students

Kahoot! offers active participation from the students' side as it provides a unique type of quizzes. It also supports a multi-players system which creates interaction between students and teachers.

### 2. Integrating multimedia elements

The vibrant colours and design of Kahoot! provides excellent visuals for the students. The lively platform with music and interactive quizzes with corresponding images and animations creates a positive environment for the players more engaged to the game as well.

### 3. Usage Flexibility

Teachers who are less experienced with technology can also get adapted to the system as its setups are simple and straightforward. Kahoot! has the ability to create quizzes within minutes and it also has a feature to import questions from users' devices or quizzes created by other users. Teachers can also save and import the quiz they created in

spreadsheet form. Teachers are able to track all the quizzes, answers, scores and participation details and review statistics in reports. Teachers are also able to analyse students' performance in formative assessments and students' reports can be accessed and downloaded in spreadsheet form.

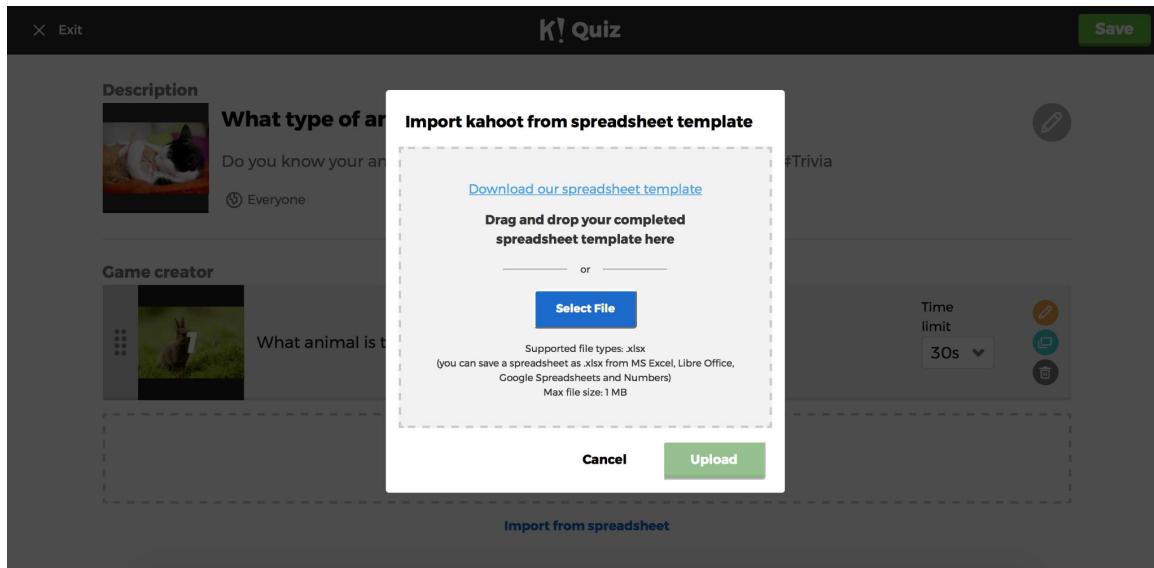


Figure 2.2.3 : Import Quiz From Spreadsheet

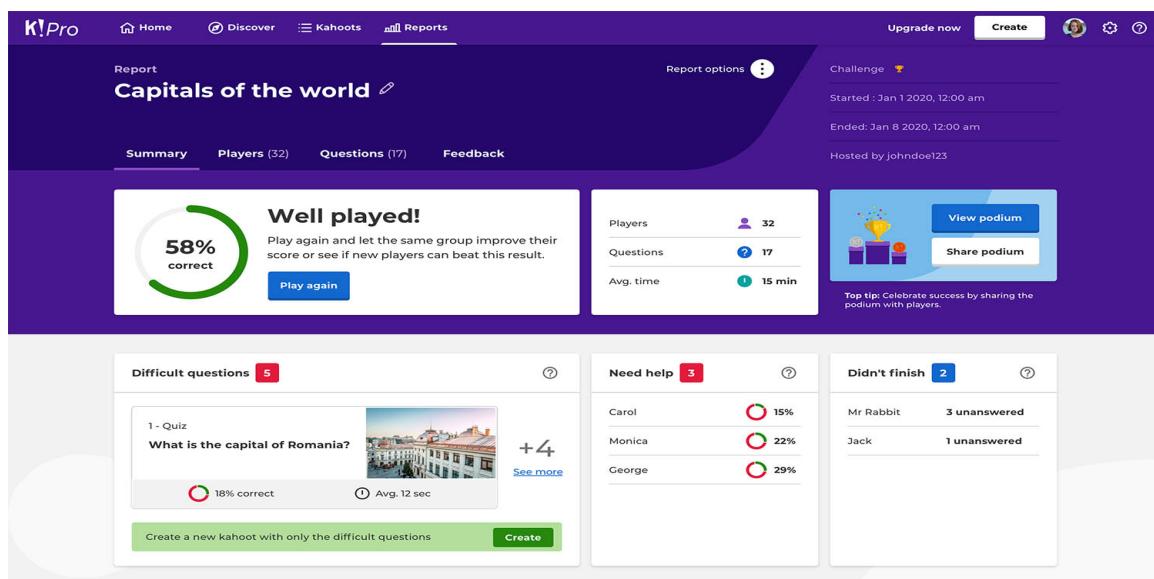


Figure 2.2.4 : Report Summary

## **WEAKNESSES**

### **1. Unfair rankings**

Being evaluated based on quick responses might not appeal to all the students, especially to the impaired and the ones without proper access to technology. Teachers need to set up the timer with the given range of time limit which is between 5 seconds and 4 minutes only. Being aware of the timer, students may answer the questions without any consideration which leaves them with wrong answers.

- A similar problem exists in an app called “Timed Quiz” where students are assessed based on their quickest responses.
- This problem is solved in the “Quizizz” as it provides a feature to disable the timer that is included for the questions in the quiz. Students can take their own time to do the quizzes and students will be evaluated based on their answers not on their possible quick responses.

### **2. Access to technology**

Multiple-players connected to the same game would require stronger internet connection. Students with poor internet connection might be left out and lose the quiz track. Also, students must have their own gadgets like mobile phones or laptops to answer the questions else they are unable to participate in the quiz.

- The similar problem exists in “Edmodo” where students require good internet connection and availability of gadgets to answer the questions. Students would lose their current progress of their quiz in case they lose internet connection.
- This problem is overcome by “Quizizz” because this platform can be used in both offline and online mode. Teachers are able to download their quiz and make it available as an offline quiz for the students.

# CHAPTER 3: DESIGN

## 3.1 STORYBOARD

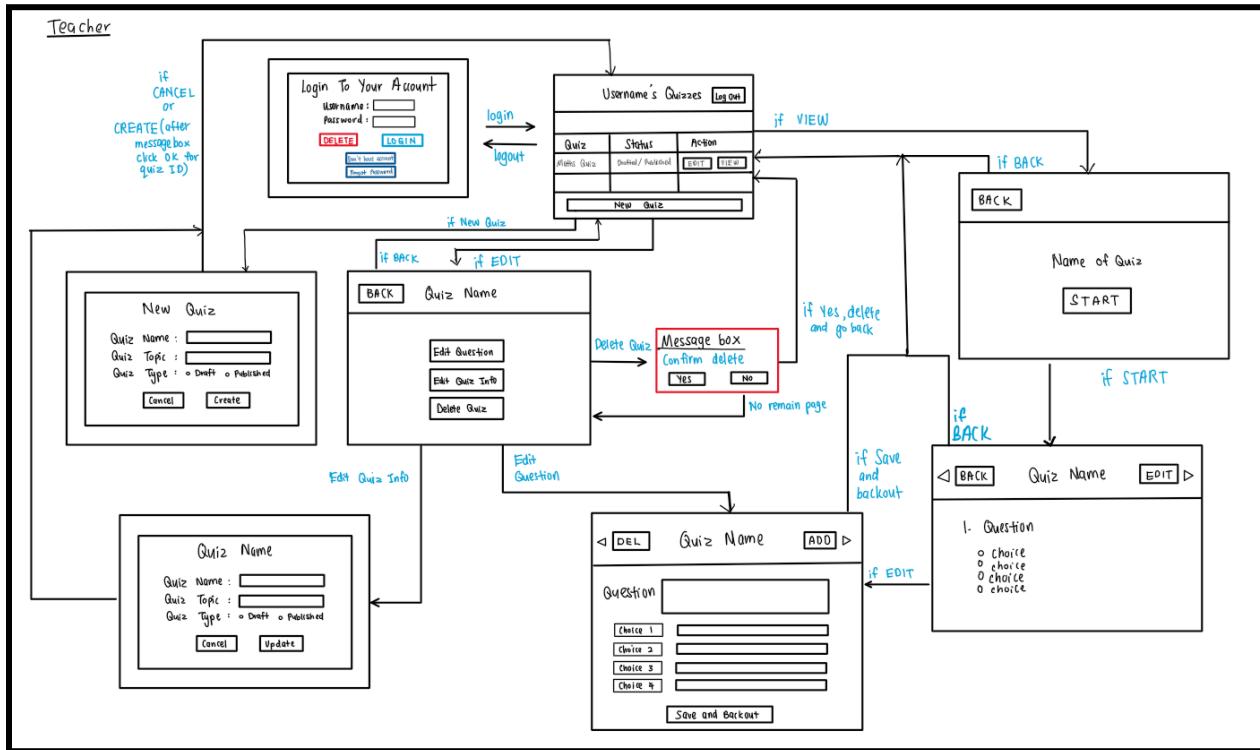


Figure 3.1.1: Teacher's View Storyboard

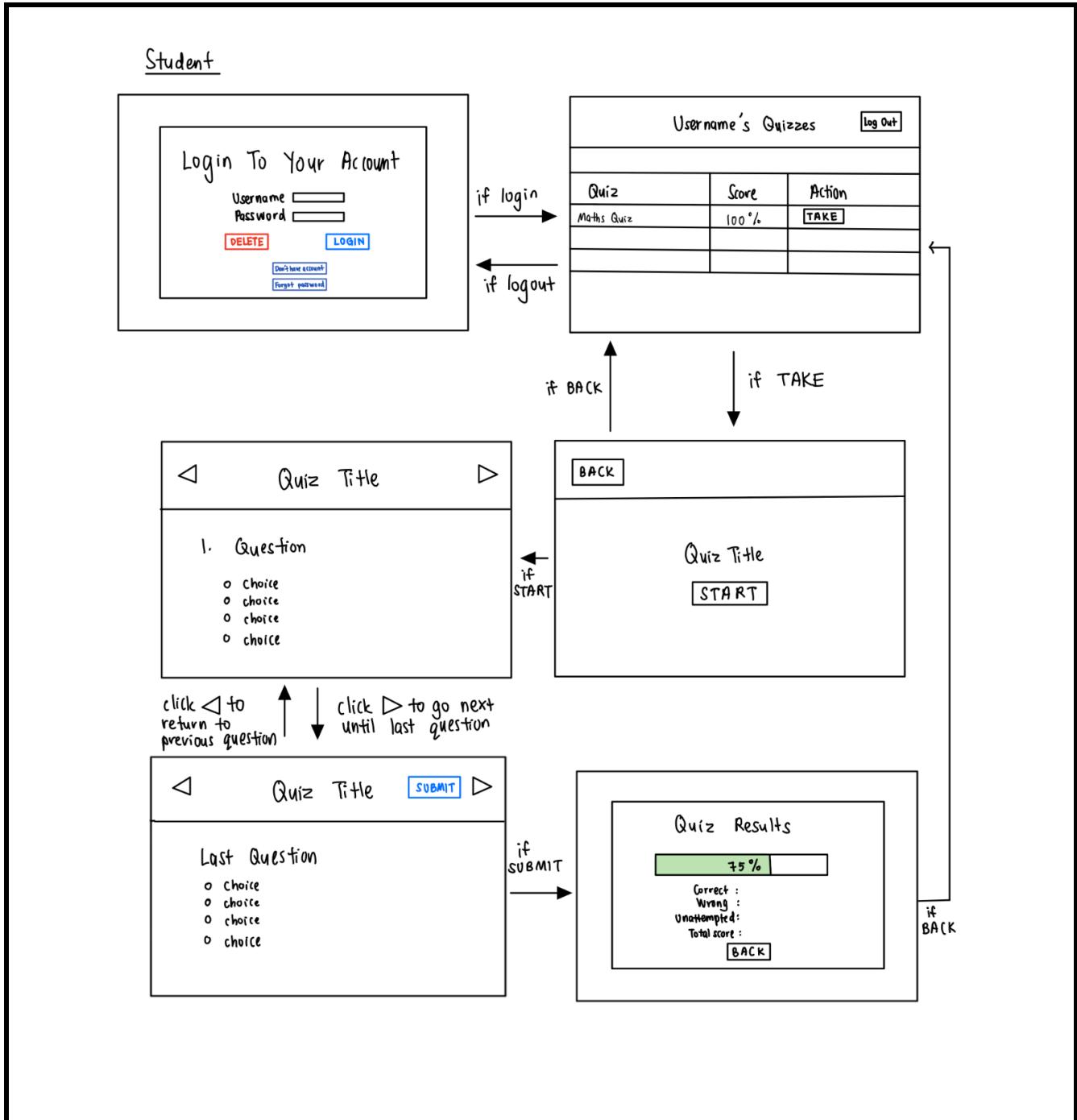


Figure 3.1.2: Student's View Storyboard

## 3.2 FLOWCHART

---

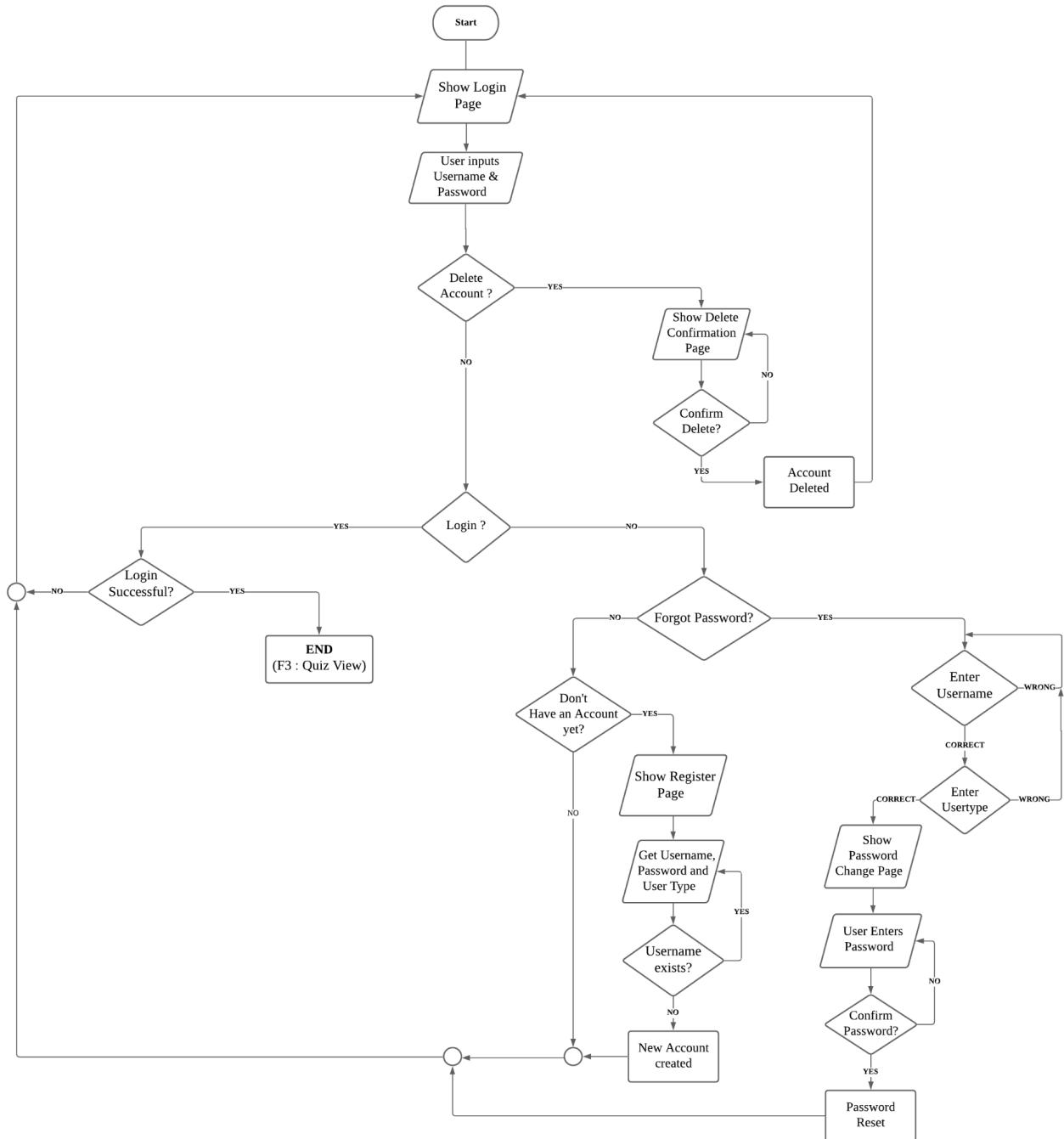
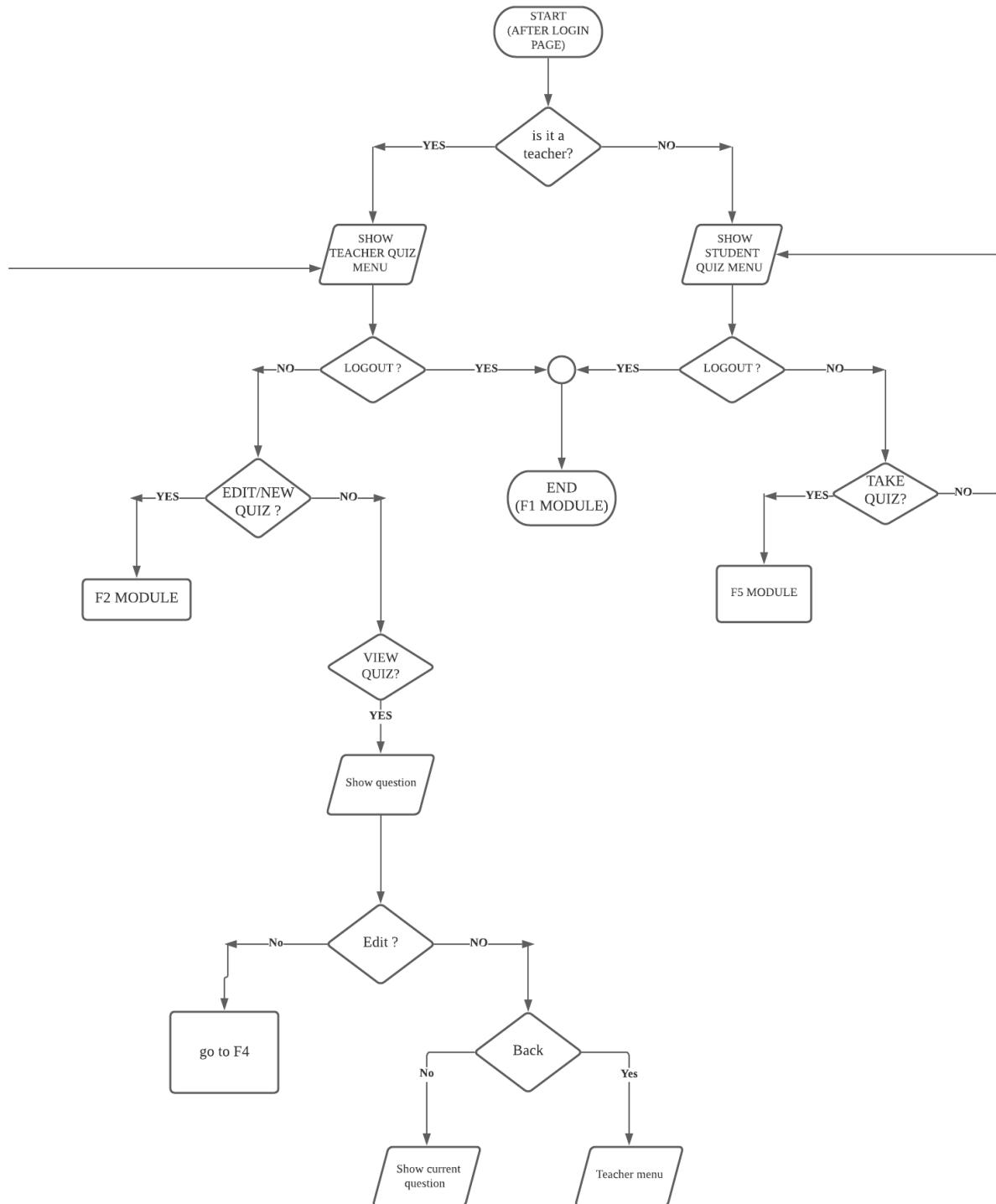
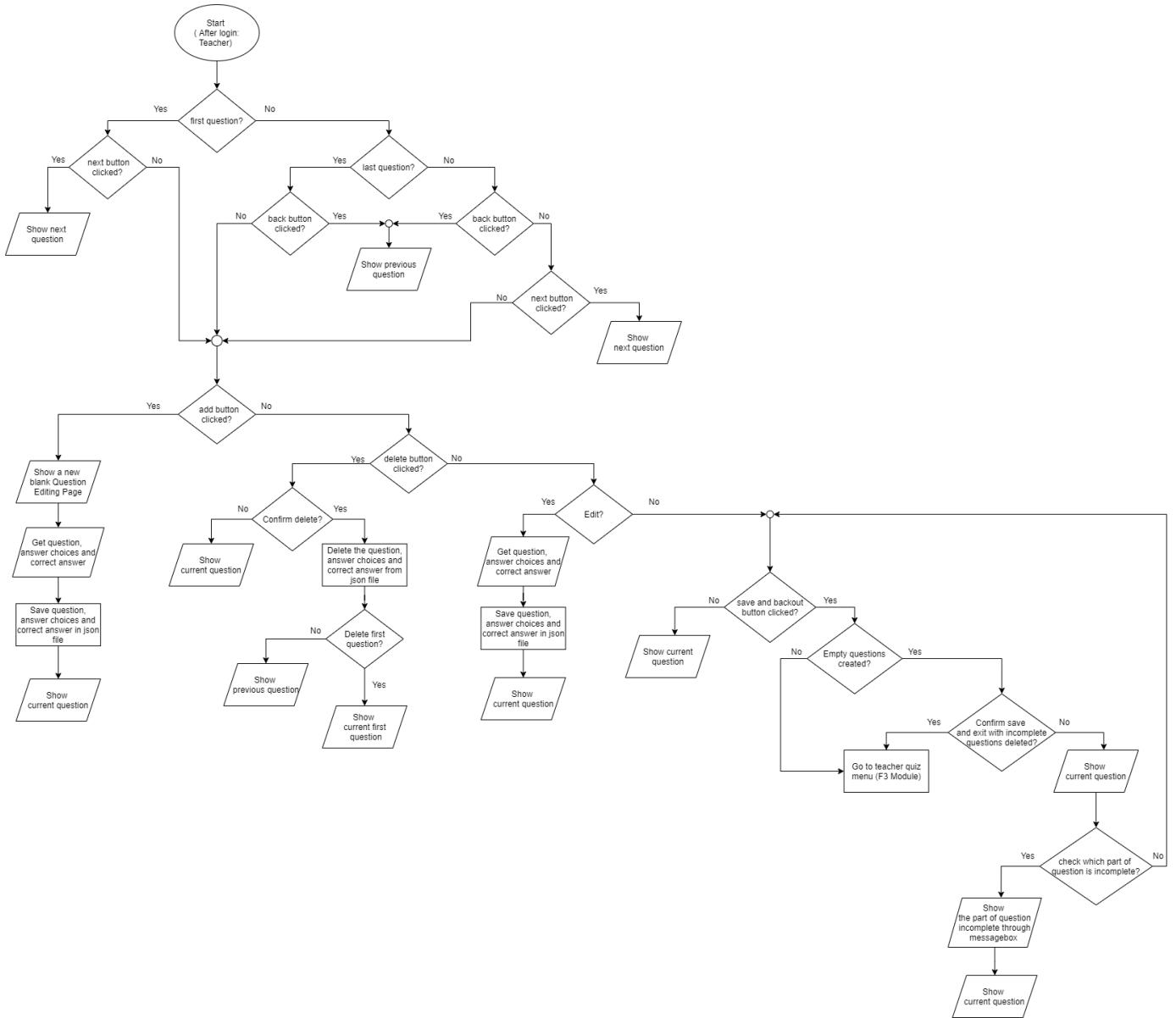


Figure 3.2.1: F1 Flowchart

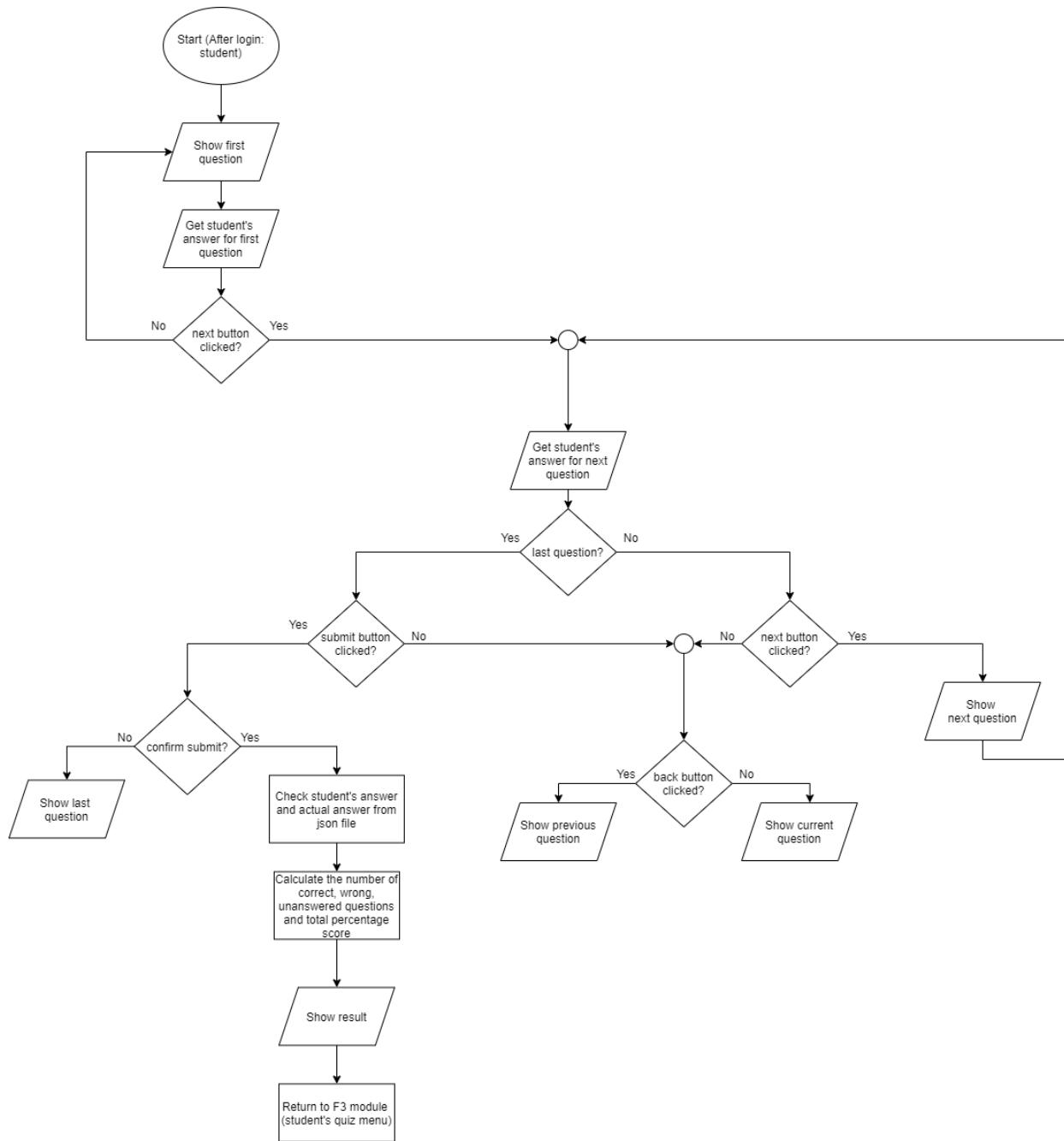
**Figure 3.2.2: F2 Flowchart**



**Figure 3.2.3: F3 Flowchart**



**Figure 3.2.4: F4 Flowchart**



**Figure 3.2.5: F5 Flowchart**

### 3.3 DATABASE / PERSISTENT STORAGE

OBJECT	ATTRIBUTES	DESCRIPTION	DATA TYPE
USER	Username	Name of the user	String
	Password	Password of the user	String
	User type	Type of user (Teacher or Student)	String
QUIZ (TEACHER)	Teacher's Name	Name of teacher	String
	Quiz Name	Name of Quiz	String
	Quiz Topic	Topic of Quiz	String
	ID	Unique Identifier (Auto-generated Quiz ID)	String
	Teacher's Quiz Status (D/P)	Type of Quiz (Draft or Published)	String
QUIZ (STUDENT)	Student's Name	Name of student	String
	Quiz Name	Name of Quiz	String
	Quiz Topic	Topic of Quiz	String
	ID	Unique Identifier (Quiz ID)	String
	Quiz Score	Percentage Score of the Quiz	String
	Student's Quiz Status (Taken)	Quiz Taken (true or false)	Boolean
QUESTION	Question	Quiz Question	String
	Choices	Answer choices for the quiz questions	String
	Answers	The supposed correct answer for the questions	String

## 3.4 USER MANUAL

---

### 1. Logging In

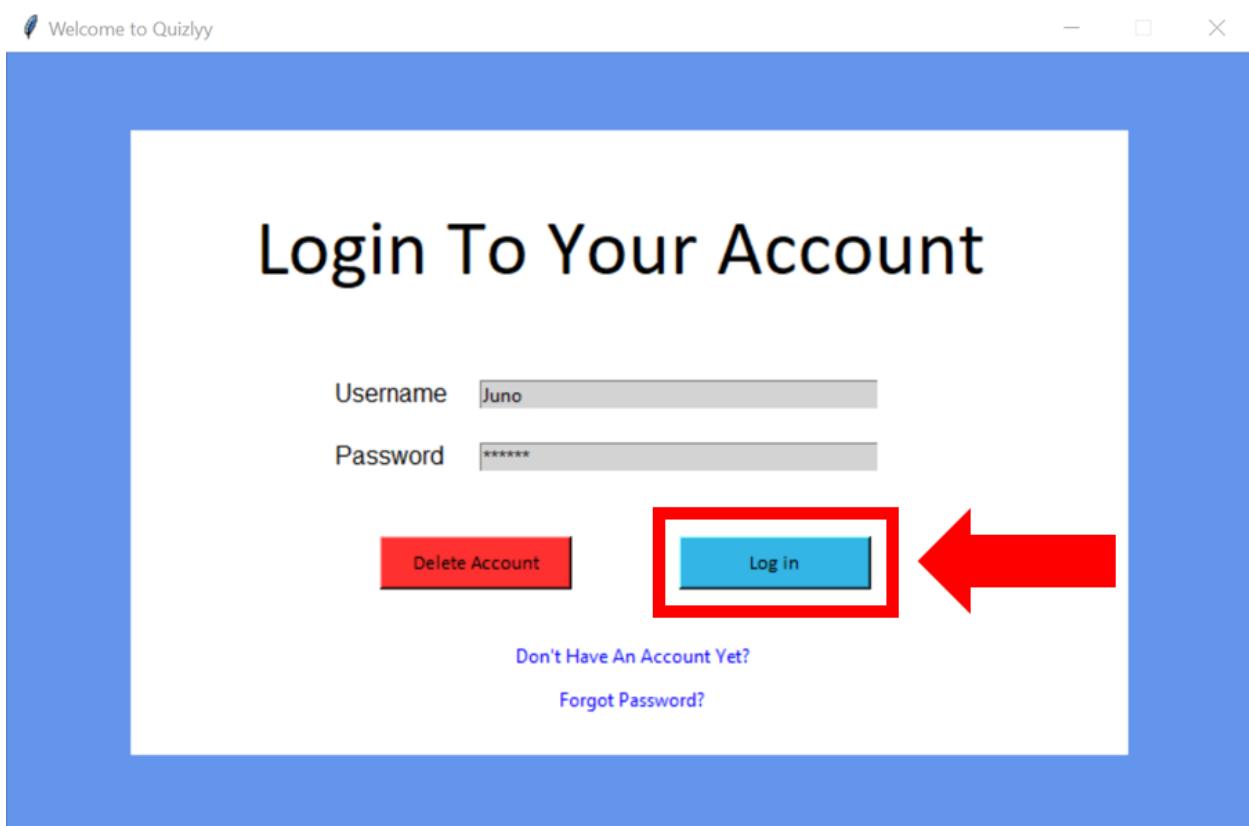


Figure 3.4.1: Login Page

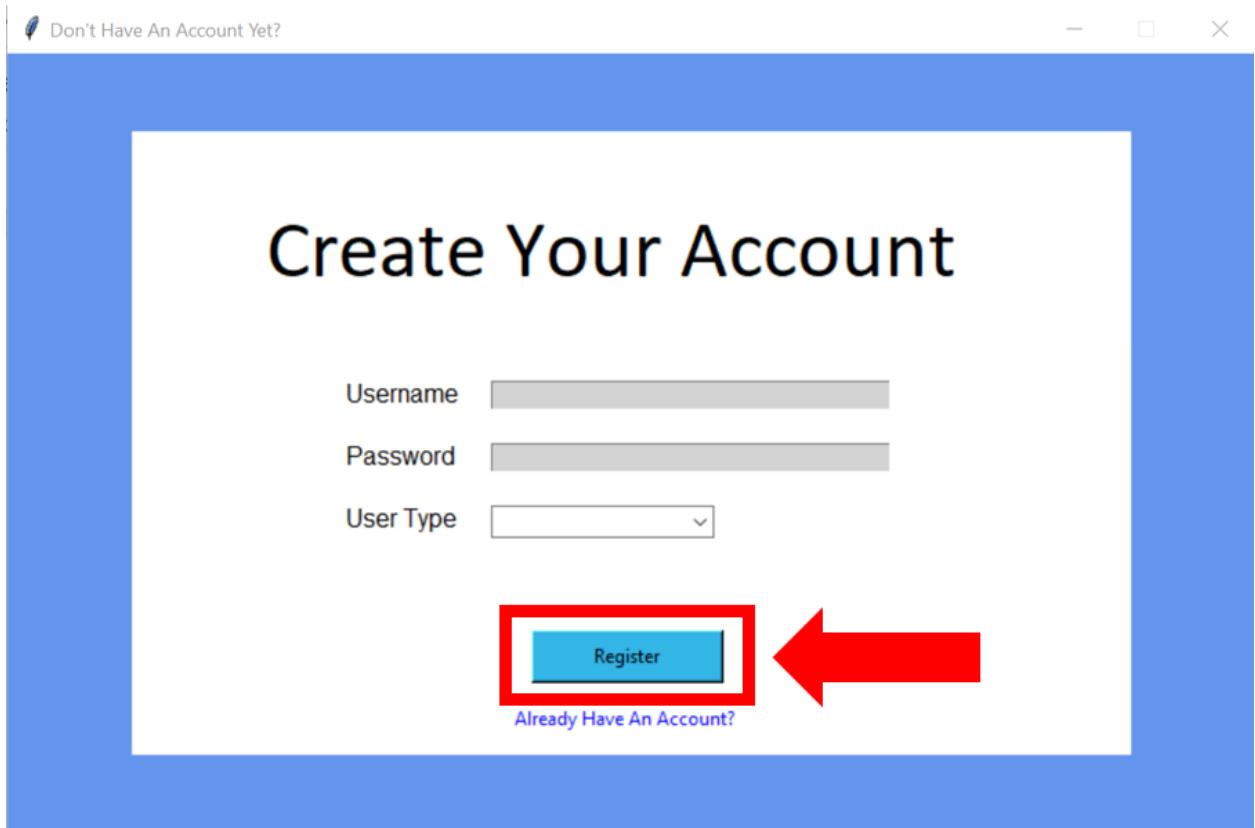
To login, existing users type in their username and password in the columns provided and click on the “Log in” button.

## 2. Creating a new account



Figure 3.4.2: Login Page

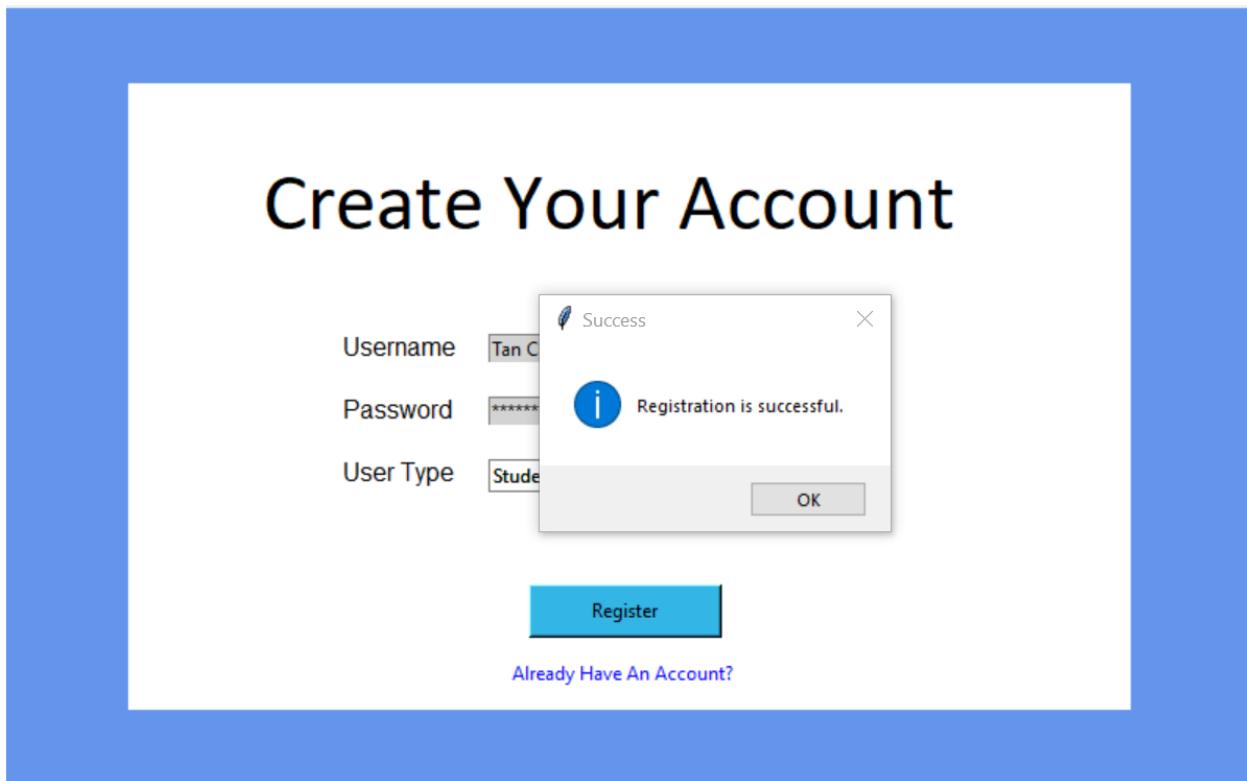
To create a new account, new users first click on the “Don’t Have An Account Yet?” button in the Login Page.



**Figure 3.4.3: Create New Account Page**

New users will then be redirected to this Create New Account Page. Users should enter their username, password and choose their user type either student or teacher. After filling up all the required fields, users click on the “Register” button to create a new account.

 Don't Have An Account Yet?



**Figure 3.4.4: Create New Account Page**

If the registration is successful, users will see a message box stating that “Registration is successful”. After confirming it, users will then be redirected to the Login Page to continue logging in.

### 3. Changing password

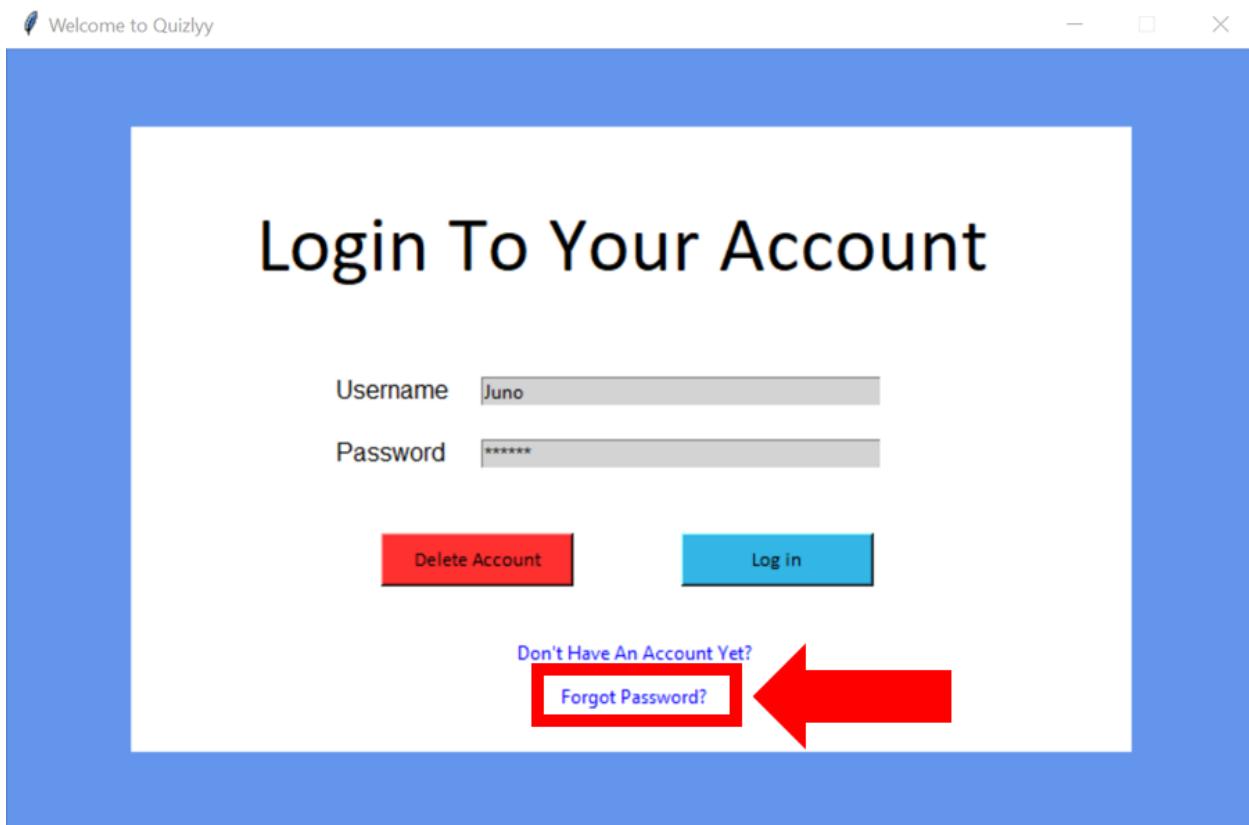


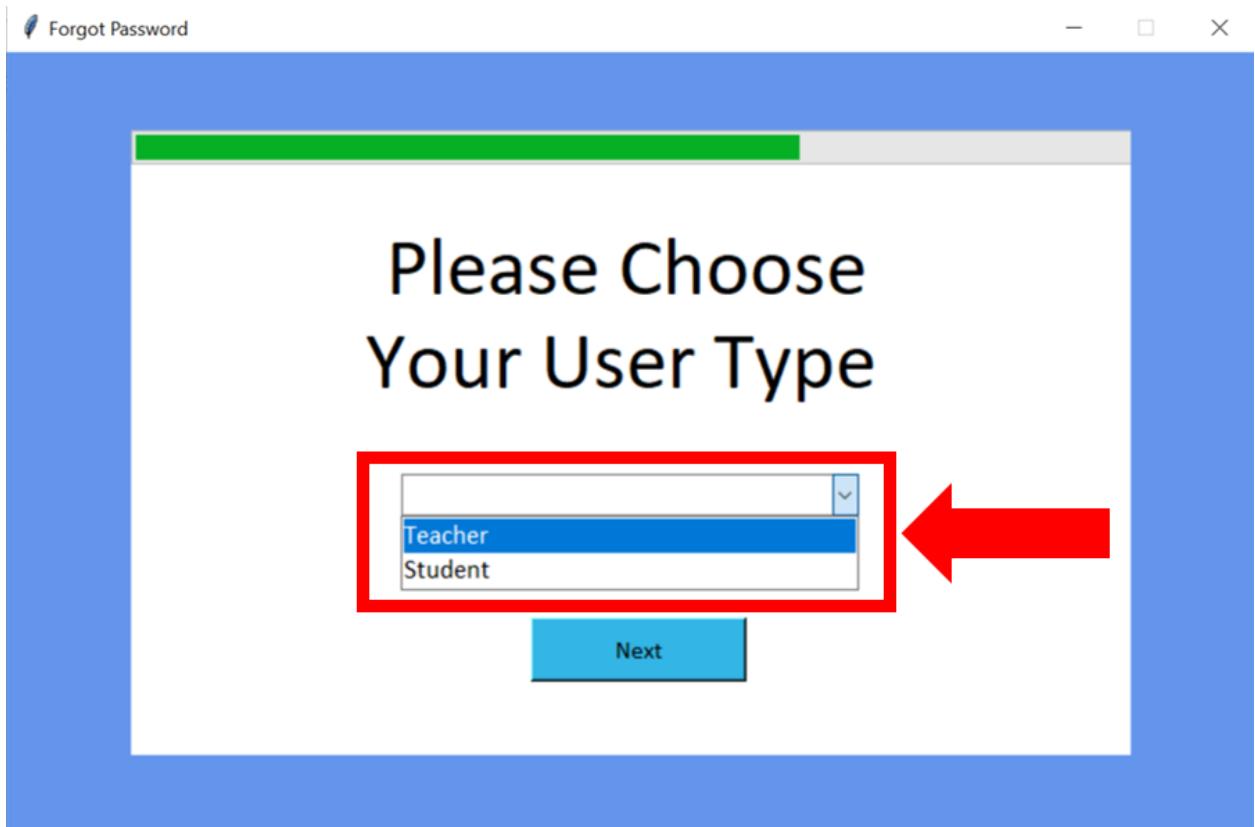
Figure 3.4.5: Login Page

Users can change their password by first clicking the “Forgot Password?” button in the Login Page.



**Figure 3.4.6: Forgot Password Verifying Username Page**

Users must then go through some verification process by first entering their username before clicking the “Next” button.



**Figure 3.4.7: Forgot Password Verifying User Type Page**

Users continue to choose their user type and click the “Next” button. If their user type does not match with username, they will be redirected to Forgot Password Verifying Username Page where they need to re-enter their username again.

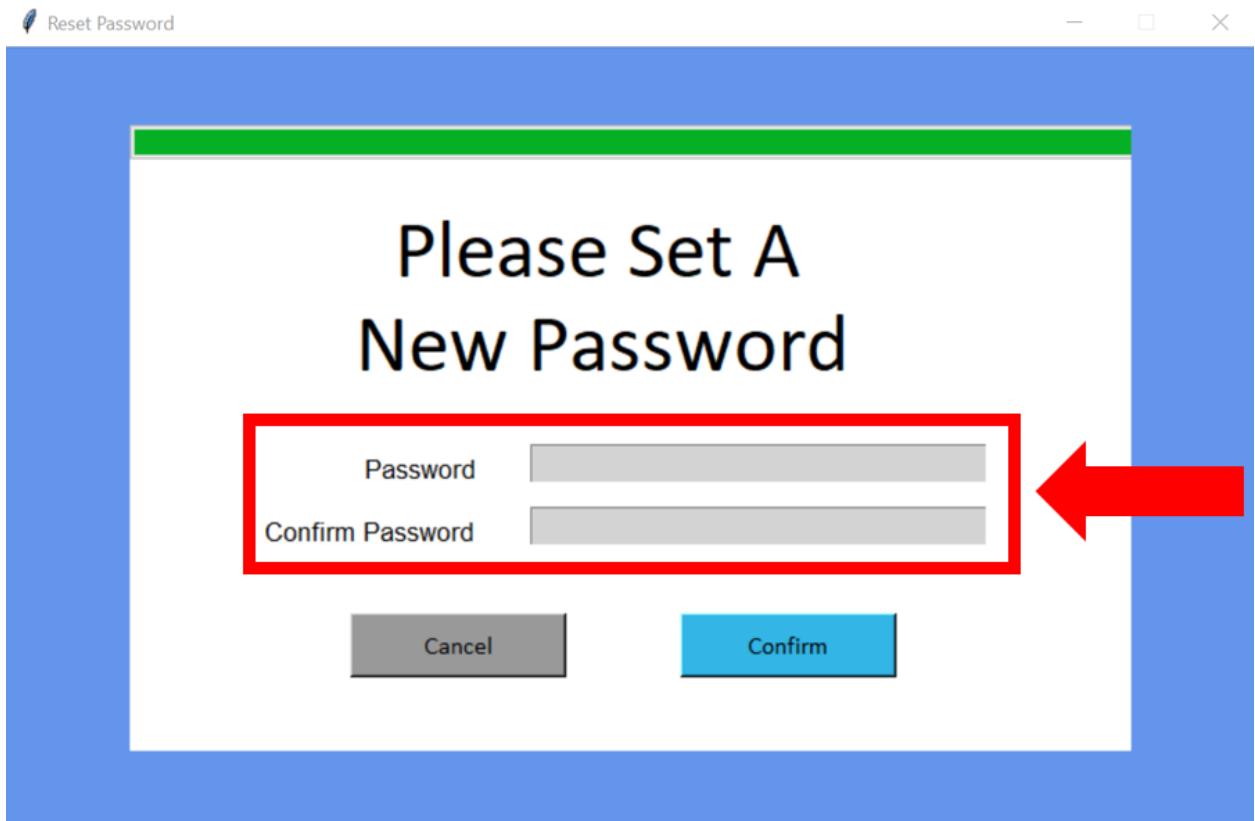
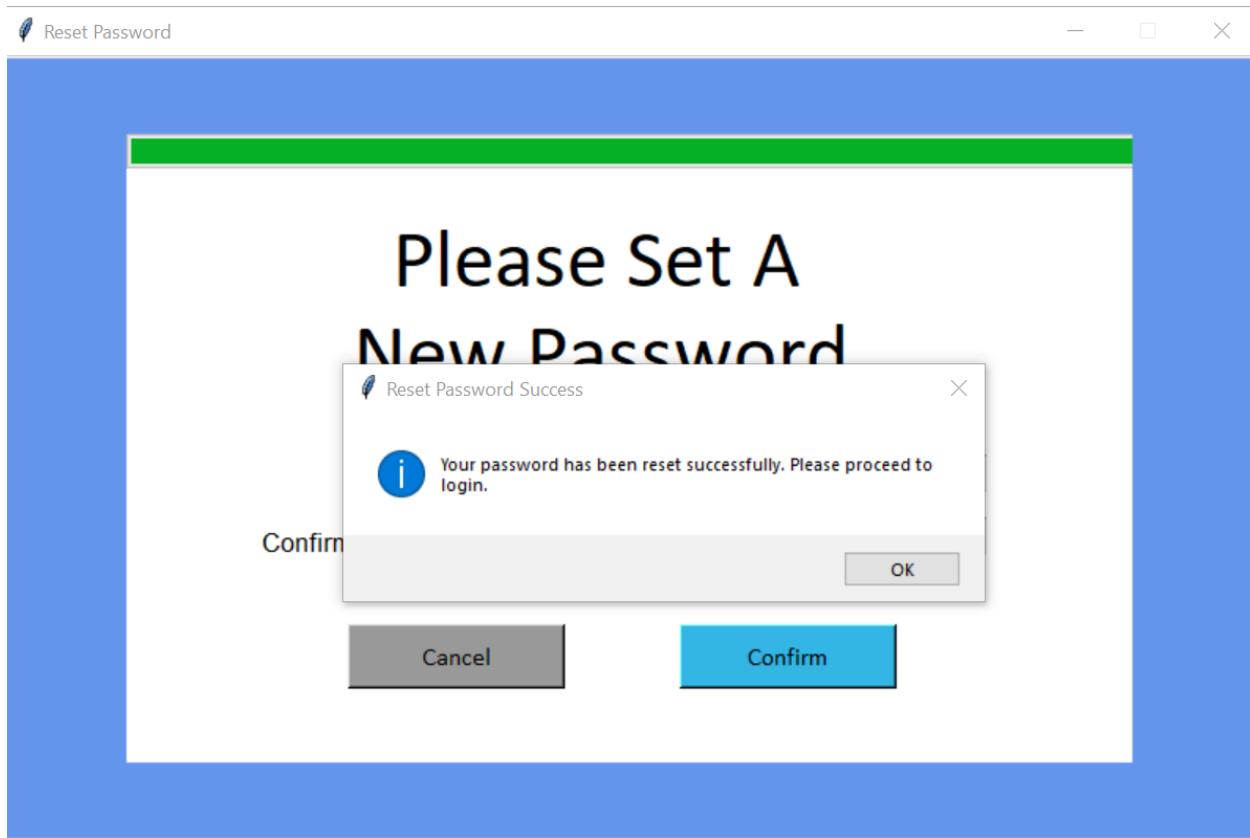


Figure 3.4.8: Reset Password Page

After users successfully verified their account, they can now set their new password. Users should fill in their new password in both columns provided and click the “Confirm” button to continue.



**Figure 3.4.9: Reset Password Page**

If the password is changed successfully, users will be able to see a message box and upon confirming, they will be redirected to the Login Page.

#### 4. Deleting Account

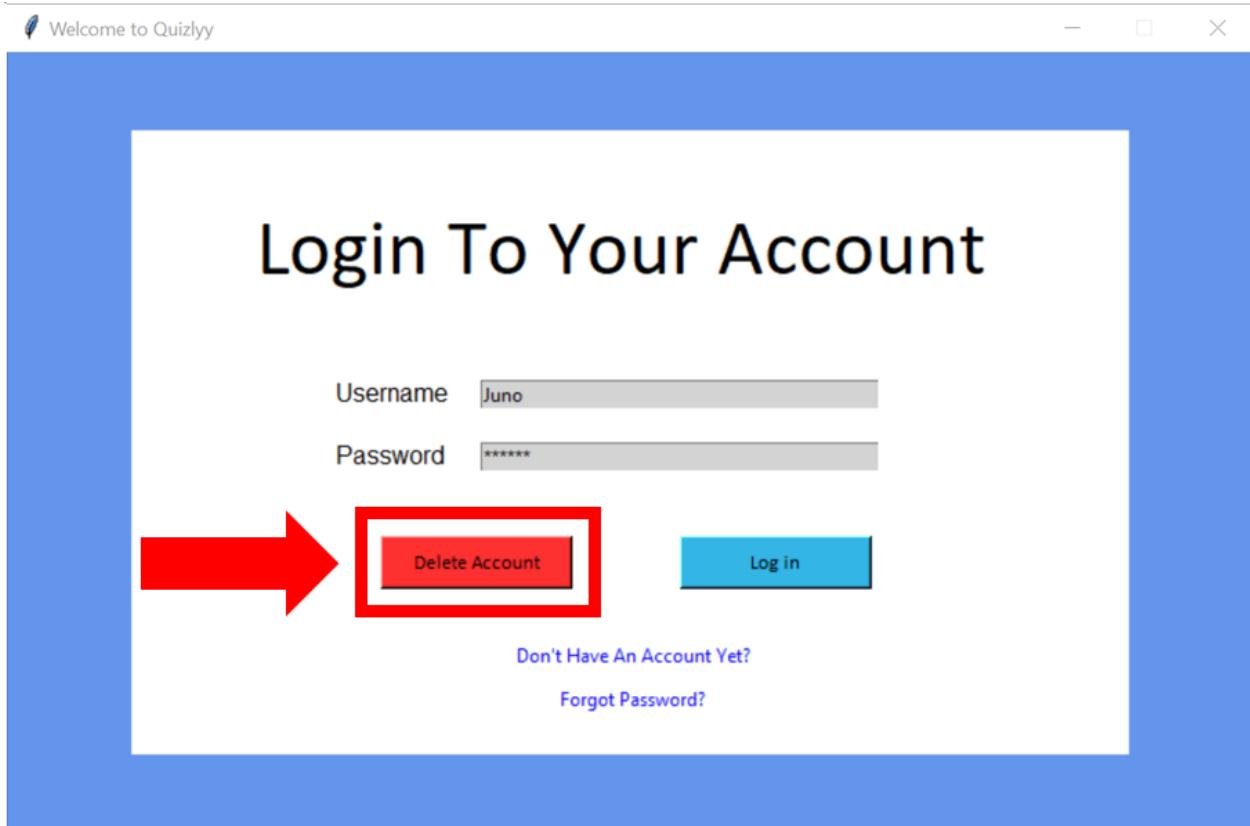
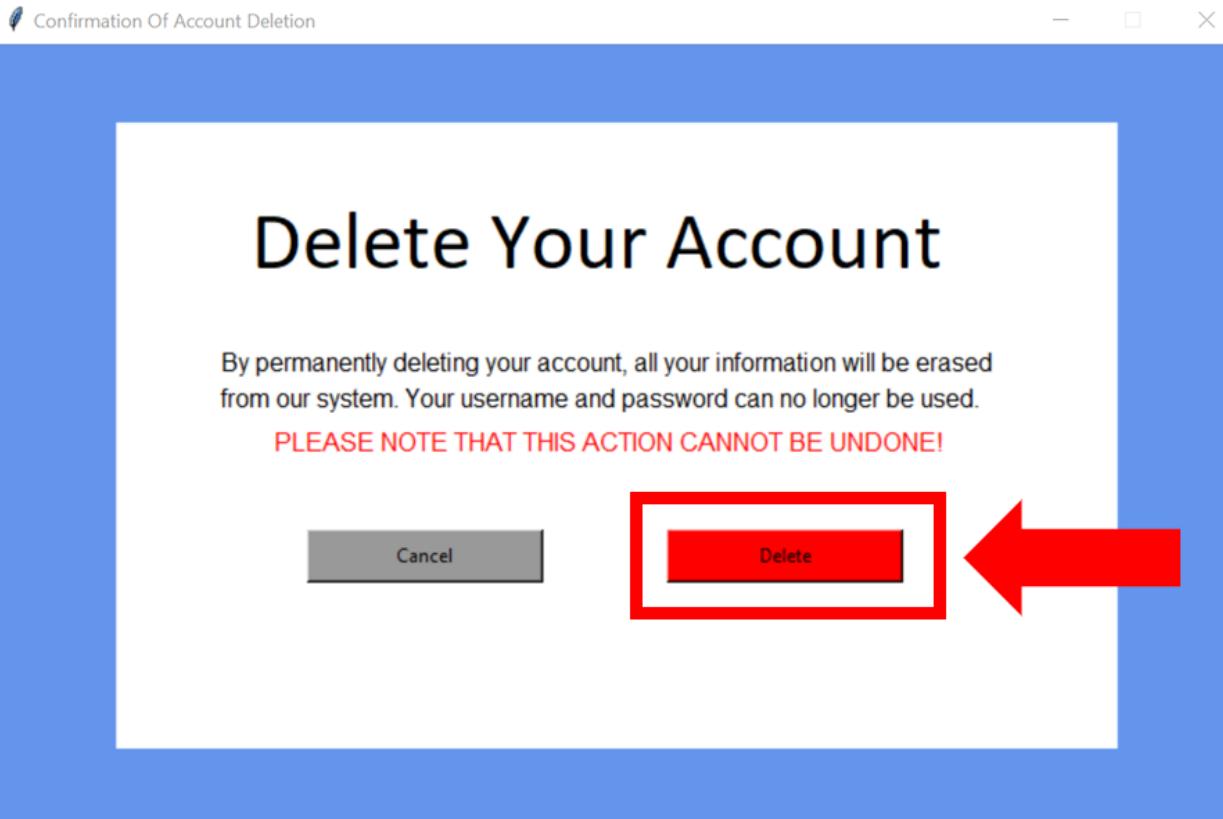


Figure 3.4.10: Login Page

To delete their account, users should enter their username and password before clicking on the “Delete Account” button.



**Figure 3.4.11: Delete Confirmation Page**

Users should click on the “Delete” button to continue account deletion after reading all the details about account deletion.

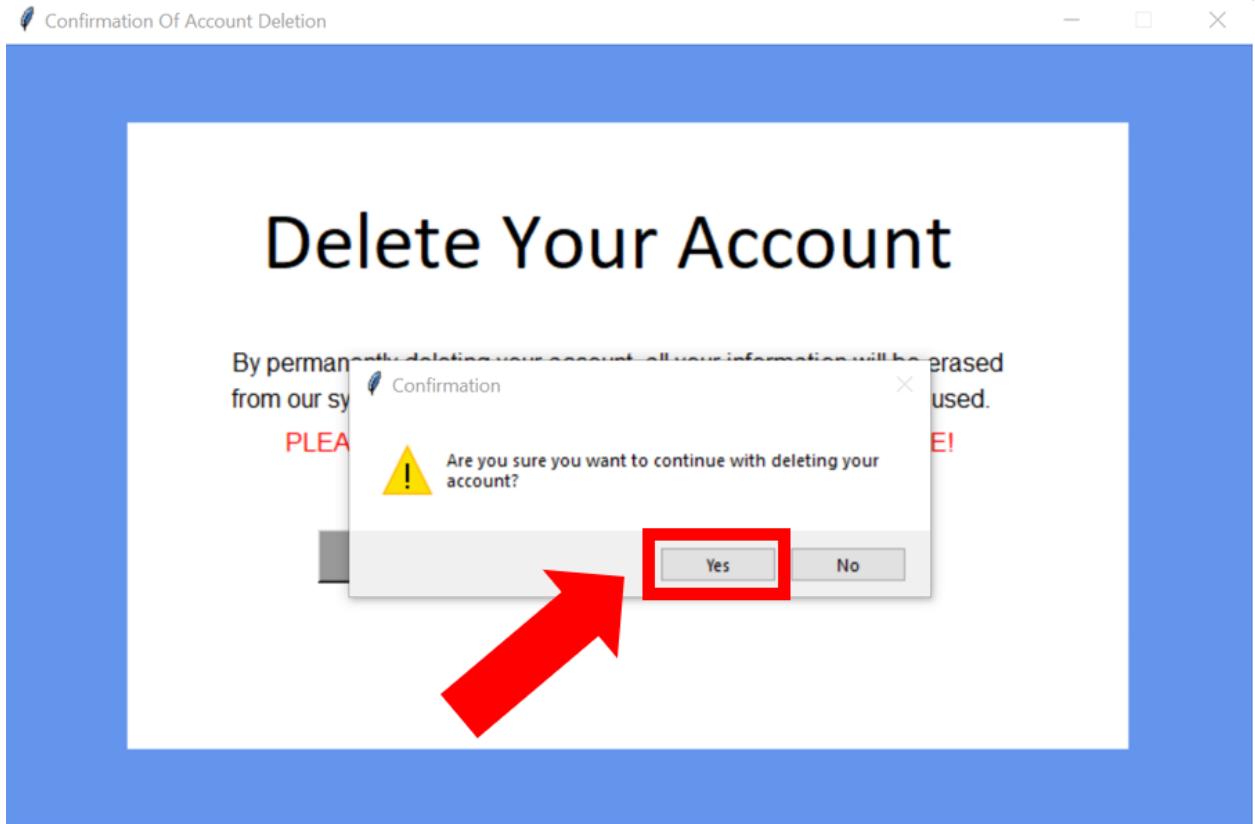


Figure 3.4.12: Delete Confirmation Page

Users click on the “Yes” button in the messagebox prompted to permanently delete their account from the system. Another message box stating that “Your account has been permanently deleted.” will also be shown after clicking “Yes”.

## 5. Logging Out

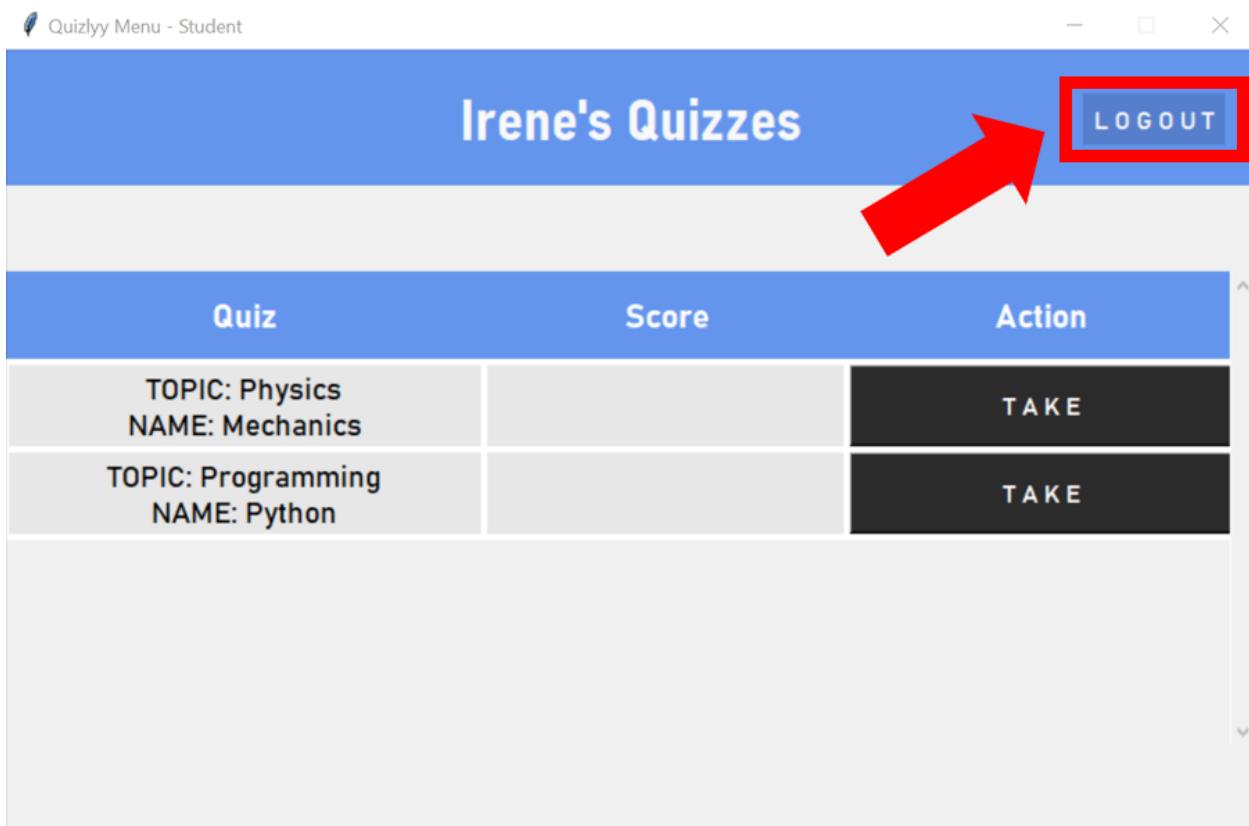


Figure 3.4.13: Student's Quiz Menu Page

To log out from their account, students click on the “Logout” button.

The screenshot shows a web-based application titled "Quizlyy Menu - Teacher". The main title is "Robert's Quizzes". Below the title is a table with three columns: "Quiz", "Status", and "Action".

Quiz	Status	Action
TOPIC: Physics NAME: Mechanics	Published	MANAGE VIEW
TOPIC: Programming NAME: Python	Published	MANAGE VIEW
TOPIC: Science NAME: Physics Quiz	Draft	MANAGE VIEW

At the bottom of the table is a blue button labeled "ADD QUIZ". In the top right corner of the page, there is a "LOGOUT" button, which is highlighted with a red box and a red arrow pointing towards it.

Figure 3.4.14: Teacher's Quiz Menu Page

To log out from their account, teachers click on the “Logout” button.

## 6. Taking A New/Attempted Quiz (Student)

The screenshot shows a web-based application titled "Quizlyy Menu - Student". At the top, there is a blue header bar with the title "Irene's Quizzes" and a "LOGOUT" button. Below the header is a table with three columns: "Quiz", "Score", and "Action". There are two rows in the table. The first row contains the text "TOPIC: Physics" and "NAME: Mechanics" under the "Quiz" column, and a large empty rectangular box under the "Score" column. The "Action" column contains a button labeled "TAKE" which is highlighted with a red rectangular border. The second row contains the text "TOPIC: Programming" and "NAME: Python" under the "Quiz" column, and a large empty rectangular box under the "Score" column. The "Action" column contains a button labeled "TAKE". A large red arrow points from the bottom left towards the "TAKE" button in the first row.

Quiz	Score	Action
TOPIC: Physics NAME: Mechanics		TAKE
TOPIC: Programming NAME: Python		TAKE

Figure 3.4.15: Student's Quiz Menu Page

After logging in into their account, students can take any unattempted quiz assigned to them by first clicking the “Take” button.

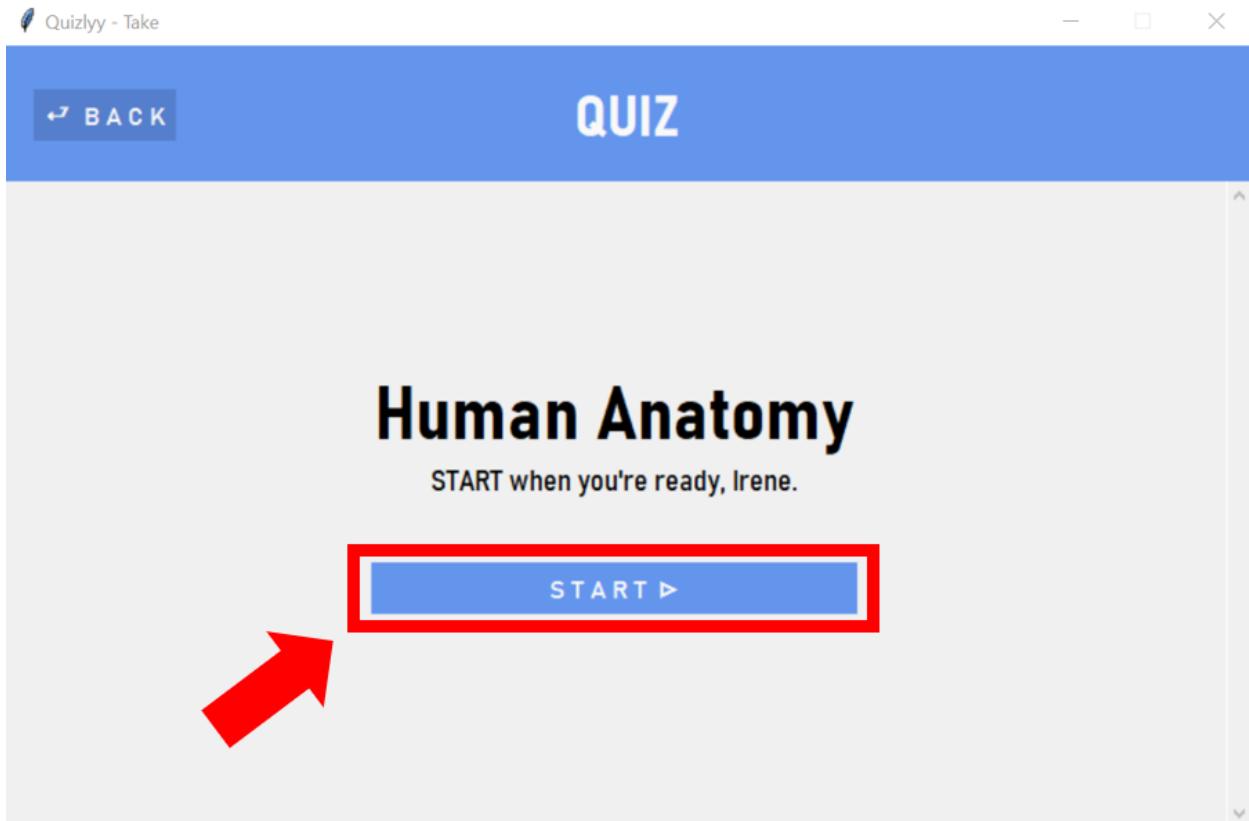
## Irene's Quizzes

[LOGOUT](#)

Quiz	Score	Action
TOPIC: Physics NAME: Mechanics		TAKE
TOPIC: Programming NAME: Python		TAKE
TOPIC: Biology NAME: Human Anatomy	100%	RETAKE

Figure 3.4.16: Student's Quiz Menu Page

Students can also retake attempted quizzes by clicking the “RETAKE” button.



**Figure 3.4.17: Quiz Start Page**

Students should click the “Start” button to start doing the quiz. Students will be able to see the questions in the quiz after clicking the “Start” button.

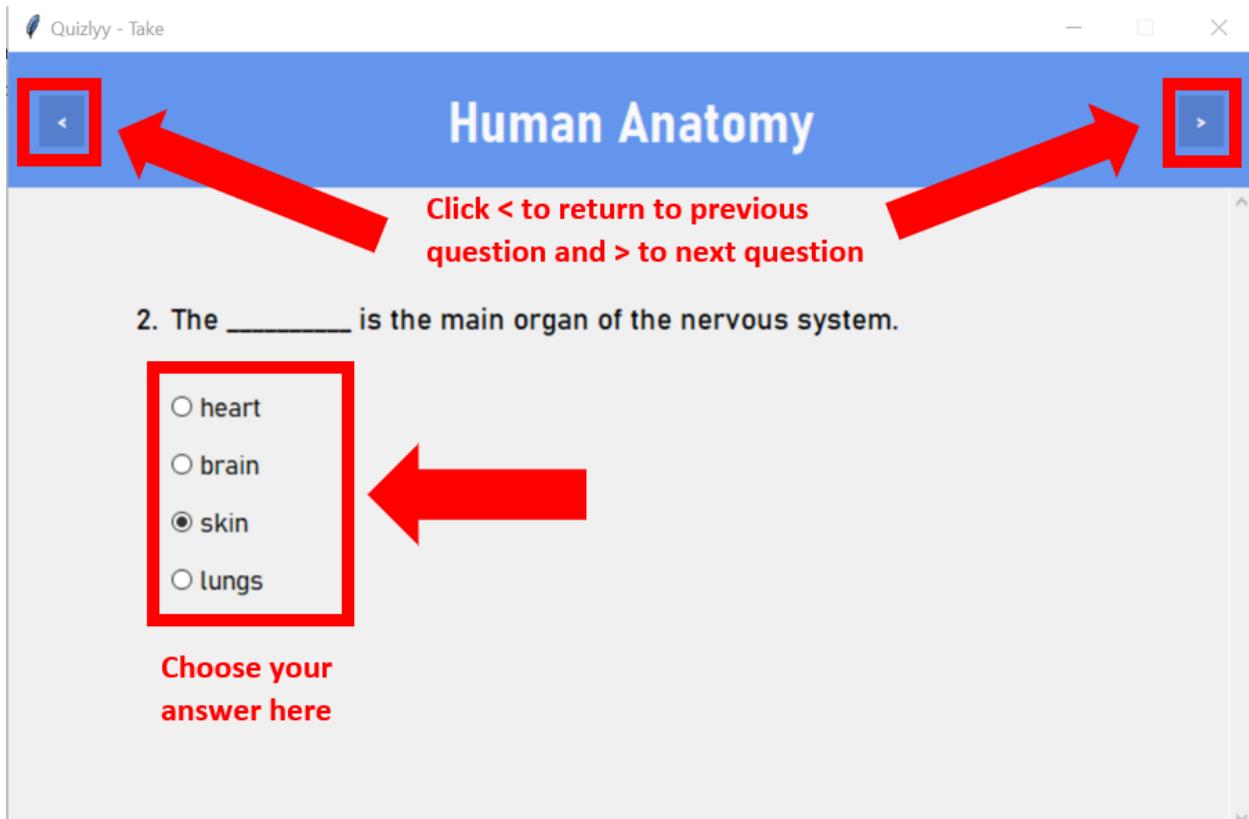


Figure 3.4.18: Quiz Page

Students can choose their answer by clicking on the option they want to select and navigate between questions using the “<” and “>” buttons.

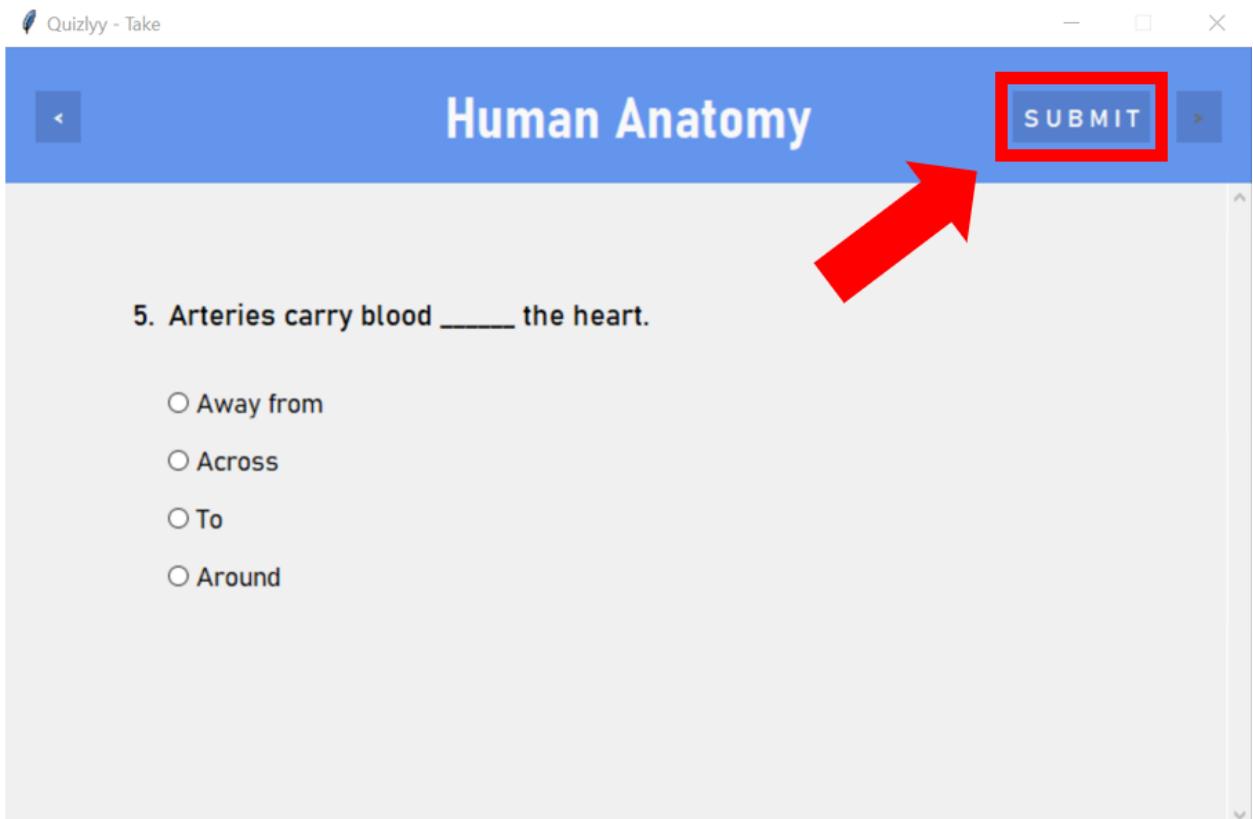


Figure 3.4.19: Quiz Page

When students reach the last question in the quiz, they will be able to see the “SUBMIT” button. Students click the “SUBMIT” button to submit the quiz.

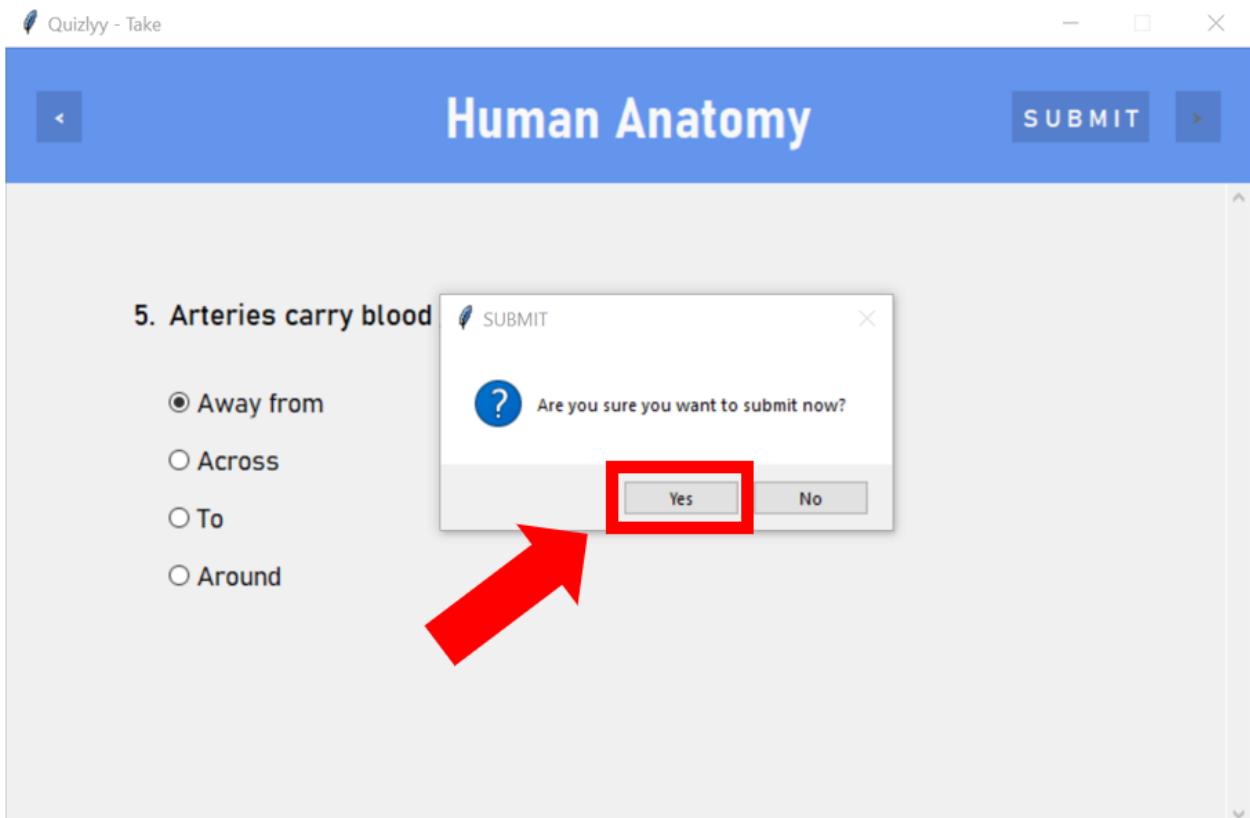


Figure 3.4.20: Quiz Page

After clicking the “SUBMIT” button, students will be asked to confirm their submission through a message box. Students click “Yes” to confirm submission.

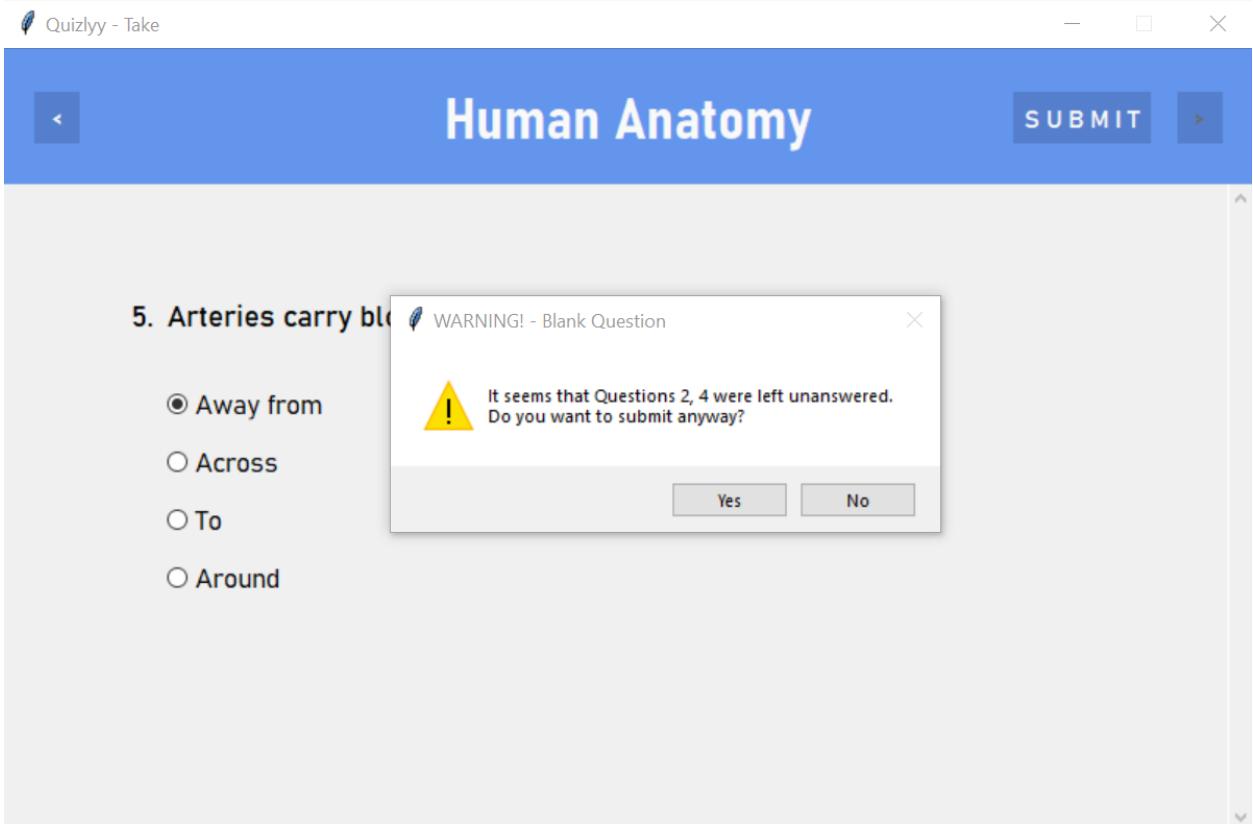


Figure 3.4.21: Quiz Page

If there are questions unanswered, they will also be reminded to complete them before submitting the quiz.

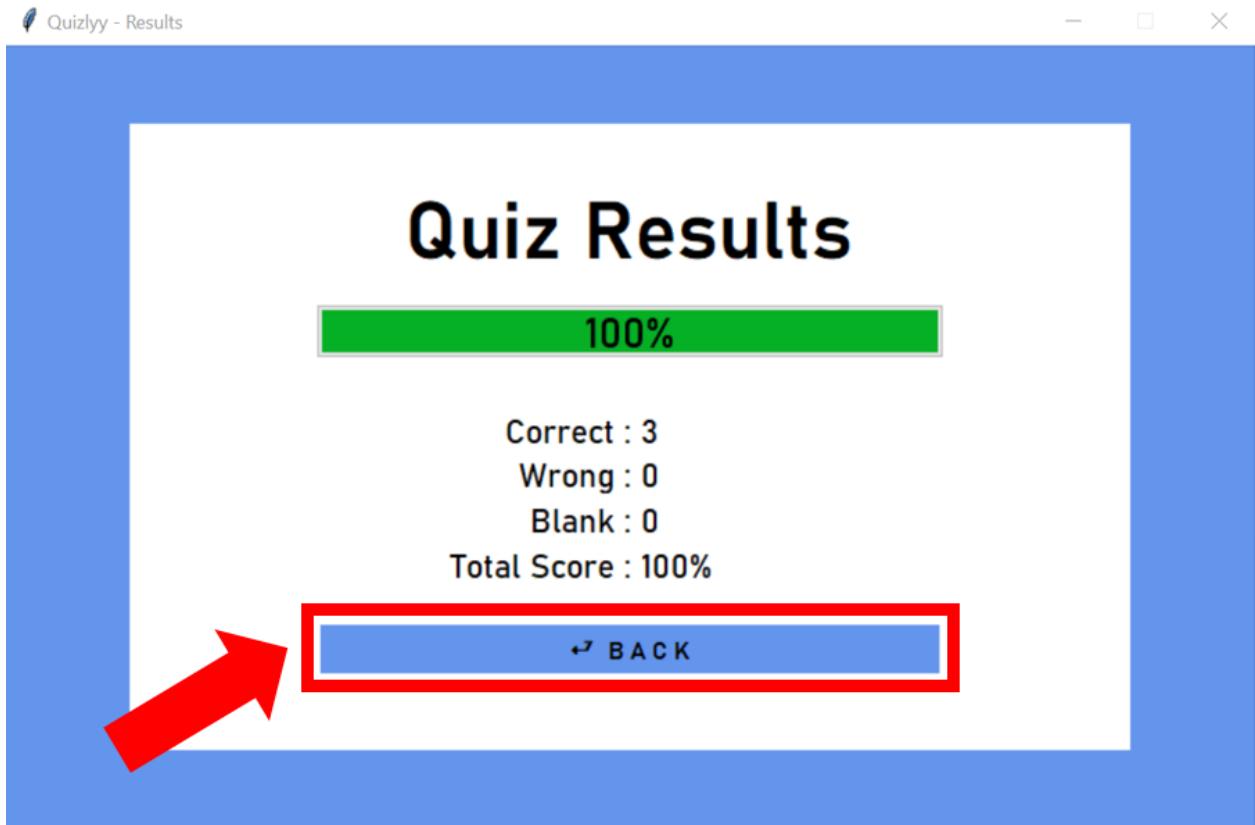


Figure 3.4.22: Quiz Results Page

After they have confirmed their submission, students will be able to see the quiz results. After seeing their quiz results, they click on the “BACK” button to return to the Student Quiz Menu Page.

## 7. Viewing New/Attempted Quiz (Student)

Quizlyy Menu - Student

Irene's Quizzes

LOGOUT

Quiz	Score	Action
TOPIC: Physics NAME: Mechanics		TAKE
TOPIC: Programming NAME: Python		TAKE
TOPIC: Biology NAME: Human Anatomy	100%	RETAKE

Figure 3.4.23: Student's Quiz Menu Page

Students can see a list of attempted and unattempted quizzes in the Student's Quiz Menu Page.

Quizlyy Menu - Student

## Irene's Quizzes

LOGOUT

Quiz	Score	Action
TOPIC: Physics NAME: Mechanics	40%	RETAKE
TOPIC: Programming NAME: Python	100%	RETAKE
TOPIC: Biology NAME: Human Anatomy	100%	RETAKE

Figure 3.4.24: Student's Quiz Menu Page

Students can also see their score for each attempted quiz in the Student's Quiz Menu Page.

## 8. Creating A New Quiz (Teacher)

The screenshot shows a web application titled "Robert's Quizzes" with a blue header bar. On the left is a navigation menu icon and the text "Quizlyy Menu - Teacher". On the right are standard window control buttons (minimize, maximize, close) and a "LOGOUT" button. The main content area has a table with three rows, each representing a quiz. The columns are "Quiz", "Status", and "Action".

Quiz	Status	Action
TOPIC: Physics NAME: Mechanics	Published	<a href="#">MANAGE</a> <a href="#">VIEW</a>
TOPIC: Programming NAME: Python	Published	<a href="#">MANAGE</a> <a href="#">VIEW</a>
TOPIC: Biology NAME: Human Anatomy	Draft	<a href="#">MANAGE</a> <a href="#">VIEW</a>

A large red arrow points from the bottom right towards the "ADD QUIZ" button, which is highlighted with a red border. The "ADD QUIZ" button is located at the bottom of the page.

Figure 3.4.25: Teacher's Quiz Menu Page

After logging in, teachers will be redirected to the Teacher's Quiz Menu Page. They can create a new quiz by clicking on the “ADD QUIZ” button.

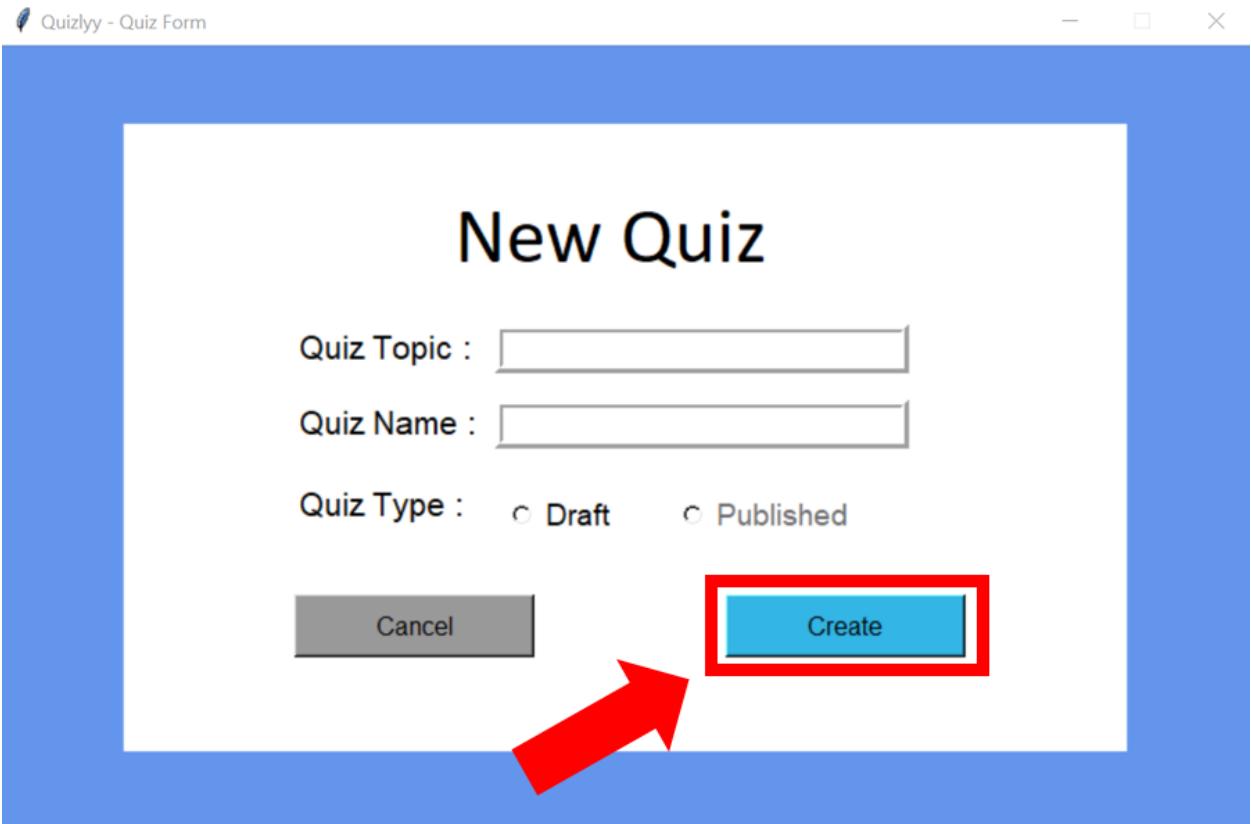


Figure 3.4.26: New Quiz Page

Teachers will then be redirected to a New Quiz Page where they can enter the new quiz topic and quiz name. Teachers can only choose the Quiz Type to be “Draft” when first creating a new quiz. After filling all the required fields, they can click on the “Create” button to create a new quiz.

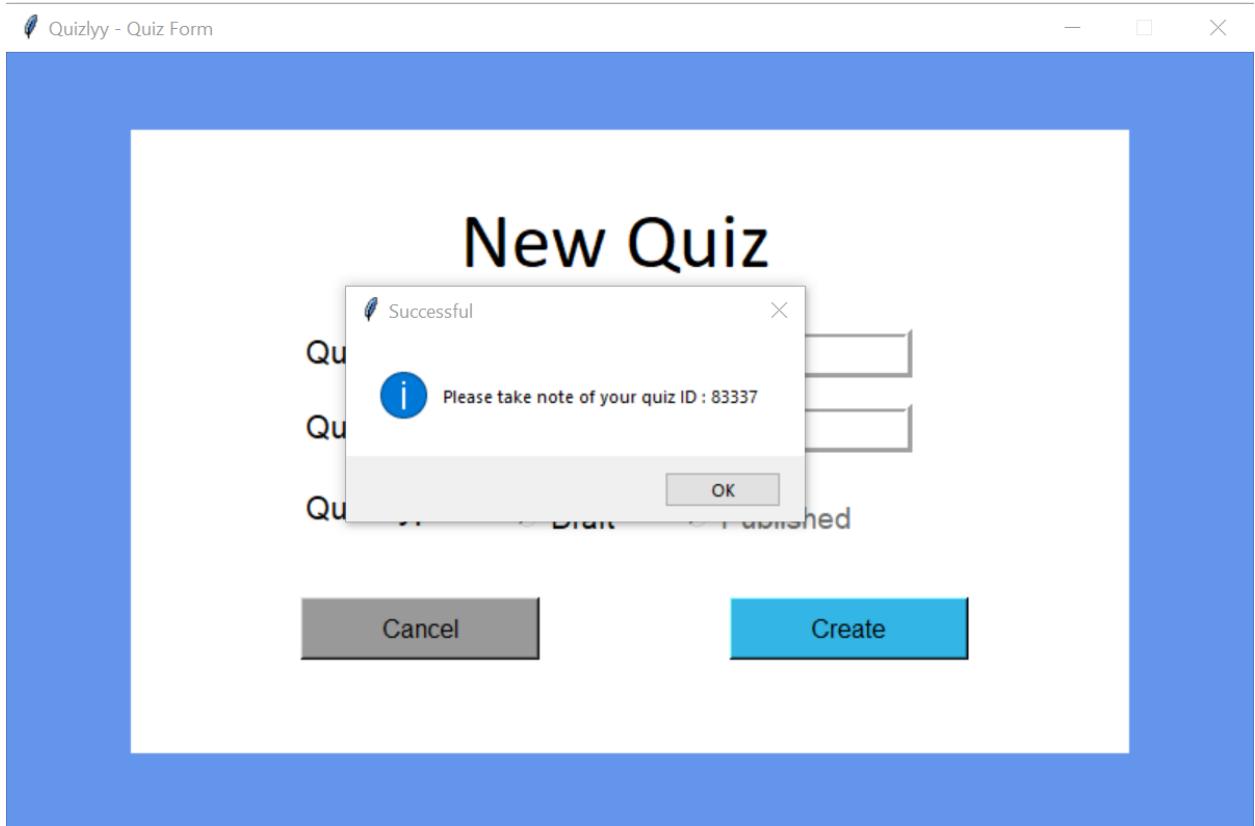


Figure 3.4.27: New Quiz Page

If the new quiz is created successfully, a new quiz ID will be shown and upon confirming it, teachers will be redirected to the Teacher's Quiz Menu Page.

## 9. Editing Quiz (Teacher)

The screenshot shows the Quizlyy Teacher's Quiz Menu Page. At the top, it displays "Robert's Quizzes" and a "LOGOUT" button. Below is a table with three rows:

Quiz	Status	Action
TOPIC: Physics NAME: Mechanics	Published	MANAGE VIEW
TOPIC: Programming NAME: Python	Published	MANAGE VIEW
TOPIC: Biology NAME: Human Anatomy	Draft	MANAGE VIEW

A red arrow points to the "MANAGE" button for the Biology quiz, which is highlighted with a red box. At the bottom of the page is a blue "ADD QUIZ" button.

Figure 3.4.28: Teacher's Quiz Menu Page

After logging in, teachers first click on the “MANAGE” button for editing existing quiz information, deleting existing quiz and editing, adding and deleting questions in the quiz.

### a. Editing Quiz Information

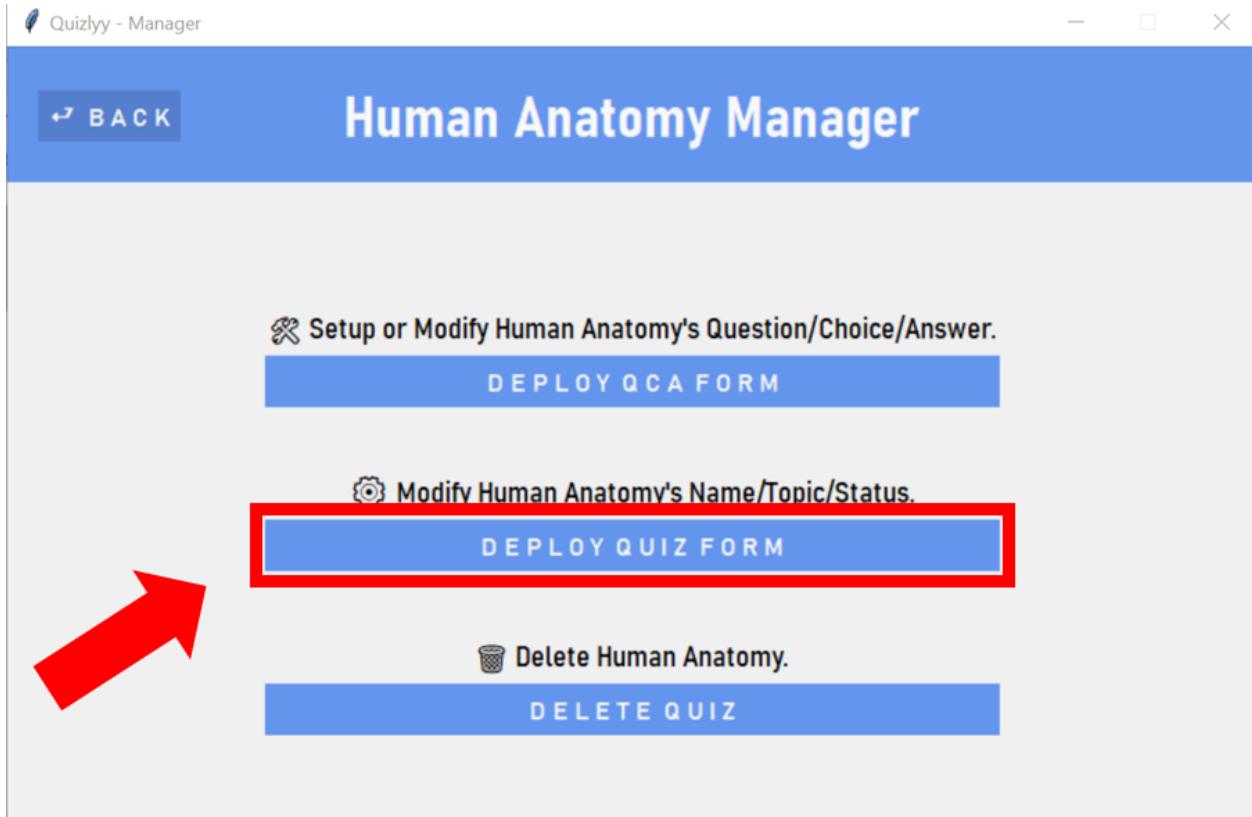


Figure 3.4.29: Quiz Manager Page

In this Quiz Manager Page, teachers can click the “DEPLOY QUIZ FORM” button to edit that quiz information such as quiz name, quiz title and quiz status.

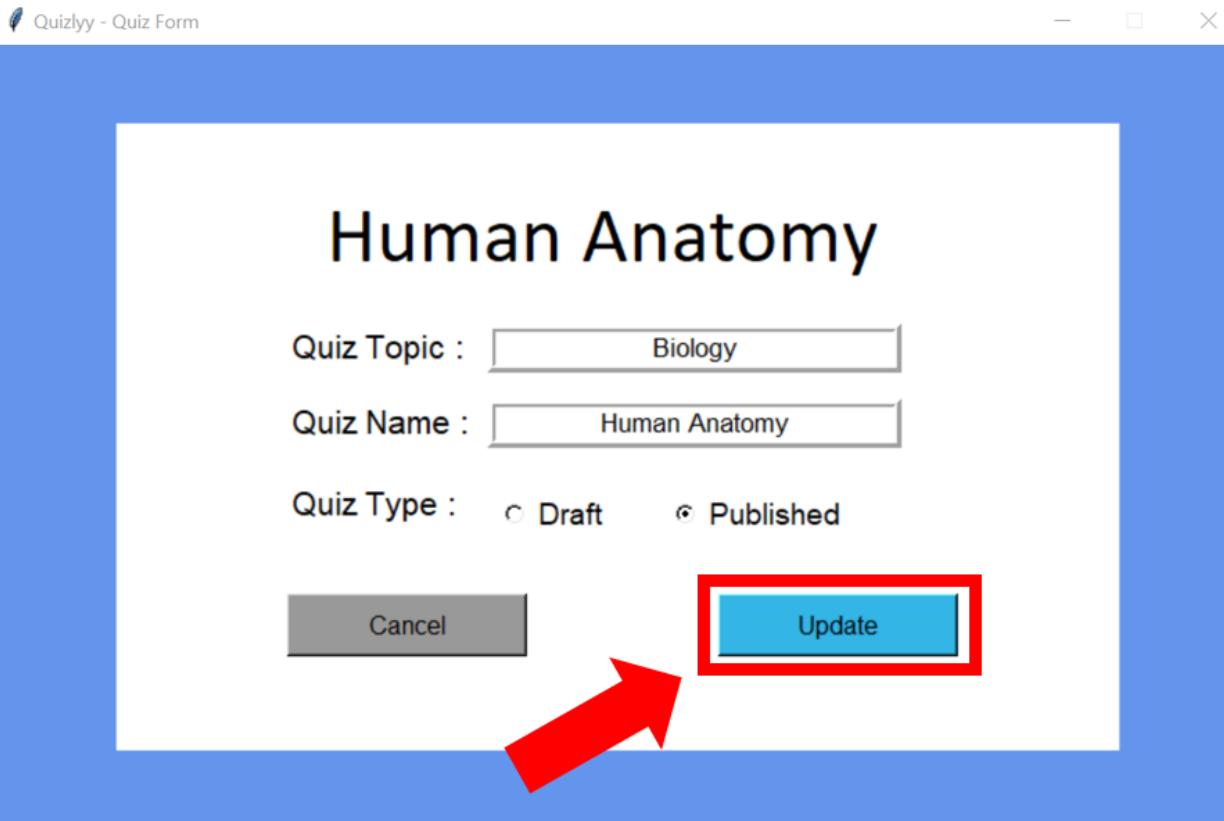


Figure 3.4.30: Edit Quiz Info Page

After clicking the “DEPLOY QUIZ FORM” button, teachers will be redirected to the corresponding quiz edit info page. They can update their quiz information after filling up all the required fields and click on the “Update” button.

b. Editing/Deleting/Adding Questions in Quiz

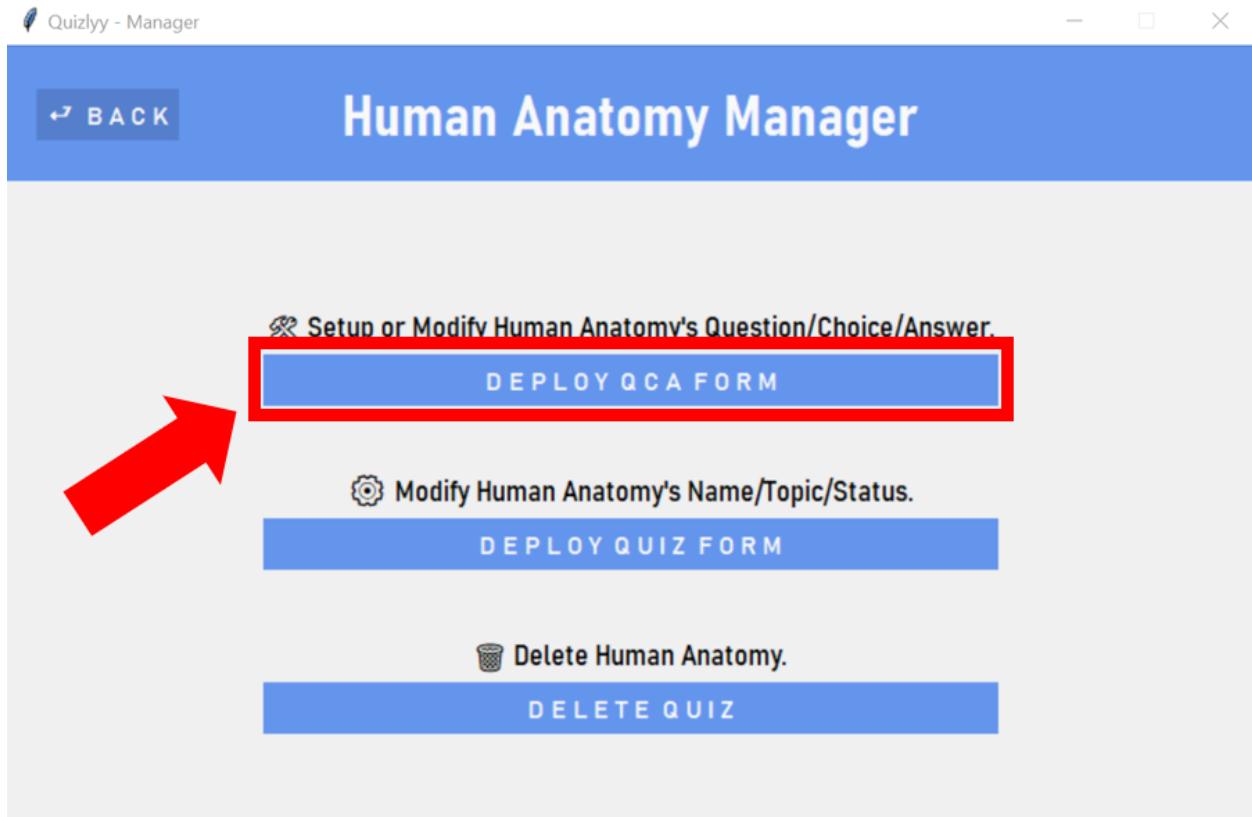


Figure 3.4.31: Quiz Manager Page

In this Quiz Manager Page, teachers can click the “DEPLOY QCA FORM” button to edit the quiz questions, answer choices and actual answer.

The screenshot shows a quiz management interface titled "Human Anatomy". A red arrow points to the question text: "The main purpose of the \_\_\_\_\_ system is to break food particles in". Another red arrow points to the "Edit answer choices here" button at the bottom right of the choices panel.

QUESTION 1

The main purpose of the \_\_\_\_\_ system is to break food particles in

CHOICE 1: circulatory  
CHOICE 2: digestive  
CHOICE 3: respiratory  
CHOICE 4: skeletal

SAVE & BACKOUT ↺

Edit answer choices here

Figure 3.4.32: Quiz Q.C.A Manager Page

In the Quiz Q.C.A Manager Page, teachers are able to edit the question and answer choices for each question in the quiz by directly typing in the columns provided.

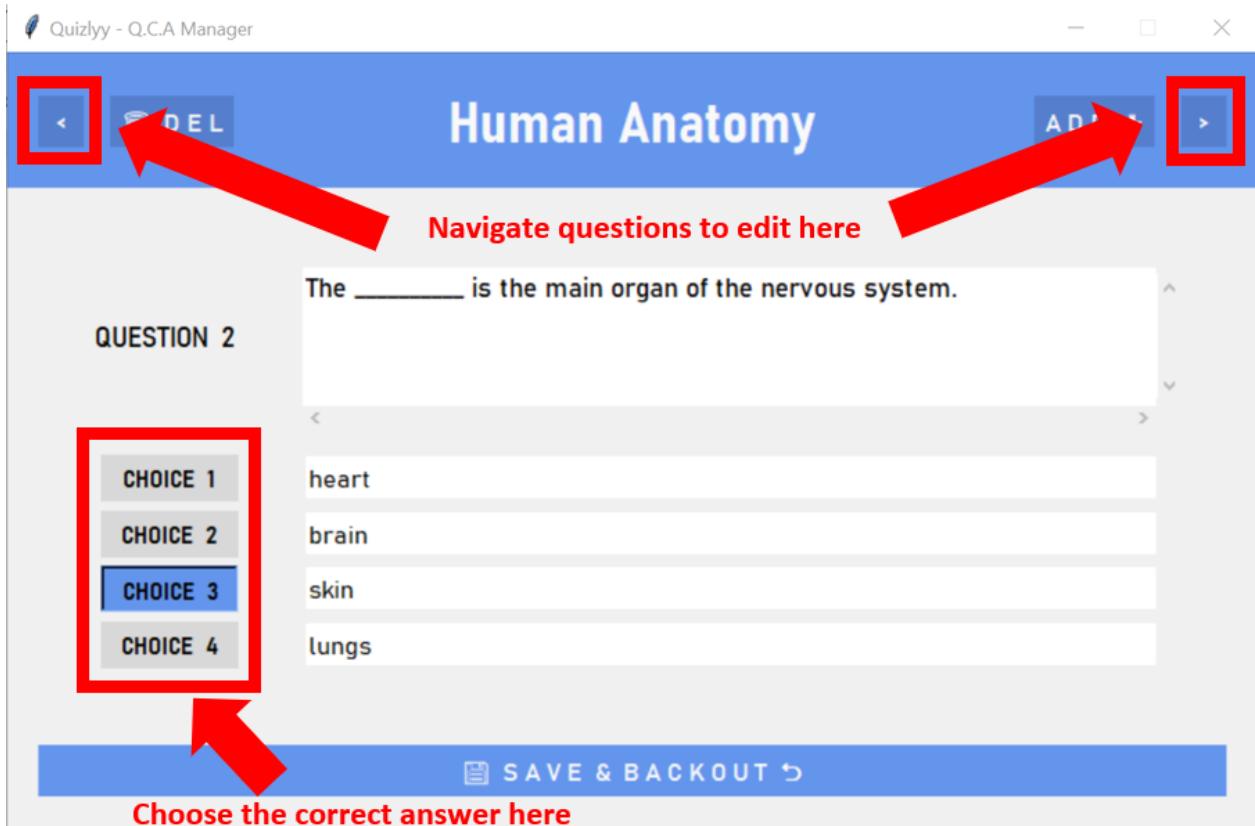


Figure 3.4.33: Quiz Q.C.A Manager Page

In the Quiz Q.C.A Manager Page, teachers can also click on the choice to choose the correct answer and navigate between questions to edit. Teachers click on “<” to return to the previous question and “>” to go to the next question in the quiz.

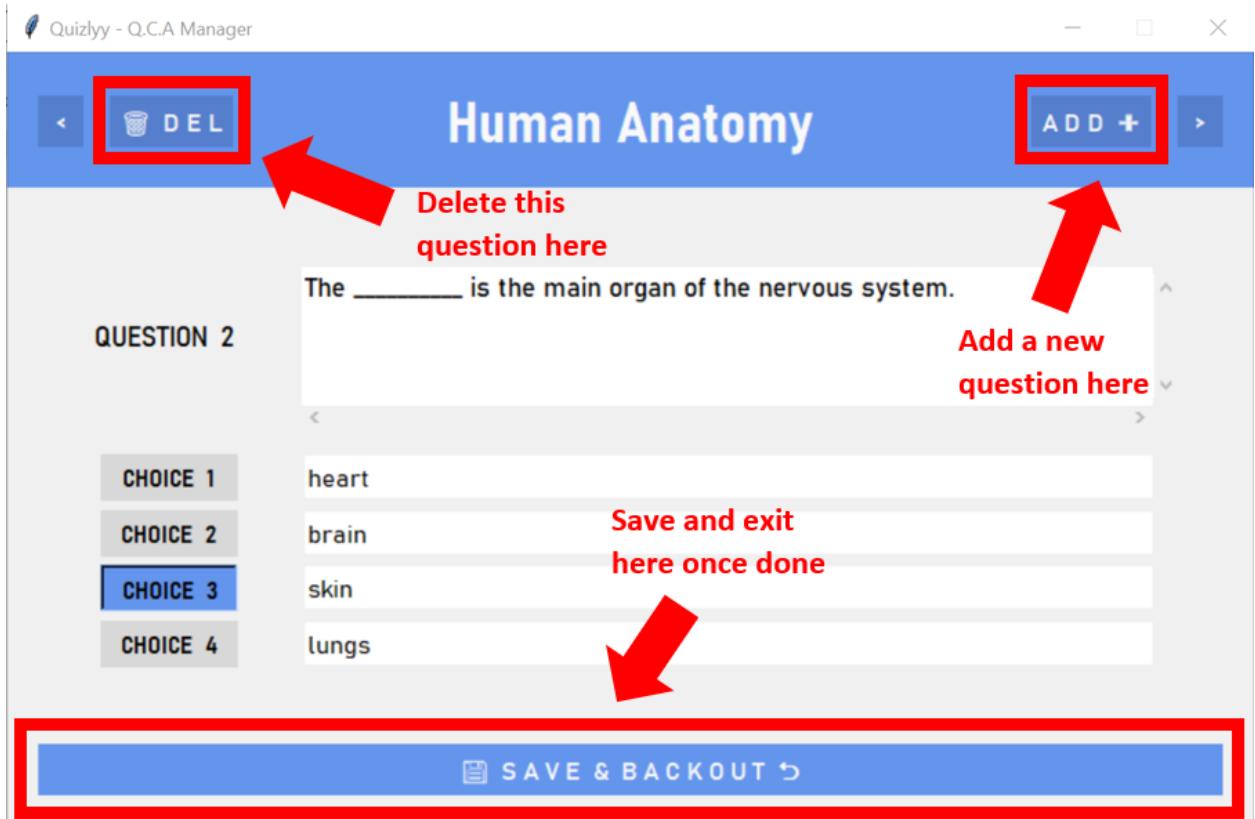


Figure 3.4.34: Quiz Q.C.A Manager Page

In the Quiz Q.C.A Manager Page, teachers can also delete that question by clicking the “DEL” button and add a new question in the quiz by clicking the “ADD”. After teachers have finished editing the quiz questions, they should click on the “SAVE & BACKOUT” button to return to the Teacher's Quiz Menu Page.

### c. Deleting Quiz

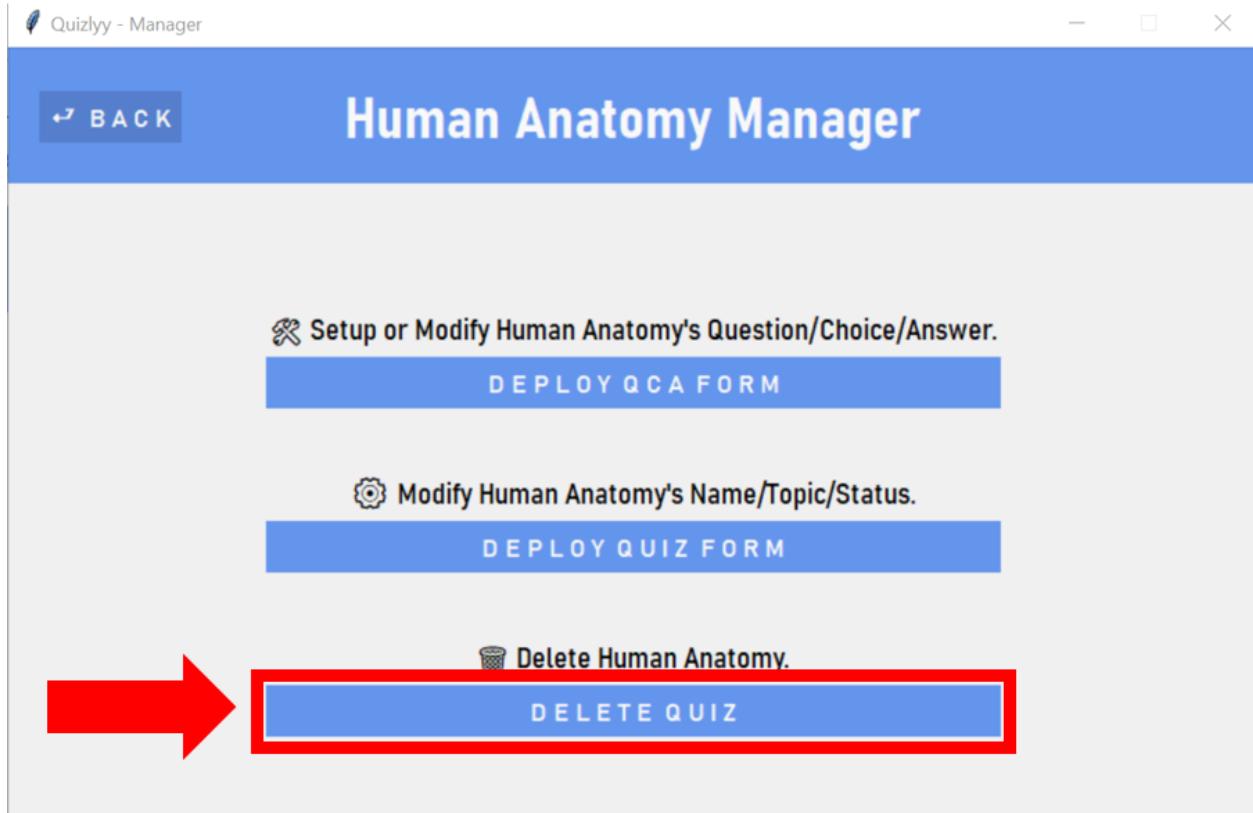


Figure 3.4.35: Quiz Manager Page

In this Quiz Manager Page, teachers can click the “DELETE QUIZ” button to delete the quiz.



Figure 3.4.36: Quiz Manager Page

After clicking on the “DELETE QUIZ” button, a confirmation message box will be displayed and teachers who wish to continue deleting should click “Yes”. They will be redirected back to the Teacher's Quiz Menu Page.

## 10. Viewing Quiz (Teacher)

The screenshot shows a web application window titled "Quizlyy Menu - Teacher". The main title is "Robert's Quizzes" and there is a "LOGOUT" button in the top right corner. Below the title is a table with three columns: "Quiz", "Status", and "Action".

Quiz	Status	Action
TOPIC: Physics NAME: Mechanics	Published	MANAGE VIEW
TOPIC: Programming NAME: Python	Published	MANAGE VIEW
TOPIC: Biology NAME: Human Anatomy	Published	MANAGE VIEW

A red arrow points to the "VIEW" button for the first quiz entry. At the bottom of the page is a blue "ADD QUIZ" button.

Figure 3.4.37: Teacher's Quiz Menu Page

Teachers can view a list of both drafted and published quizzes in the Teacher's Quiz Menu Page. They can also view the questions and answer choices in the quiz by clicking on the "VIEW" button in the Teacher's Quiz Menu Page.

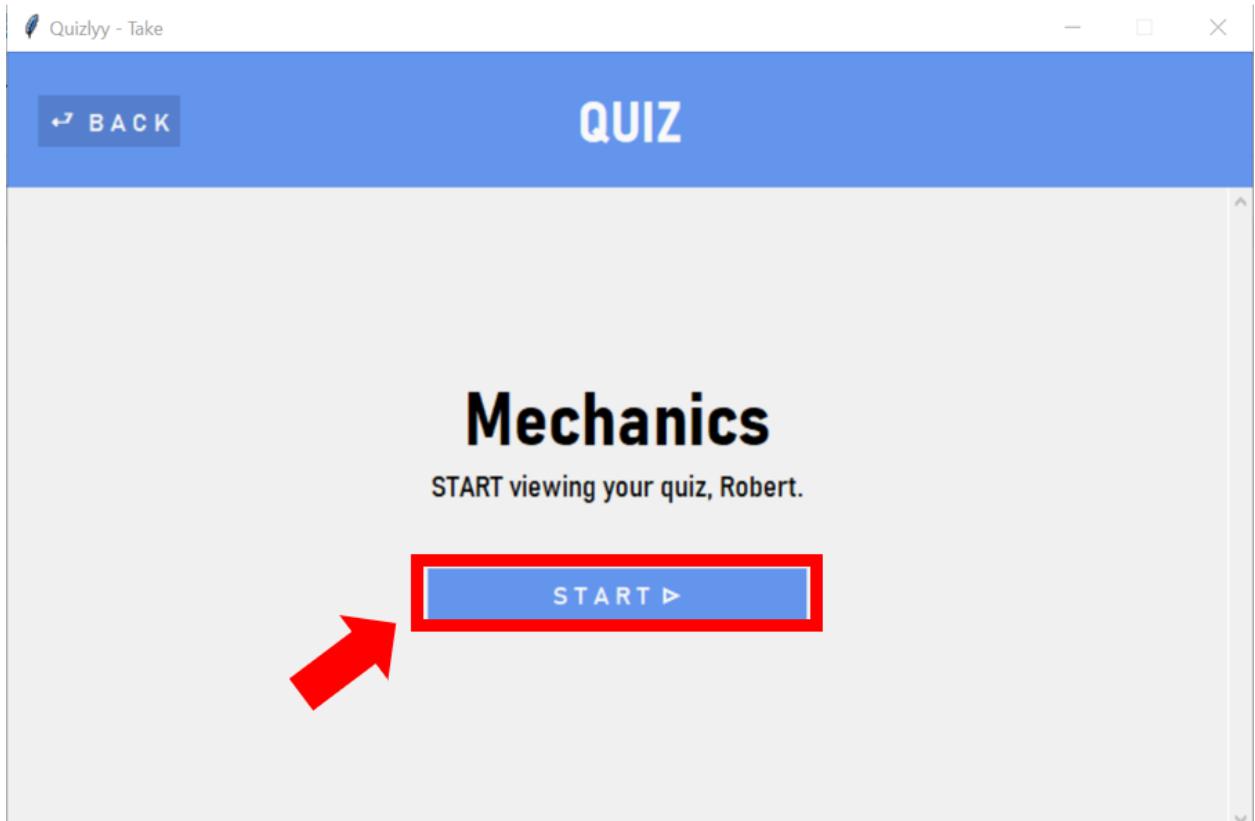


Figure 3.4.38: Quiz Start Page

After clicking on the “VIEW” button in the Teacher’s Quiz Menu Page, teachers should click on the “START” button in the Quiz Start Page to be able to view the questions and answer choices in the quiz.

The screenshot shows a quiz interface. At the top, there's a blue header bar with the Quizlyy logo, the title 'Quizlyy - Take', and window control buttons. Below the header, the word 'Mechanics' is centered in large white text. On the left, there's a 'BACK' button with a red arrow pointing to it from the left. On the right, there's an 'EDIT' button with a red arrow pointing to it from the right. A red box highlights the 'Edit' button. In the center, there's a question and four multiple-choice options. Red text above the question says 'Click here to return to Teacher's Quiz Menu'. Red text next to the 'Edit' button says 'Click here to edit quiz questions'.

Click here to return to Teacher's Quiz Menu

1. Laura throws a ball vertically. She notices it reaches a maximum height of 10 meters. What was the initial velocity of the ball?  
 $g = -9.8 \text{ m/s}^2$

- 14 m/s
- 22.2 m/s
- 29 m/s
- 196 m/s

Figure 3.4.39: Quiz View Page

Upon clicking the “START” button, teachers are able to view this Quiz View Page. Teachers are also able to edit the questions by clicking on the “EDIT” button and go back to the Teacher’s Quiz Menu by clicking on the “BACK” button.

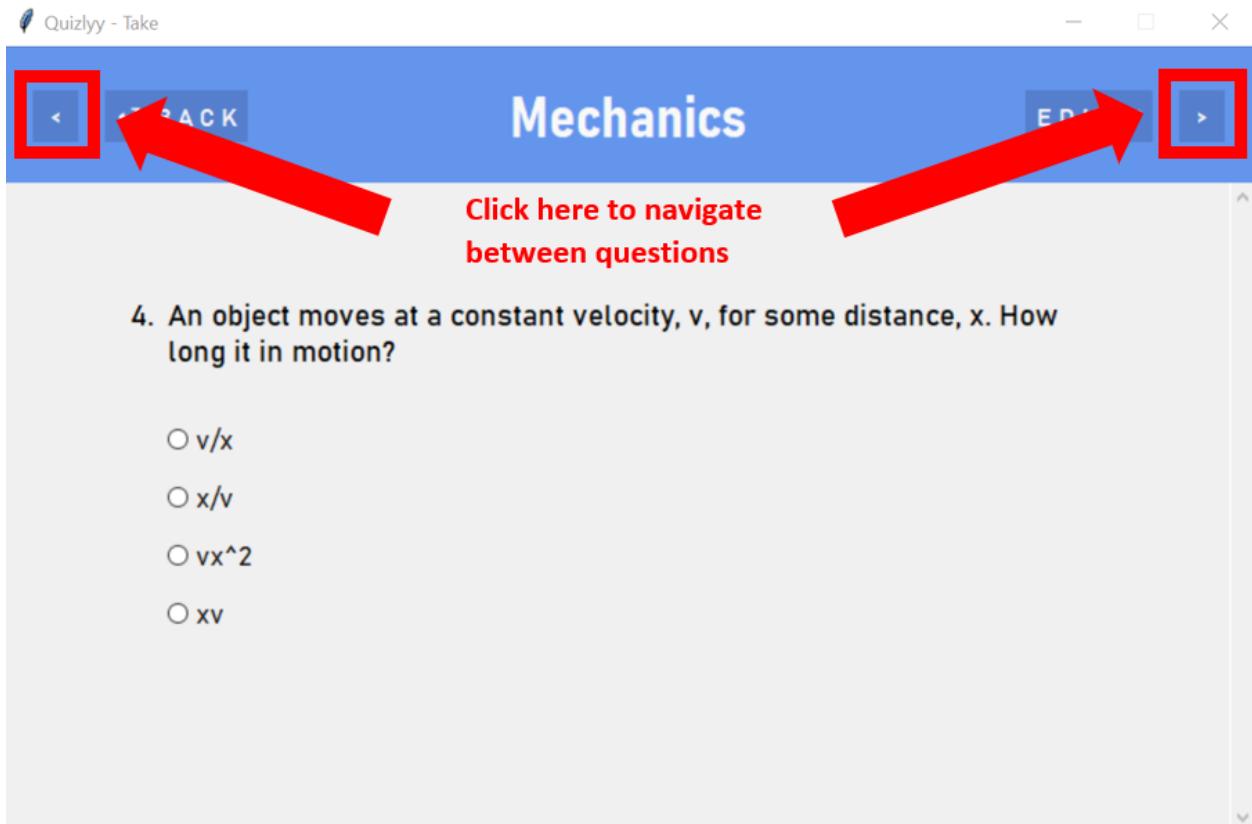


Figure 3.4.40: Quiz View Page

Teachers can also click on the “ $<$ ” to return to the previous question and “ $>$ ” button to navigate to the next question.

# **CHAPTER 4 : IMPLEMENTATION**

## **4.1 SYSTEM OVERVIEW**

---

Throughout the implementation phase of this project, much coding was done to form and combine different features of this program together. All of the coding work is done by using Python version 3.9.2 and the Tkinter library. The main reason we used python was because Python is designed with features to facilitate data analysis and visualization and we used the tkinter GUI because it's the standard GUI library for Python which comes together with Python version 3.9.2. We used JSON (JavaScript Object Notation) file format which is a built-in package from Python as the persistent storage of our program. The following are some of the snapshots of the codes used in this project to achieve the goal of this project.

## 4.2 CODE RESULT

---

### F1 (REGISTER NEW ACCOUNT)

```
374
375     def register_btn_clicked():
376         u_name = u_input.get()
377         p_word = p_input.get()
378         u_type = user_choice.get()
379
380         acc_dic = {'username': u_name, 'password': p_word, 'user type': u_type}
381         l = []
382
383         for info in acc_dic:
384             if acc_dic[info] == "":
385                 l.append(info)
386             s = ''
387             for word in l:
388                 if word == l[-1]:
389                     s += word + '.'
390                 elif word == l[-2]:
391                     s = s + word + ' and '
392                 else:
393                     s = s + word + ', '
394
395             # if there are fields that user haven't fill in
396             if l != []:
397                 mb.showinfo('Registration error', f'Please enter your {s}')
398
399         else:
400             # check whether username is unique or has been taken
401             with open("acc-info.json") as f:
402                 data = json.loads(f.read())
403                 result = False
404                 for item in data:
405                     if item["username"] == u_name:
406                         result = True
407
408             if result:
409                 mb.showinfo('Registration error', f'This username has been taken. Please enter another username.')
410             else:
411                 account_save('acc-info.json', acc_dic)
```

Figure 4.2.1: Registering Account

The figure above shows the Python code for the register function. The code begins with getting the input for username, password and user type and then updating them to a dictionary (*acc\_dic*). At line 396, it checks whether the user enters all the required details, else a message will be shown to inform the user on the field(s) that the user didn't fill in. At line 401, after the user enters all the required details, it checks for the existence of any duplicate or repeated username in the “*acc-info.json*” file before saving in the “*acc-info.json*” file.

## F2 (SAVE NEW QUIZ)

```
311
312     def backOut():
313         rootFrame.destroy()
314
315         TEACHER = username
316         tData = loadJSON(TEACHER_QUIZ_INFO_FILENAME)
317         quizDictList = tData[TEACHER]
318
319         tData_lst = []
320         for quizDict in quizDictList:
321             quizName = quizDict["Quiz Name"]
322             quizID = quizDict["ID"]
323             quizStatus = quizDict["D/P"]
324
325             sub_tData_lst = [f'{quizName}:{quizID}', quizStatus, ""]
326             tData_lst.append(sub_tData_lst)
327
328         DATA_LST = tData_lst
329         if DATA_LST == []:
330             DATA_LST = [[]]
331         mainPage.teacherPage(root, TEACHER, DATA_LST)
332
```

Figure 4.2.2: Save New Quiz

The above Python code shows the function “*backOut()*” to save new quizzes to its persistent storage(“*teacher-quiz-info.json*”) when the user (teacher) enters all the required details and proceeds to add a New Quiz. If *DATA\_LST* is empty, then it would add the New Quiz details into the list. Finally, the data from list (*DATA\_LST*) is used to update the quiz information on Teacher’s Quiz Menu Table and save New Quizzes to the JSON file as well.

### F3 (CREATE TABLE)

```
210     # createTable
211     for i in range(sRows):
212         for j in range(sColumns):
213
214             # Header Row
215             if i == 0 and j < sColumns-2:
216                 #createHeaderLabel
217                 self.lbl = tk.Label(
218                     self.tableFrame,
219                     text=self.modded_sData_lst[i][j],
220                     width=25,
221                     height=2,
222                     fg='white',
223                     bg='cornflower blue',
224                     font=HDR)
225
226                 self.lbl.grid(row=i, column=j, pady=(0,2), sticky="NSEW")
227
228             # Score Column Header
229             if i == 0 and j == sColumns-2:
230                 #createHeaderLabel
231                 self.lbl = tk.Label(
232                     self.tableFrame,
233                     text=self.modded_sData_lst[i][j],
234                     height=2,
235                     width=19,
236                     fg='white',
237                     bg='cornflower blue',
238                     font=HDR)
239
240                 self.lbl.grid(row=i, column=j, pady=(0,2), sticky="NSEW")
241
242             # Action Column Header
243             elif i == 0 and j == sColumns-1:
244                 #createHeaderLabel
245                 self.lbl = tk.Label(
246                     self.tableFrame,
247                     text=self.modded_sData_lst[i][j],
248                     height=2,
249                     width=20,
250                     fg='white',
251                     bg='cornflower blue',
252                     font=HDR)
253
254                 self.lbl.grid(row=i, column=j, pady=(0,2), sticky="NSEW")
```

Figure 4.2.3: Create Table

The figure above shows the Python code that creates a table dynamically according to the size of the data. ‘*sRows*’ and ‘*sColumns*’ are the number of data lists and the number of elements in the data list respectively. The if statements (example: if *i* == 0 and *j* < *sColumns*-2) serve the purpose to separate each row/column from one another for design purposes. Else, the whole table will look the same in terms of width, height, padding, font and so on.

#### F4 (QUESTION-CONFIGURATION)

```
294 #----- question-config -----#
295 def get_question_SPACE_var(self, event):
296     self.question_SPACE_var.set(self.quesQuestion)
297     self.question_SPACE.replace("1.0", tk.END, self.question_SPACE_var.get())
298
299 def create_question_TAG(self, n):
300     question_SPACE = ttk.Label(self.mainFrame, text=f'QUESTION {n+1}', font=TAG)
301     question_SPACE.grid(row=1, column=1, padx=(0,0), pady=(20,50), sticky="E")
302     return question_SPACE
303
304 def create_question_SPACE(self):
305     question_SPACE = tk.Text(self.mainFrame,
306                             font=QSPACE,
307                             wrap=tk.NONE,
308                             width=60,
309                             height=4,
310                             relief="flat",
311                             bd=0.5)
312
313     xscrollbar = ttk.Scrollbar(self.mainFrame, orient="horizontal", command=question_SPACE.xview)
314     yscrollbar = ttk.Scrollbar(self.mainFrame, orient="vertical", command=question_SPACE.yview)
315     question_SPACE.configure(xscrollcommand=xscrollbar.set, yscrollcommand=yscrollbar.set)
316
317     xscrollbar.grid(row=1, column=2, columnspan=2, padx=(41, 65), pady=(75,0), sticky="EW")
318     yscrollbar.grid(row=1, column=3, padx=(0, 48), pady=(5,35), sticky="NES")
319     question_SPACE.grid(row=1, column=2, columnspan=2, padx=(24,48), pady=(0,30), ipadx=1)
320
321     question_SPACE.bind('<Configure>', self.get_question_SPACE_var)
322     return question_SPACE
323 #-----
```

Figure 4.2.4: Question-Configuration

The figure above shows the Python code for configuring the question input in the Question Form. At line 299 is a function that creates a Label that shows the Question number beside the Question space. At line 304 is a function that creates a Text box to input a Question. It also creates horizontal and vertical scrollbars. At line 321 under the previous function, replaces content inside the Text to show the current question (whether it's empty or not) according to the JSON file.

## F5 (NAVIGATE QUESTION: NEXT BUTTON)

```
385      #----- button-config -----
386      ##### button-next #####
387      def nextQues(self):
388          if self.mode == "student":
389              self.save_student_guess()
390          self.quesIndex+=1
391          self.startQuiz()
392
393
394      def create_btn_next(self):
395          btn_next = tk.Button(self.apexFrame,
396                               text=' > ',
397                               font=BTN,
398                               fg=WHITEV1,
399                               bg=BLUEV2,
400                               activeforeground=WHITEV2,
401                               activebackground=BLUEV1,
402                               relief="flat",
403                               command=self.nextQues)
404
405          btn_next.grid(row=0, column=3, padx=(0,20), pady=(28,0), sticky="NE")
406          return btn_next
407      #####
```

Figure 4.2.5 Next Button

The figure above shows the Python code for the next button function. The next button is basically for the user (students) to move forward to the next question of the quiz they're attempting. From line 387 to 391, the code saves the answer of the current question and then moves to the next question by incrementing the question number counter. From line 394 to 406, it creates a button variable with the listed attributes values and places it on the screen.

## F5 (NAVIGATE QUESTION: PREVIOUS QUESTION BUTTON)

```
409 ##### button-prev #####
410 def prevQues(self):
411     if self.mode == "student":
412         self.save_student_guess()
413     self.quesIndex-=1
414     self.startQuiz()
415
416
417 def create_btn_prev(self):
418     btn_prev = tk.Button(self.apexFrame,
419                           text=' < ',
420                           font=BTN,
421                           fg=WHITEV1,
422                           bg=BLUEV2,
423                           activeforeground=WHITEV2,
424                           activebackground=BLUEV1,
425                           relief="flat",
426                           command=self.prevQues)
427
428     btn_prev.grid(row=0, column=1, padx=(20,0), pady=(28,0), sticky="NW")
429     return btn_prev
430 #####
```

**Figure 4.2.6 Previous Button**

The figure above shows the Python code for the previous button function. The previous button is basically for the user (students) to move backward to the previous question of the quiz they're attempting. From line 410 to 411, the code saves the answer of the current question and then moves to the previous question by decrementing the question number counter. From line 417 to 429, it creates a button variable with the listed attributes values and places it on the screen.

# CHAPTER 5 : CONCLUSION

---

The main purpose of this project is to make a paperless quiz management system to defeat the restrictions of the paper-based quiz systems and improve user interaction. To accomplish this project, we mainly used Python version 3.9.2, tkinter GUI library and JSON (JavaScript Object Notation) for the data format.

Firstly, several studies have been done to investigate the qualities and shortcomings of the current existing systems to acquire a superior comprehension of the problems faced by the current quiz systems. Then, the objectives of the project are laid out to distinguish the key features that the quiz management system should provide.

From that point onward, a proposed solution was drafted alongside identifying the strategy and tools for constructing the quiz management systems. Lastly, several designs were drawn to demonstrate the functions and database of the system. These phases have assisted with the initial requirements and design for the quiz management system.

The construction and full implementation of the system was done in the latter phase of the project. Among the other features, the question batch upload feature will help the teacher to create quizzes quickly. There are several other enhancements that can be added to the system in the future. For example, the system can be modified to become more informative by adding in visual elements such as charts and tables. The system can also be improvised by allowing teachers to set a timer for each quiz.

We have also learnt several lessons throughout the process of completing this project. First and foremost, we learned how to use the tkinter library which comes together with Python and how to work with JSON data including getting and passing data from a JSON file to python files. Besides learning new knowledge, we are also given the opportunity to improve our existing knowledge in Python programming. Furthermore, we are also exposed to proper ways to develop a Python program or application by analysing and designing before starting to program.

All in all, the project has achieved its goal of creating a paperless quiz management system. It has also created a good platform for further developments.

# CHAPTER 6 : REFERENCES

---

1. Punetha, A., Rohith, V. V. S., Chheda, R. M., & M.H.M, M. P. (2020, September 15). *Creating a JSON file with Tkinter Python (with example)* *Creating a JSON file with Tkinter Python (with example)*. IoTEDU.  
[https://iot4beginners.com/creating-a-json-file-with-tkinter-python-with-example/.](https://iot4beginners.com/creating-a-json-file-with-tkinter-python-with-example/)
2. srishtirajani. (2020, April 21). *Create Table Using Tkinter*. GeeksforGeeks.  
[https://www.geeksforgeeks.org/create-table-using-tkinter/.](https://www.geeksforgeeks.org/create-table-using-tkinter/)
3. Sun Bear 5, S. B. (2019, March 27). *How to position a progressbar widget in a Text widget in tkinter?* Stack Overflow.  
[https://stackoverflow.com/questions/55378737/how-to-position-a-progressbar-widget-in-a-text-widget-in-tkinter.](https://stackoverflow.com/questions/55378737/how-to-position-a-progressbar-widget-in-a-text-widget-in-tkinter)
4. Mokhtar Ibrahim Founder of LikeGeeks. I've been working as a Linux system administrator since 2010. I'm responsible for maintaining, M. E. (2020, May 27). *Python GUI examples (Tkinter Tutorial)*. Like [Geeks](#).  
[https://likegeeks.com/python-gui-examples-tkinter-tutorial/.](https://likegeeks.com/python-gui-examples-tkinter-tutorial/)
5. Salvatierra, J. (2020, February 13). *Scrollable Frames in Tkinter*. The Teclado Blog.  
[https://blog.teclado.com/tkinter-scrollable-frames/.](https://blog.teclado.com/tkinter-scrollable-frames/)
6. Choudhary, P. (2020, August 17). *Append to JSON file using Python [Updating a JSON file.]*. Python JSON Append to JSON file using Python [Updating a JSON file.] | Python.  
<https://cppsecrets.com/users/7815651141171101091031111191009757576410310997105108469911109/Python-JSON-Append-to-JSON-file-using-Python-Updating-a-JSON-file.php>