

# Development of an Online Question and Answer Management System using Flask

submitted by

Subhajeet Muhuri

Class Roll No: 002210503051

Exam Roll: MCA244050

Master of Computer Application

under the supervision of

Prof. (Dr.) Nibaran Das

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

2024

Faculty of Engineering and Technology  
Jadavpur University

**CERTIFICATE OF RECOMMENDATION**

**To Whom It May Concern**

I hereby recommend that the project report entitled “Development of an Online Question and Answer Management System using Flask” has been carried out by Subhajeet Muhuri (Reg. No.: 163643 of 2022-2023, Roll No.: 002210503051, Exam Roll: MCA244050), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2023-2024.

---

Prof. (Dr.) Nibaran Das  
Project Supervisor  
Dept. of Computer Science and  
Engineering  
Jadavpur University

---

Prof. (Dr.) Nandini Mukhopadhyay  
Head of the Department  
Dept. of Computer Science & Engineering  
Jadavpur University

---

Prof. Dipak Laha  
Dean, Faculty Council  
of Engineering & Technology  
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology  
Jadavpur University

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled “**Development of an Online Question and Answer Management System using Flask**” is a bonafide record of work carried out by SUBHAJEET MUHURI, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

---

Signature of Examiner 1

Date :

---

Signature of Examiner 2

Date :

Faculty of Engineering and Technology  
Jadavpur University

**DECLARATION OF COMPLIANCE WITH  
ACADEMIC ETHICS**

I hereby declare that this project report entitled "Development of an Online Question and Answer Management System using Flask" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

Subhajeet Muhuri

Registration No.: 163643 of 2022-23

Examination Roll No.: MCA244050

Thesis Title :- Development of an Online Question and Answer Management System using Flask

---

Signature with Date

# ACKNOWLEDGEMENT

With my most sincere and gratitude, I would like to thank Prof. (Dr.) Nibaran Das, Department of Computer Science & Engineering, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I needed them.

---

Subhajeet Muhuri

Class Roll No.: 002210503051

Examination Roll No.: MCA244050

Department of Computer Science & Engineering  
Jadavpur University

# Contents

---

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>1. Introduction</b>                             | <b>5</b>  |
| 1.1. Existing available systems                    | 6         |
| 1.2. Limitations of the available systems          | 7         |
| 1.3. Objective                                     | 7         |
| <b>2. Existing Systems</b>                         | <b>9</b>  |
| <b>3. Present work</b>                             | <b>10</b> |
| 3.1. Front-End Technologies                        | 10        |
| 3.1.1. HTML (Hypertext Markup Language)            | 10        |
| 3.1.2. CSS (Cascading Style Sheets)                | 10        |
| 3.1.3. JavaScript                                  | 11        |
| 3.1.4. Bootstrap                                   | 11        |
| 3.2. Back-End Technologies                         | 12        |
| 3.2.1. Python                                      | 12        |
| 3.2.2. Flask                                       | 12        |
| 3.2.3. Jinja                                       | 13        |
| 3.3. Database                                      | 13        |
| 3.3.1. MySQL                                       | 13        |
| 3.4. Question-Answer Generation                    | 13        |
| 3.5. Enhanced Entity Relationship (EER) Diagram    | 14        |
| 3.5.1. Cardinalities between two Entities          | 16        |
| 3.6. Database Schema (Simplified from EER Diagram) | 16        |
| 3.7. Functional Requirements                       | 18        |
| 3.7.1 User Registration and Authentication         | 18        |
| 3.7.2 Role-Based Dashboards                        | 18        |
| 3.7.3 Test Creation and Management for Examiners   | 18        |
| 3.7.4 Question Management                          | 19        |
| 3.7.5 Test Taking for Students                     | 19        |

|           |                                                      |           |
|-----------|------------------------------------------------------|-----------|
| 3.7.6     | Results Management                                   | 19        |
| 3.7.7     | Test and Question Visibility                         | 19        |
| 3.7.8     | Feedback                                             | 20        |
| 3.8.      | Non-Functional Requirements                          | 20        |
| 3.8.1     | Performance                                          | 20        |
| 3.8.2     | Usability                                            | 20        |
| 3.8.3     | Security                                             | 20        |
| 3.8.4     | Compatibility                                        | 20        |
| 3.9.      | Structure and Flow of the System                     | 21        |
| <b>4.</b> | <b>Result and Analysis</b>                           | <b>22</b> |
| 4.1.      | Demonstration of the proposed system's various steps | 22        |
| 4.2.      | Evaluation of Students' Attempted Tests              | 31        |
| 4.3.      | Analysis of Question-Answer Generation               | 33        |
| <b>5.</b> | <b>Conclusion and Future Scope</b>                   | <b>35</b> |
|           | References                                           | 36        |

## List of Figures

---

|      |                                            |    |
|------|--------------------------------------------|----|
| 1.1  | Simple overview of this Project            | 6  |
| 3.1  | Enhanced Entity Relationship (EER) Diagram | 15 |
| 3.2  | Structure and Flow of the Website          | 21 |
| 4.2  | Login Page                                 | 22 |
| 4.3  | Registration Page                          | 23 |
| 4.4  | Examiner Dashboard Page                    | 24 |
| 4.5  | Create Test Page                           | 24 |
| 4.6  | View Tests Page                            | 25 |
| 4.7  | Edit Test Page                             | 26 |
| 4.8  | Add Question Page                          | 26 |
| 4.9  | View Tests Attempted by Students Page      | 27 |
| 4.10 | View Students Who Attempted Tests Page     | 27 |
| 4.11 | Student Test Performance Page              | 28 |
| 4.12 | Student Dashboard Page                     | 28 |
| 4.13 | Tests to Attempt Page                      | 29 |
| 4.14 | Test Attempt Page                          | 29 |
| 4.15 | Practice Test Page                         | 30 |
| 4.16 | Tests Attempted Page                       | 30 |
| 4.17 | Student Attempting Test                    | 31 |
| 4.18 | Student Submitting Test                    | 32 |
| 4.19 | Student Viewing Evaluated Test Results     | 32 |
| 4.20 | Student Viewing Detailed Results           | 33 |

## List of Tables

---

|     |                                               |    |
|-----|-----------------------------------------------|----|
| 3.1 | User Database Table                           | 16 |
| 3.2 | Test Database Table                           | 17 |
| 3.3 | Question Database Table                       | 17 |
| 3.4 | Student's Test Performance Database Table     | 17 |
| 3.5 | Student's Question Performance Database Table | 18 |

# Chapter 1: Introduction

In today's digital age, education is evolving rapidly with technology becoming integral to learning and assessment. This project "Development of an Online Question and Answer Management System using Flask" aims to create a user-friendly platform for managing tests and quizzes online by leveraging Flask, a powerful Python web framework.

Upon visiting this online platform for the first time, users need to navigate to the registration page to sign up either as an examiner or a student. Examiners can create tests, view all tests they have created, and analyze student results. While creating a test, they can upload questions through a CSV file or manually enter them. They can also specify if it's a practice test and choose whether the test is public or private. The examiner can then edit or delete tests and questions.

On the other hand, students can view and attempt both practice and actual tests created by the examiners. After completing a test, they can view their results. In their results, they can see the number of correct, incorrect, and unattempted questions. For self-assessment purposes, practice tests offer the additional feature of showing correct answers.

Creating an online question-answer management system also requires a wide and accurate set of questions and answers. We used publicly available free NCERT Textbooks and IIT Bombay's Computer Masti Series as sources for generating questions and answers related to Computer Science. We used powerful AI technologies like ChatGPT's GPT-4o Model and Claude AI to simplify the process of creating questions and answers from these resources.

While developing this project, we have aimed to create a user-friendly interface. However, as students, we understand that there is always room for improvement. We have focused on the core functionalities, ensuring that the system works efficiently and meets the primary needs of its users. Future iterations of the project could include additional features such as more varied question types, enhanced security measures, and more detailed analytics for both students and examiners.

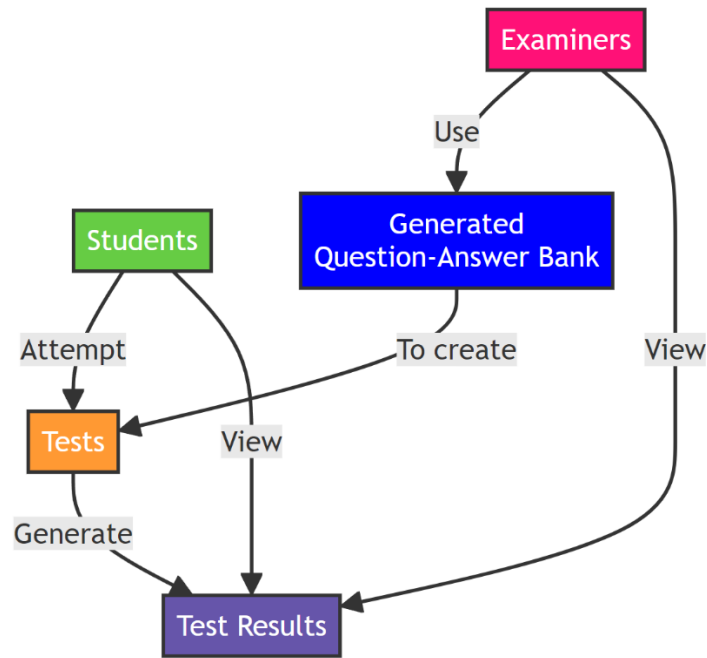


Figure 1.1: Simple overview of this Project

## 1.1 Existing available systems

The current state of question-and-answer management systems includes both contemporary digital platforms and conventional techniques. Many educational institutions still use traditional methods like paper-based testing and in-person assessments.

Scalable and convenient online evaluation solutions are available on modern digital platforms. Popular technologies that make online tests, quizzes, and test management easier to use include various dedicated platforms [1, 2, 3, 4]. These technologies facilitate automatic grading, prompt feedback, and simple test delivery to a wide student body. They frequently interface with other teaching resources as well, which improves the quality of learning overall.

However, while these platforms take care of the basic needs, they frequently lack capabilities that are specific to roles, such as examiners and students. Also, these platforms' user interfaces sometimes lack intuitiveness, which makes it difficult for users to navigate and make full use of all the functions.

## 1.2 Limitations of the available systems

Despite the gains provided by previous technology, significant limits remain. Traditional evaluation systems are frequently limited, providing little flexibility for examiners and pupils. This rigidity can inhibit the development of new question types and the capacity to customize tests to meet specific educational demands.

The user experience on different platforms and devices may differ significantly. Accessibility and usability may be hampered by inconsistent interfaces and navigational problems, especially for students and examiners with limited technology experience.

Few options for maintaining and creating tests are available for customization in many of the current systems. Examiner flexibility to adjust tests to each person's requirements and learning objectives may be limited as a result.

## 1.3 Objective

The primary objective of this project is to create a user-friendly, efficient platform that simplifies the process of creating, managing, and taking tests online for both examiners and students. The main objectives of our project are as follows:

### **Simplify the Creation and Management of Tests:**

- Provide examiners with an interface to create tests manually or by uploading CSV files.
- Allow examiners to manage tests easily by toggling practice and public statuses.

### **Enhance Student Accessibility:**

- Provide students with a platform to easily access tests.
- Ensure students can practice with designated practice tests and take actual tests conveniently.

### **Automate Scoring and Feedback:**

- Automatically record student responses and calculate scores to provide immediate feedback.
- Enable examiners to view detailed student results, including correct, incorrect, and not attempted answers.

## **User Roles and Permissions:**

- Implement distinct roles for examiners and students to ensure proper access and functionalities.

# Chapter 2: Existing Systems

The existing landscape of question-and-answer management systems comprises both traditional processes and cutting-edge digital platforms, each with unique approaches and strategies. While simple and familiar, traditional systems like manual grading and in-person exams take a long time and are prone to human errors too.

Modern digital platforms have developed a variety of automated ways to help manage queries and responses. These approaches usually employ various technologies to create scalable, effective, and interactive solutions. For the development of an online question-and-answer management system, we employed several key technologies like HTML, CSS, JavaScript, Flask, and MySQL.

Many organizations and platforms offer online assessment solutions to make the evaluation process easier in educational environments. These include:

- **Google Forms [1] and Microsoft Forms [2]:** These are widely used for creating simple online quizzes and surveys. They offer basic functionalities but lack advanced features for detailed test management and analytics.
- **Moodle [3]:** It is a popular Learning Management System (LMS) that offers extensive features for course management, including quizzes and assessments. It supports automated grading, feedback, and integration with other educational tools.
- **Kahoot! [4]:** It is a platform designed for interactive learning and assessments. It engages students with game-like features and real-time feedback but may not cater to more formal testing requirements.
- **Custom-built Solutions:** Many educational institutions and organizations create their customized platforms to fit their specific requirements. These solutions can be customized to provide additional reporting capabilities, security measures, and interaction with current systems.

# Chapter 3: Present Work

## 3.1 Front-End Technologies

### 3.1.1 HTML (Hypertext Markup Language)

HyperText Markup Language, or HTML, is the standard markup language for creating web documents that can be viewed in a web browser. Links between web pages, whether they are on the same website or separate websites, are referred to as hypertext. Links are one of the most crucial components of the Web.

Web browsers convert HTML data, whether from a web server or local storage, into multimedia web pages. HTML is intended to reflect the meaningful structure of a webpage.

Components used in the construction of web pages must have HTML elements. It enables us to quickly add forms and other interactive elements, such as pictures, to a website. Assigning tags to different elements, such as headings, paragraphs, lists, links, and quotations, facilitates document structure. Angle bracket tags are used to identify these HTML elements. While `<p>` and `</p>` provide textual information, the elements contribute content to a page. These HTML tags are not shown by web browsers. Rather, they represent the appearance of the page.

In our project, we used HTML to create the web pages for our project. The forms, like the registration form with email, name, and password fields, and the login page, all are created with HTML. HTML tags helped us in creating the input fields, buttons, links, tables, and all the other elements on the pages.

### 3.1.2 CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) is a language used to style and display markup-based documents such as HTML or XML. It is comparable to HTML and JavaScript, two of the Web's essential technologies.

We can distinguish text from presentation with CSS, which also lets us control fonts, colors, and layout. The advantages of this separation include improved accessibility as well as increased flexibility and control over the look of web material. Multiple web pages can share the same styling by using separate .css files. This reduces the amount of repetition and complexity in the HTML content.

Furthermore, pages that use a single .css file can have faster load times when that file is cached. It's important to keep formatting and content separate because it allows the same page to be displayed in different ways depending on the device. For example, it can be read aloud by a screen reader, printed on paper, etc. CSS provides instructions for alternative layouts when the content is viewed on a mobile device.

In our project, CSS helped us make the rounded checkboxes with check marks for toggling the practice and public statuses of an examiner's created tests. It also helped in the vertical alignment of the edit and delete buttons in the middle of the action column in the tables.

### **3.1.3 JavaScript**

JavaScript was initially created to bring interactivity to web pages. It allows scripts, written in this language, to run when a web page loads. These scripts are written in basic text format and do not need any additional configuration or compilation.

Nowadays, JavaScript can run on web browsers, servers, and almost any device with a JavaScript engine.

JavaScript virtual machine refers to one of the embedded engines in a browser. Browsers use different engines with various codenames; for example, Firefox uses SpiderMonkey, while Chrome, Opera, and Edge use V8.

In our project, when we click the "Show Password" checkbox in the login and register pages, the password and confirm password fields toggle between showing and hiding the password. That's done using JavaScript. It is also used to show the correct answer in practice tests and show the chosen CSV question file's filename in the examiner's create test page.

### **3.1.4 Bootstrap**

Bootstrap is a toolkit consisting of HTML, CSS, and JavaScript which is used for developing web pages in a much simpler way. It focuses on using pre-built layouts, colors, and size options. Once Bootstrap is installed, it applies basic styles to all HTML components to ensure a consistent appearance across different browsers. Additionally, we can modify the website's appearance using the CSS classes provided with Bootstrap, allowing us to create tables with various colors, large page headings, highlighted text, and more.

Bootstrap has several JavaScript components that do not require additional frameworks, such as jQuery. These add aspects to the user interface, such as progress bars, dialog boxes, tooltips, navigation dropdowns, and carousels.

These functionalities also enhance current interface elements, such as autocomplete for input fields.

Our overall frontend UI is made almost entirely from a free Neumorphism-inspired user interface kit made by themesberg.com [5]. This UI kit's template pages are built with the Bootstrap Framework.

## **3.2 Back-End Technologies**

### **3.2.1 Python**

Python is a high-level programming language that prioritizes code readability through indentation. It supports various programming paradigms, including structured, object-oriented, and functional programming. It is a garbage-collected language and supports dynamic typing. Python's standard library is very extensive and versatile.

Python is consistently ranked among the most popular programming languages worldwide. Its simplicity and readability make it an excellent choice for beginners. Many educational institutions use Python to teach programming and computer science fundamentals. Python is used in a wide range of applications from web development (with frameworks like Django and Flask) to scientific computing (with libraries like NumPy and SciPy), data analysis (with pandas), artificial intelligence, and machine learning (with libraries like TensorFlow, PyTorch, and scikit-learn).

Python is the backbone of our entire project. It is the programming language we have used to write all our backend logic.

### **3.2.2 Flask**

We used Flask, a web framework written in Python, to build our application. It is called a micro-framework as it gives us just the essentials to get started with web development.

Flask handles routing for us, which means it takes care of mapping URLs to functions in our code. When we go to `/examiner/dashboard`, Flask knows which function to call to handle that request. Flask also makes it easy to handle different types of HTTP requests like GET and POST. We just define routes and methods, and Flask takes care of the rest.

### 3.2.3 Jinja

Jinja is the templating engine that comes with Flask. It allows us to create HTML templates that can dynamically display data from our backend. With Jinja, we can pass data from our Flask routes to our HTML templates. For example, when displaying the list of tests, we pass the test data to the template and Jinja takes care of rendering it in the right format. Jinja also supports control structures like loops and conditionals, which we use to render lists of data or show/hide content on certain conditions.

## 3.3 Database

### 3.3.1 MySQL

MySQL is a relational database management system (RDBMS). It helps us organize and manage all the information our website needs to work. This includes everything from user information (like their names, emails, and passwords) to test questions, student results, etc.

MySQL uses tables to store data in a structured way. Each table has rows and columns. In our project, we have a user table that stores all the user details, another table for tests that stores information about each test, and so on.

MySQL can handle relationships between different pieces of data properly and efficiently. In our project, we have a one-to-many relationship between examiners and tests, that is, one examiner can create many tests. MySQL helps us define these relationships using foreign keys, which ensures data integrity and makes it easier to query related data.

With MySQL, we can write SQL (Structured Query Language) queries to retrieve the needed data. In our project, when an examiner wants to see all the tests they've created, we can write a query to fetch this data from the database.

## 3.4 Question-Answer Generation

Creating an online question-and-answer management system requires the generation of questions and answers. Precise and diverse questions and their respective correct answers improve students' learning and assessment experiences. Here, we go into detail about the content generation strategies used.

To provide a wide range of questions and answers for our project, we used publicly accessible educational resources. Specifically, we used the NCERT textbooks and IIT Bombay's Computer Masti series. These resources offer thorough coverage of computer science-related subjects, guaranteeing that the information is accurate and of the highest quality.

We used advanced AI tools such as ChatGPT's GPT-4o model and Claude AI to simplify the process of collecting questions and answers from these resources. ChatGPT's free GPT-3.5 model does not permit file uploads, which is a limitation. However, the recently introduced GPT-4o model allows users to upload files, which aids in the automated extraction of material. It is made available to users for free up to a certain limit. Similarly, Claude AI's free file upload feature also allowed us to quickly process and generate questions and answers from the materials provided.

As a result, we were able to generate a significant volume of questions and answers systematically, covering a wide range of difficulty levels and topics. This method not only reduced time but also ensured that the questions came from trustworthy sources

### **3.5 Enhanced Entity Relationship (EER) Diagram**

Entity Relationship (ER) diagrams are a crucial part of database design, serving as a visual representation of the database's structure. An ER diagram is a blueprint of a database that defines how data is interconnected and how entities relate to each other within the system. The primary components of an ER model are entity sets and relationship sets.

Entity sets are collections of similar entities, which can be thought of as tables in a database. Each entity within an entity set has attributes, which are the properties or characteristics of that entity. Relationships among these entities are depicted in the ER diagram, illustrating how tables relate to one another.

Enhanced Entity Relationship (EER) diagrams extend the traditional ER model to capture more complex requirements and provide a more detailed and accurate representation of data properties and constraints. EER diagrams include all the modeling concepts of ER diagrams but introduce additional features such as subclasses and superclasses, specialization and generalization, category or union type, and attribute and relationship inheritance.

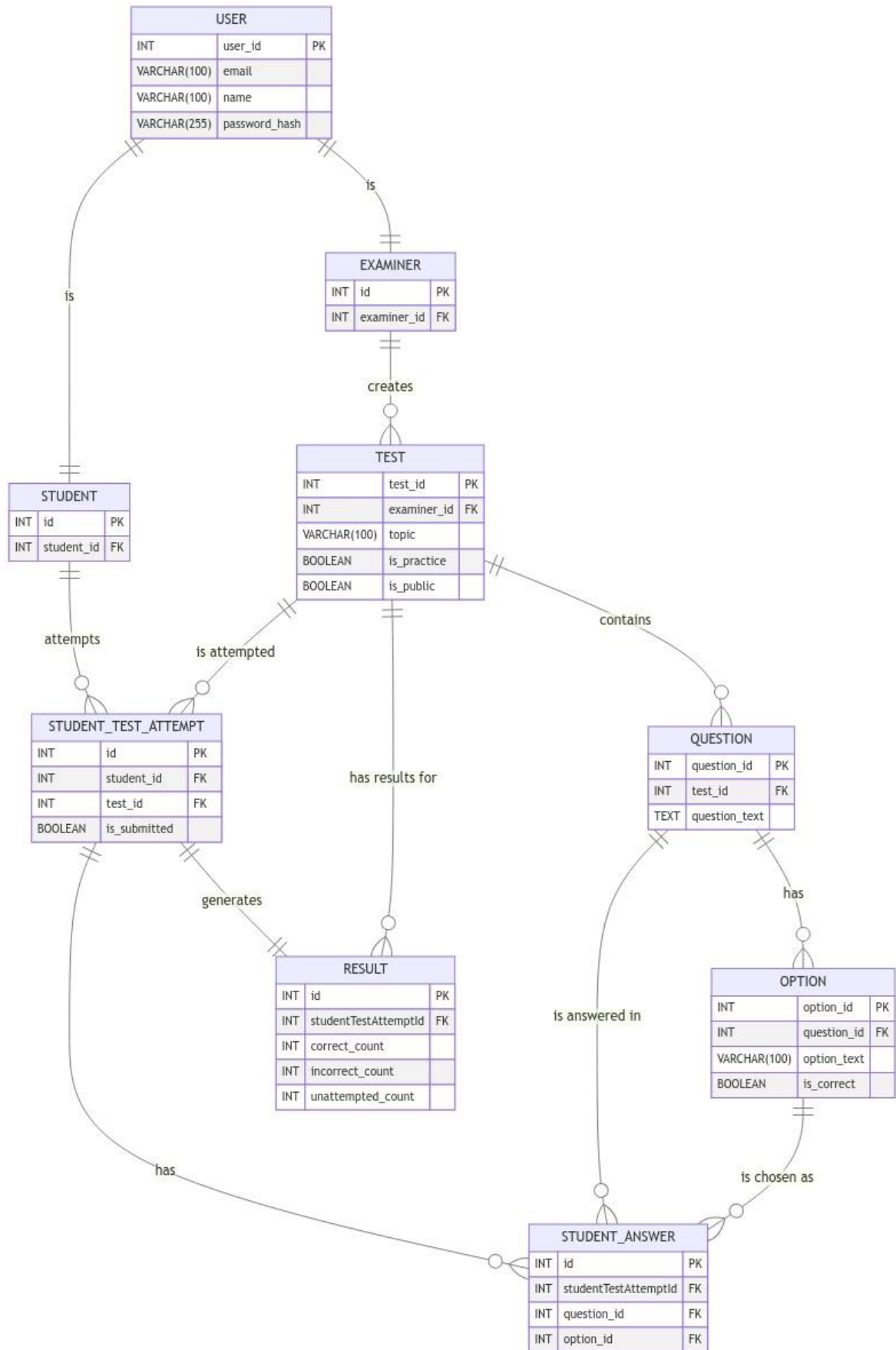


Figure 3.1: Enhanced Entity Relationship (EER) Diagram

### 3.5.1 Cardinalities between two Entities

- User to Examiner: One-to-One
- User to Student: One-to-One
- Examiner to Test: One-to-Many
- Test to Question: One-to-Many
- Question to Option: One-to-Many
- Student to Student\_Test\_Attempt: One-to-Many
- Test to Student\_Test\_Attempt: One-to-Many
- Student\_Test\_Attempt to Student\_Answer: One-to-Many
- Question to Student\_Answer: One-to-Many
- Option to Student\_Answer: One-to-Many
- Student\_Test\_Attempt to Result: One-to-One
- Student to Result: One-to-Many
- Test to Result: One-to-Many

## 3.6 Database Schema (Simplified from EER Diagram)

Database schema is the logical framework that defines the organization of data in a database. It includes tables, fields, relationships, views, indexes, and other elements. It is crucial for ensuring data integrity and efficiency in database operations.

Derived from the previous Enhanced Entity Relationship Diagram, the database tables used in this project are as follows:

Table “user”:

- Stores information about all users (students and examiners).

| Field Name    | Data Type                   | Key         |
|---------------|-----------------------------|-------------|
| user_id       | INT                         | PRIMARY KEY |
| email         | VARCHAR(100)                | -           |
| name          | VARCHAR(100)                | -           |
| password_hash | VARCHAR(255)                | -           |
| role          | ENUM('student', 'examiner') | -           |

Table 3.1: User Database Table

### Table “test”:

- Stores information about tests created by examiners.

| Field Name  | Data Type   | Key                                      |
|-------------|-------------|------------------------------------------|
| test_id     | INT         | PRIMARY KEY, FOREIGN KEY → user(user_id) |
| user_id     | INT         | -                                        |
| topic       | VARCHAR(30) | -                                        |
| is_practice | TINYINT     | -                                        |
| is_public   | TINYINT     | -                                        |

Table 3.2: Test Database Table

### Table “question”:

- Stores questions, their respective options, and correct options for each test.

| Field Name     | Data Type    | Key                         |
|----------------|--------------|-----------------------------|
| question_id    | INT          | PRIMARY KEY                 |
| test_id        | INT          | FOREIGN KEY → test(test_id) |
| question_text  | TEXT         | -                           |
| option_1       | VARCHAR(100) | -                           |
| option_2       | VARCHAR(100) | -                           |
| option_3       | VARCHAR(100) | -                           |
| option_4       | VARCHAR(100) | -                           |
| correct_option | TINYINT      | -                           |

Table 3.3: Question Database Table

### Table “student\_test\_performance”:

- Tracks which students have submitted which tests and their performance.

| Field Name        | Data Type | Key                                      |
|-------------------|-----------|------------------------------------------|
| user_id           | INT       | PRIMARY KEY, FOREIGN KEY → user(user_id) |
| test_id           | INT       | PRIMARY KEY, FOREIGN KEY → test(test_id) |
| is_submitted      | TINYINT   | -                                        |
| correct_count     | INT       | -                                        |
| incorrect_count   | INT       | -                                        |
| unattempted_count | INT       |                                          |

Table 3.4: Student’s Test Performance Database Table

## Table “student\_question\_performance”:

- Stores the answers given by students for each question in a test.

| Field Name    | Data Type | Key                                              |
|---------------|-----------|--------------------------------------------------|
| user_id       | INT       | PRIMARY KEY, FOREIGN KEY → user(user_id)         |
| test_id       | INT       | PRIMARY KEY, FOREIGN KEY → test(test_id)         |
| question_id   | INT       | PRIMARY KEY, FOREIGN KEY → question(question_id) |
| chosen_option | TINYINT   | -                                                |

Table 3.5: Student’s Question Performance Database Table

## 3.7 Functional Requirements

### 3.7.1 User Registration and Authentication

- The system must allow new users to register with their email, name, password, and role (either Student or Examiner).
- Registered users must be able to log in using their email and password.
- The system should provide a password visibility toggle for convenience during registration and login.

### 3.7.2 Role-Based Dashboards

- After logging in, users should be redirected to their respective dashboards based on their role.
- Examiners should see options for creating tests, viewing tests, and viewing student results on their dashboards.
- Students should see options for giving tests, practicing tests, and viewing their results.

### 3.7.3 Test Creation and Management for Examiners

- Examiners must be able to create new tests by entering a topic name and optionally uploading a CSV file with questions.
- The system should provide a downloadable CSV template for examiners to format their questions correctly.

- Examiners should have options to mark a test as a practice test or make it public.
- Created tests should be editable and deletable. Examiners must be able to add, update, or delete questions within a test.

### **3.7.4 Question Management**

- Examiners should be able to add questions to a test, specifying the question text, four options, and the correct answer.
- The system should allow editing and deleting of individual questions within a test.

### **3.7.5 Test Taking for Students**

- Students should be able to view and attempt tests that are marked as public and not practice tests.
- While attempting a test, students should see the questions with four options and select their answers.
- For practice tests, students should have the option to see the correct answers immediately after attempting each question.

### **3.7.6 Results Management**

- The system must store and display the results of tests attempted by students.
- Examiners should be able to view the results of tests, seeing how many questions were correct, incorrect, or not attempted by each student.
- Students should also be able to view their results, like the number of correct, incorrect answers and unattempted questions. The correct answers and their chosen options should also be available to students.

### **3.7.7 Test and Question Visibility**

- Examiners should be able to toggle the visibility of tests (public/private) and the practice status of tests.
- The system should accurately reflect these changes in the student's view of available tests.

### **3.7.8 Feedback**

- The system should provide meaningful feedback messages for actions like successful registration, login failures, test submissions, etc.

## **3.8 Non-Functional Requirements**

### **3.8.1 Performance**

- The system should load and respond quickly. Page load time should be less to ensure a smooth user experience.

### **3.8.2 Usability**

- The interface should be intuitive and user-friendly, allowing users to navigate without extensive instructions.

### **3.8.3 Security**

- User data, especially sensitive information like passwords, should be encrypted and stored securely.

### **3.8.4 Compatibility**

- The system should be compatible with major web browsers like Chrome, Firefox, Safari, and Edge.

### 3.9 Structure and Flow of the System

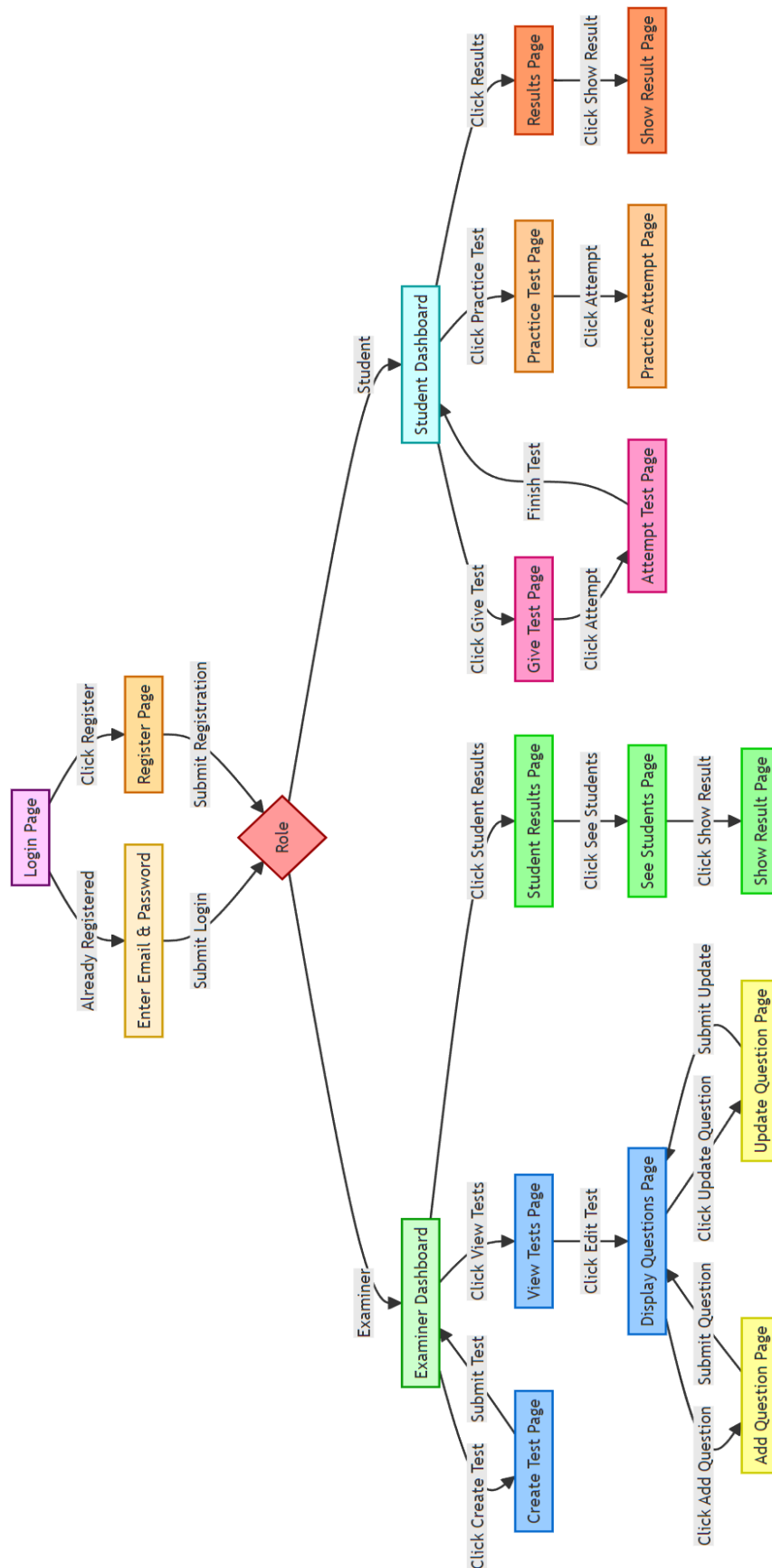


Figure 3.2: Structure and Flow of the System

# Chapter 4: Result and Analysis

After outlining the required methodologies and evaluating the necessary technologies, we implemented them systematically to develop our working proposed system.

## 4.1 Demonstration of the proposed system's various steps

Initially, we are greeted with the login page [Figure 4.2].

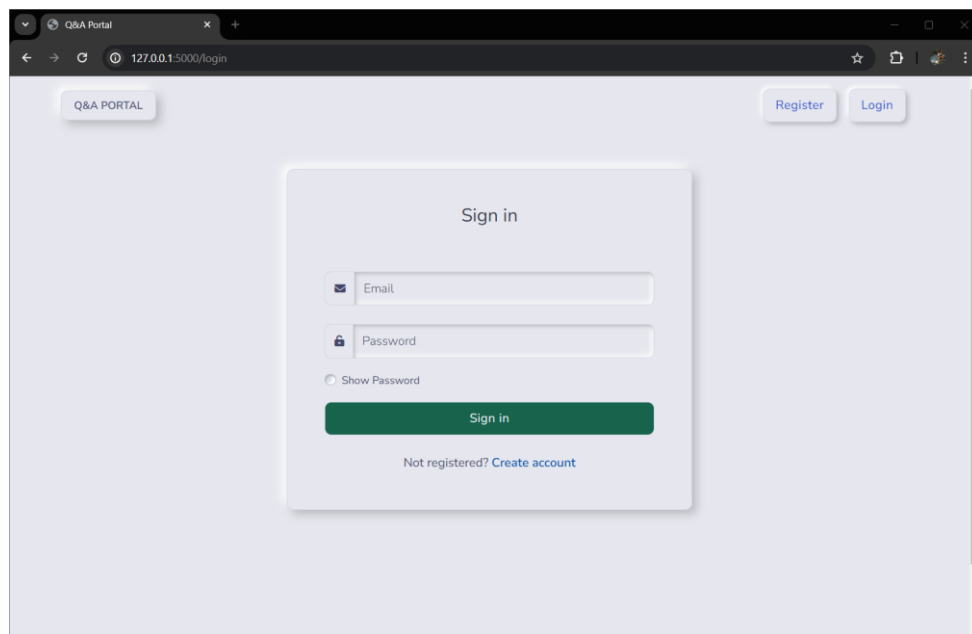


Figure 4.2: Login Page

If we do not have an account yet, we can click on the "Register" button located on the top navbar. We will be taken to the registration page [Figure 4.3].

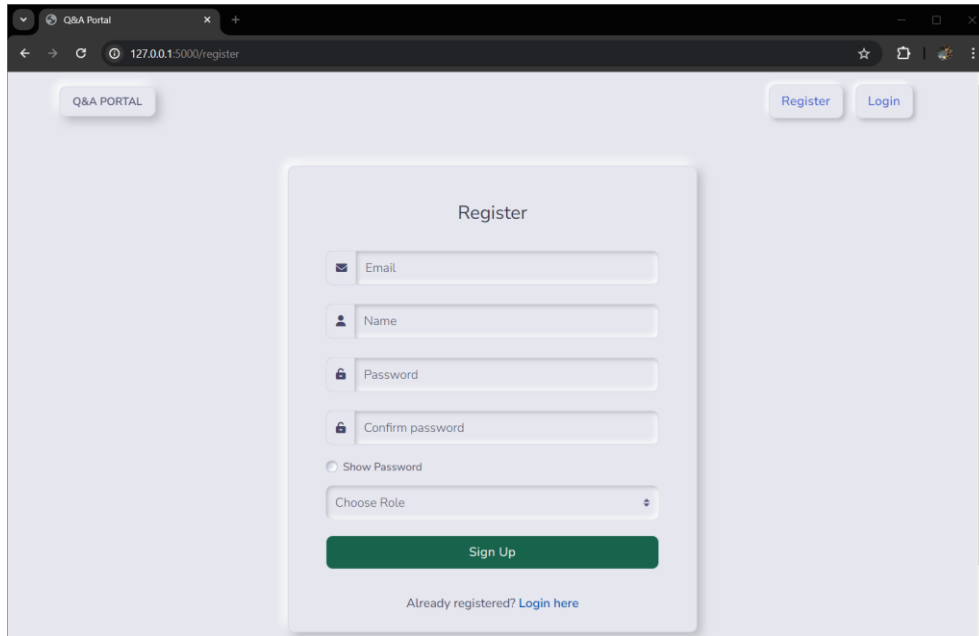


Figure 4.3: Registration Page

Now we need to fill in the following details in the Registration Page [Figure 4.3]: Email, Name, Password, Confirm Password.

We can check the "Show Password" checkbox to see the passwords we are typing. Next, we need to choose our role from the dropdown menu. The available roles are "Student" and "Examiner." For now, let's choose the Examiner role.

After filling in all the information, let's click on the "Sign Up" button. We will be redirected to the login page.

Now, we will enter the previously registered email and password to log in. We will now be taken to the Examiner Dashboard [Figure 4.4].

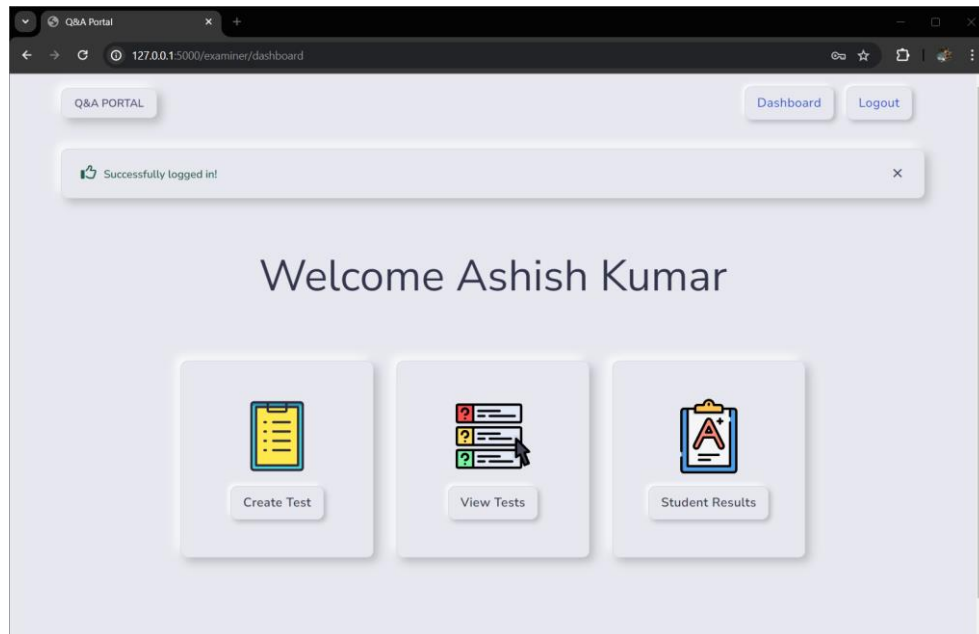


Figure 4.4: Examiner Dashboard Page

As an examiner, we will see the Examiner Dashboard page [Figure 4.4] upon logging in. The dashboard has three main buttons: Create Test, View Tests, and Student Results.

Let's click on the "Create Test" button.

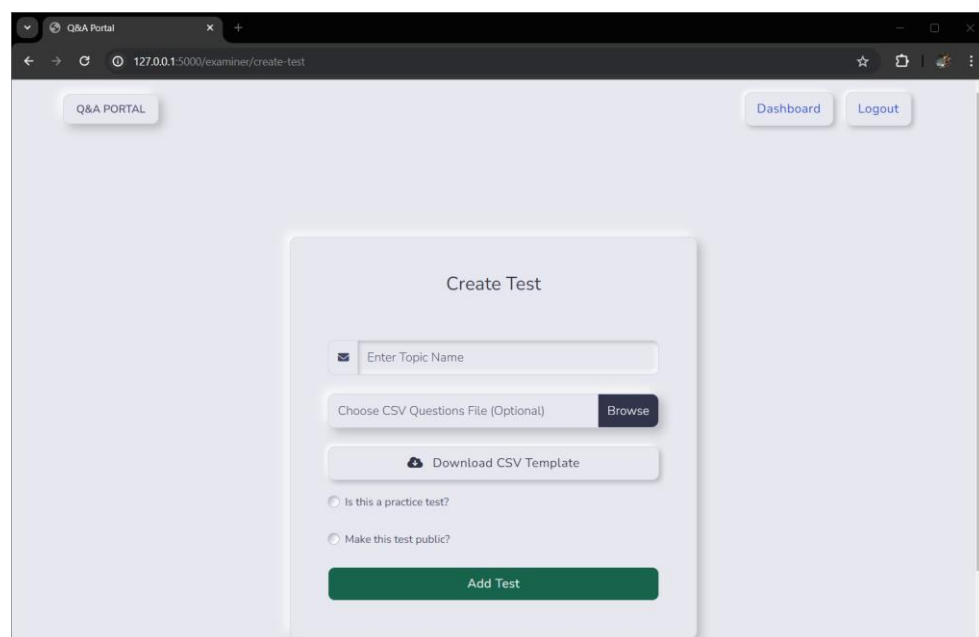


Figure 4.5: Create Test Page

Now, in Create Test Page [Figure 4.5], let's enter the Topic Name for the test. If we have a CSV file with questions, we can upload it by choosing the file by clicking the "Choose CSV Questions File (Optional)" button. We can also download a CSV template to see the required format for the questions file.

There are two checkboxes:

Is this a practice test? (default is unchecked), and

Make this test public? (default is unchecked)

Now we will click on the "Add Test" button to create the test. We will be redirected back to the dashboard.

Let's click on the "View Tests" button to see all the tests we have created.

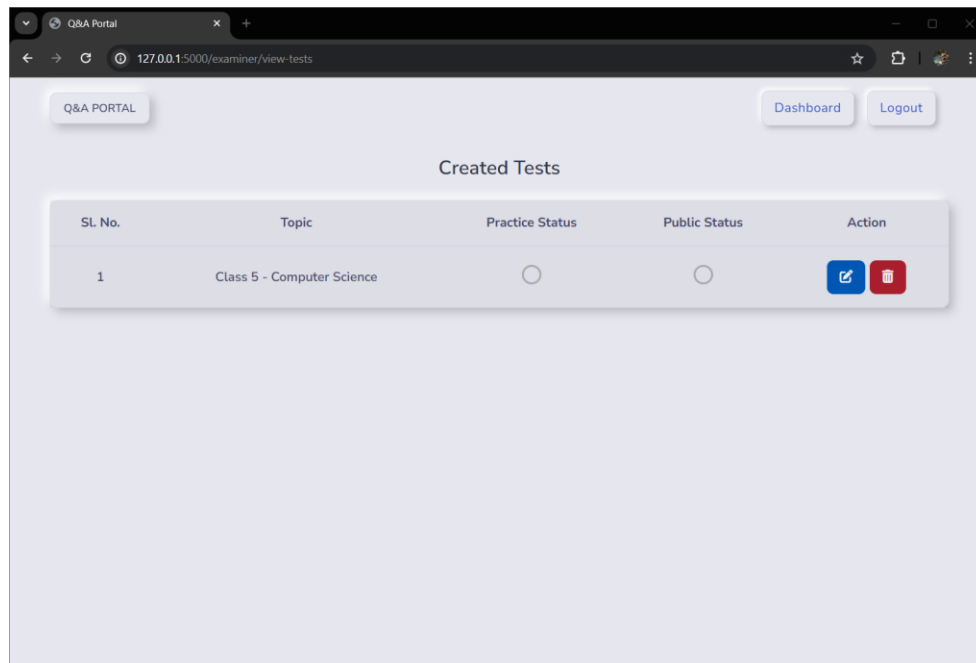


Figure 4.6: View Tests Page

We can see a table with the columns in View Tests Page [Figure 4.6]:  
Sl. No., Topic, Practice Status, Public Status, and Action.

We can toggle the Practice Status and Public Status checkboxes to change the status of the tests.

The Action column has two buttons:

Delete: Deletes the test.

Edit: Takes us to the Display Questions page of the test.

Let's click on the Edit button of the test we added.

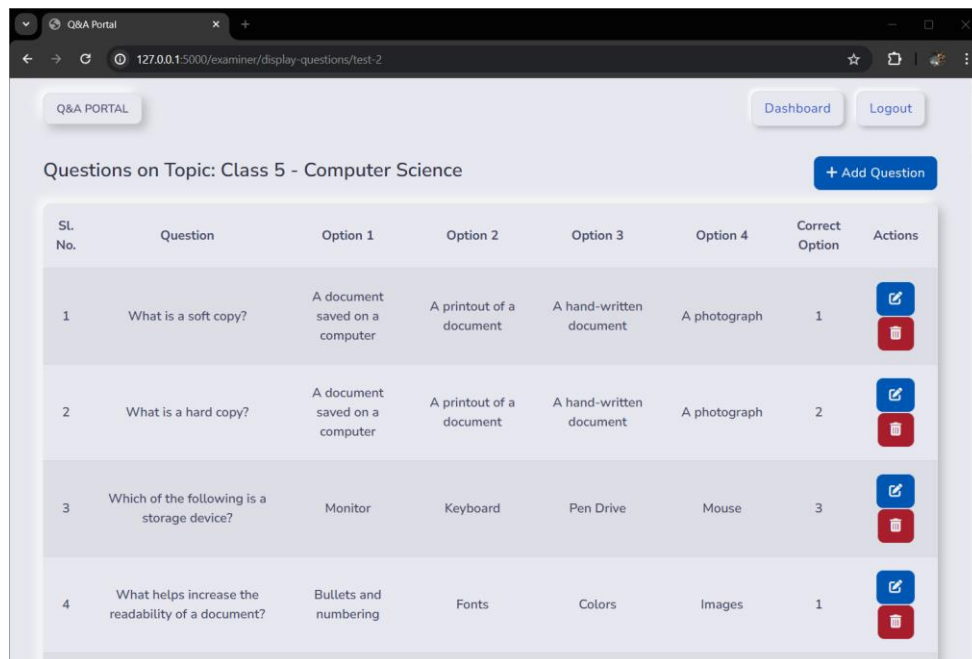


Figure 4.7: Edit Test Page

We can see a table with the questions in the Edit Test Page [Figure 4.7], and we can delete or update each question.

We need to click on the "Add Question" button to add a question.

Q&A PORTAL

Dashboard Logout

### Add Question

Question

? Enter Question Text

Options

1 Enter Option 1 Text

2 Enter Option 2 Text

3 Enter Option 3 Text

4 Enter Option 4 Text

Correct Option

✓ Enter Correct Option Number

Submit

Figure 4.8: Add Question Page

Now, in the Add Question Page [Figure 4.8], we must enter the question text, four possible options, and the correct option. Clicking on the Submit button will save the question.

To view Student Results, we need to click on the "Student Results" button.

Now, in the View Tests Attempted by Students Page [Figure 4.9], we can see a table with the topics of the tests attempted by students.

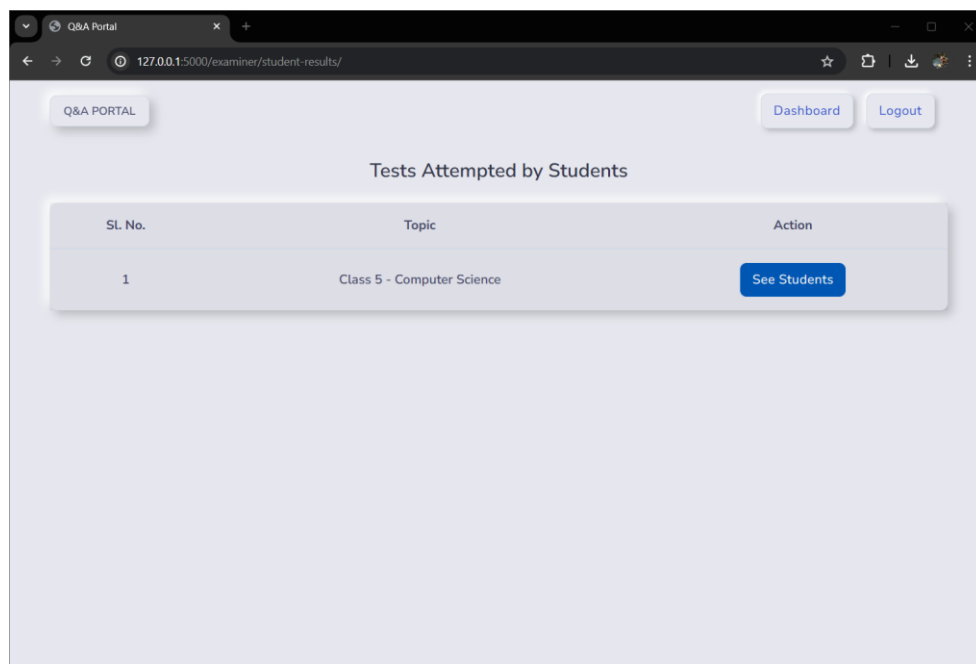


Figure 4.9: View Tests Attempted by Students Page

Clicking on the "See Students" button will let us view the students who attempted the test.

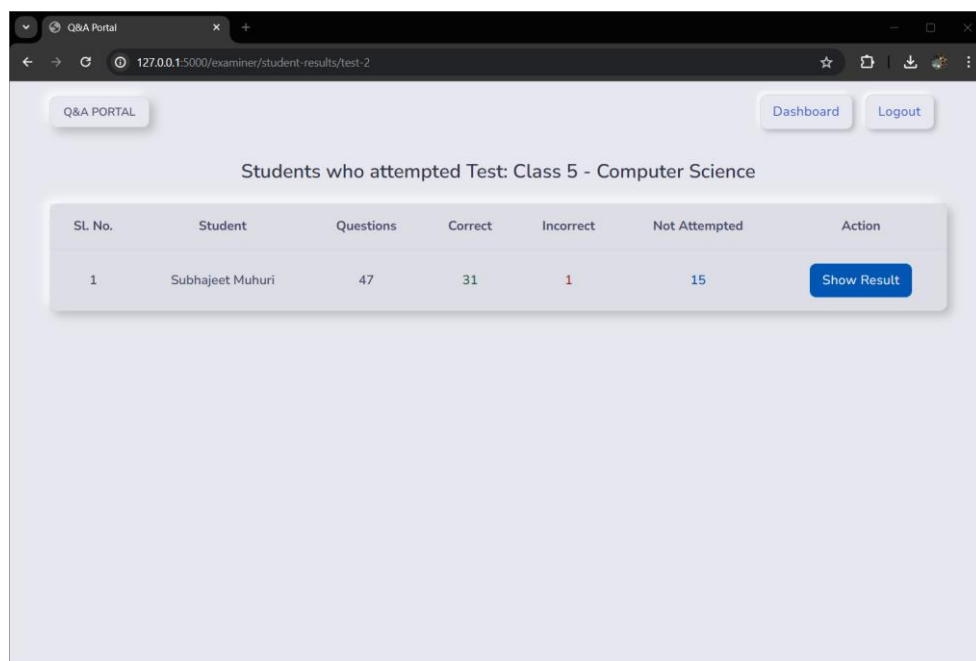
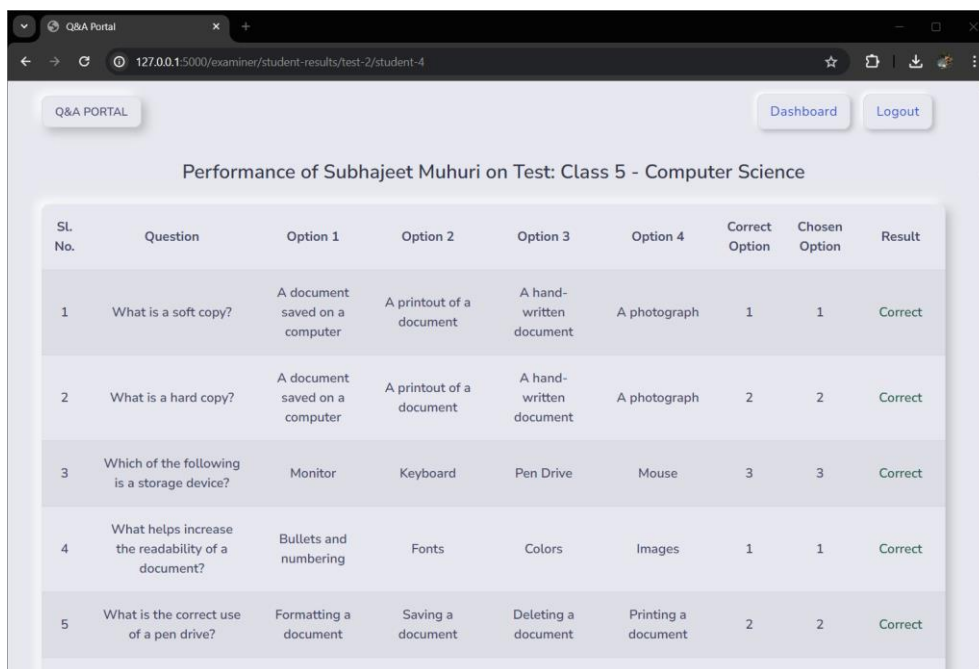


Figure 4.10: View Students Who Attempted Tests Page

We can see a table with the students' names and their results in the View Students Who Attempted Tests Page [Figure 4-10].

Clicking on the "Show Result" button will let us see the detailed results of a student in Student Test Performance Page [Figure 4.11]



The screenshot shows a web browser window with the URL `127.0.0.1:5000/examiner/student-results/test-2/student-4`. The page title is "Q&A Portal". There are buttons for "Dashboard" and "Logout". The main heading is "Performance of Subhajeet Muhuri on Test: Class 5 - Computer Science". Below this is a table with 9 columns: SL. No., Question, Option 1, Option 2, Option 3, Option 4, Correct Option, Chosen Option, and Result. The table contains 5 rows of test results, all marked as "Correct".

| SL. No. | Question                                           | Option 1                       | Option 2                 | Option 3                | Option 4            | Correct Option | Chosen Option | Result  |
|---------|----------------------------------------------------|--------------------------------|--------------------------|-------------------------|---------------------|----------------|---------------|---------|
| 1       | What is a soft copy?                               | A document saved on a computer | A printout of a document | A hand-written document | A photograph        | 1              | 1             | Correct |
| 2       | What is a hard copy?                               | A document saved on a computer | A printout of a document | A hand-written document | A photograph        | 2              | 2             | Correct |
| 3       | Which of the following is a storage device?        | Monitor                        | Keyboard                 | Pen Drive               | Mouse               | 3              | 3             | Correct |
| 4       | What helps increase the readability of a document? | Bullets and numbering          | Fonts                    | Colors                  | Images              | 1              | 1             | Correct |
| 5       | What is the correct use of a pen drive?            | Formatting a document          | Saving a document        | Deleting a document     | Printing a document | 2              | 2             | Correct |

Figure 4.11: Student Test Performance Page

Now, let's register as a student and login. Upon logging in, we will see the Student Dashboard [Figure 4.12].

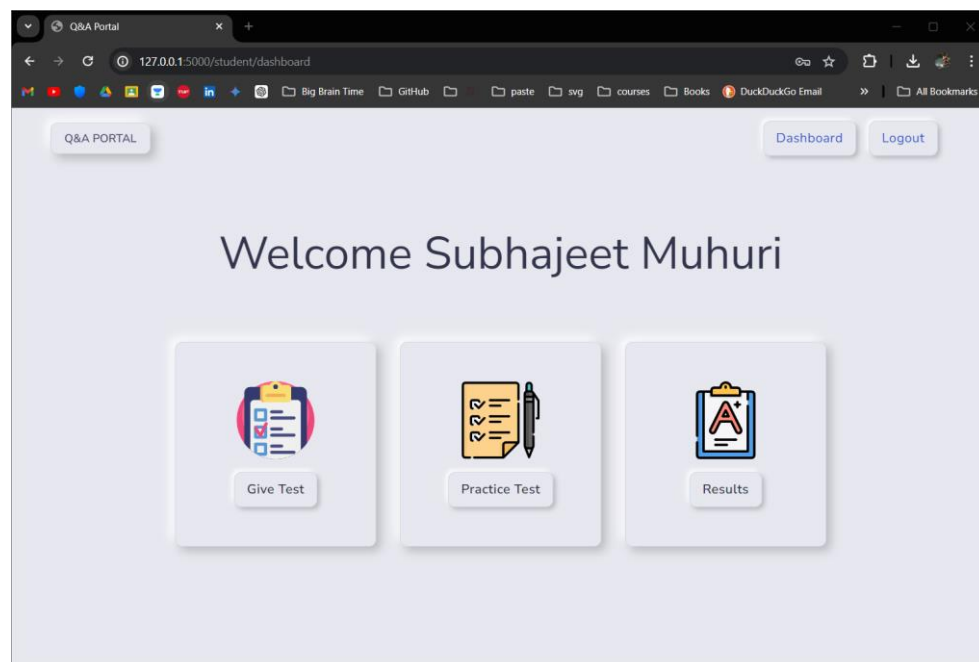


Figure 4.12: Student Dashboard Page

The dashboard has three main buttons:  
Give Test, Practice Test, Results.

Now, let's click on the "Give Test" button.

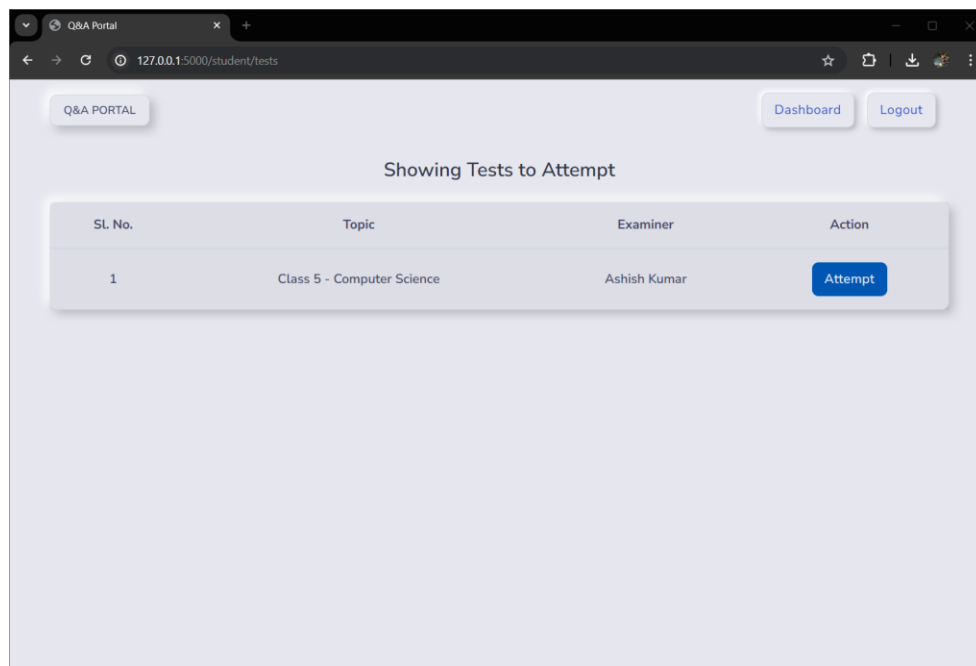


Figure 4.13: Tests to Attempt Page

We can see a table with the tests created by examiners in the Tests to Attempt Page [Figure 4.13]. Let's click on the "Attempt" button to take a test. This will take us to the Test Attempt Page [Figure 4.14].

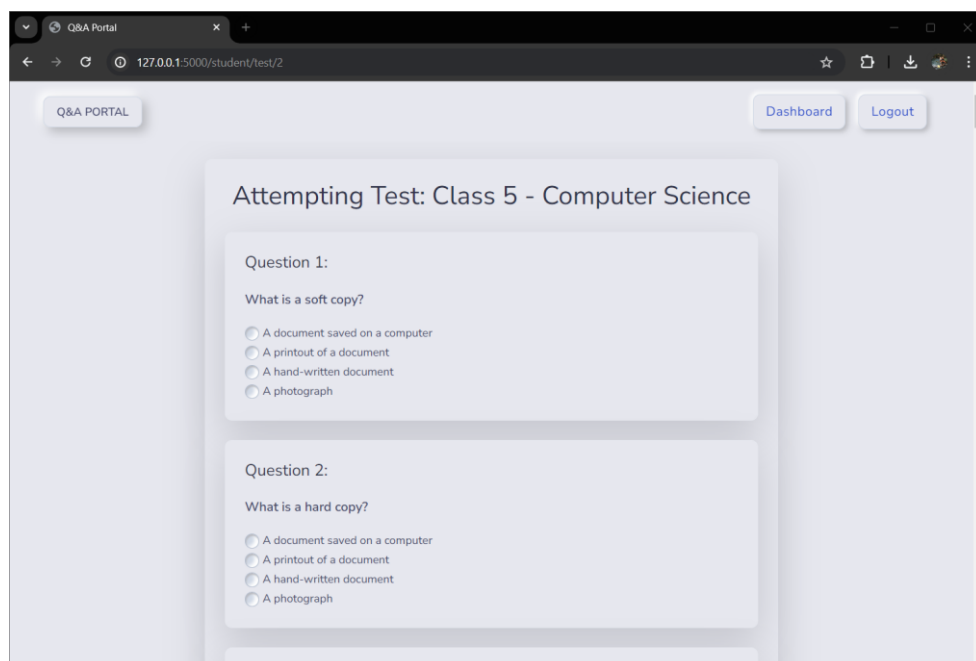


Figure 4.14: Test Attempt Page

We can select the correct options and click on the "Finish Test" button to submit the test.

The "Practice Test" button on the student dashboard takes us to a page containing a table with the practice tests. This is like the Tests to Attempt page. Clicking on the "Attempt" button lets us practice a test [Figure 4.15].

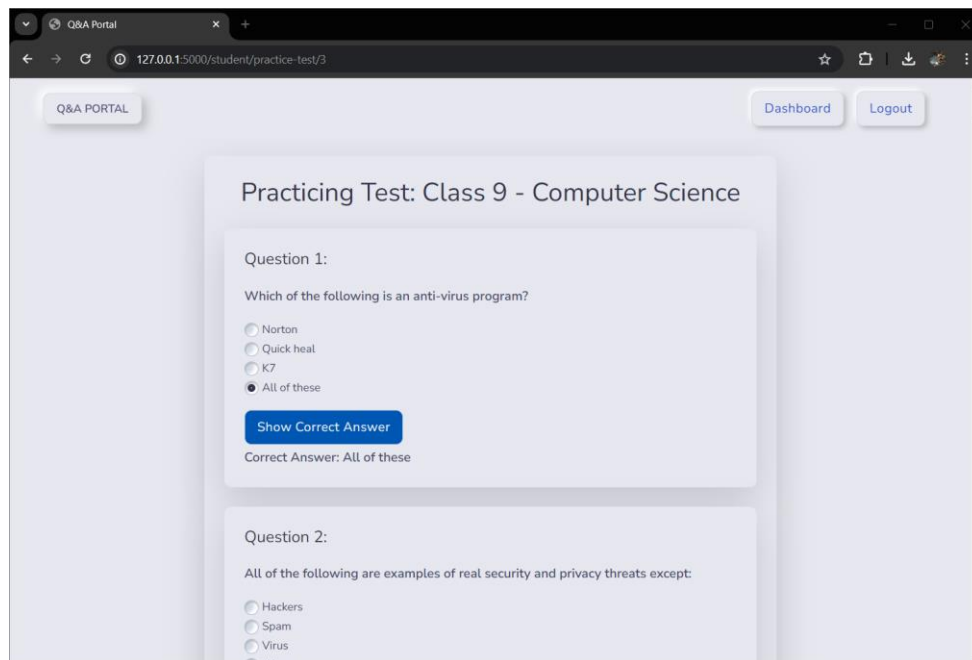


Figure 4.15: Practice Test Page

We need to choose one of the options for answering a question and we can see the correct answers by clicking the "Show Correct Answer" button.

To view our test results we need to click on the "Results" button on the student dashboard. This will take us to the Tests Attempted Page [Figure 4.16]

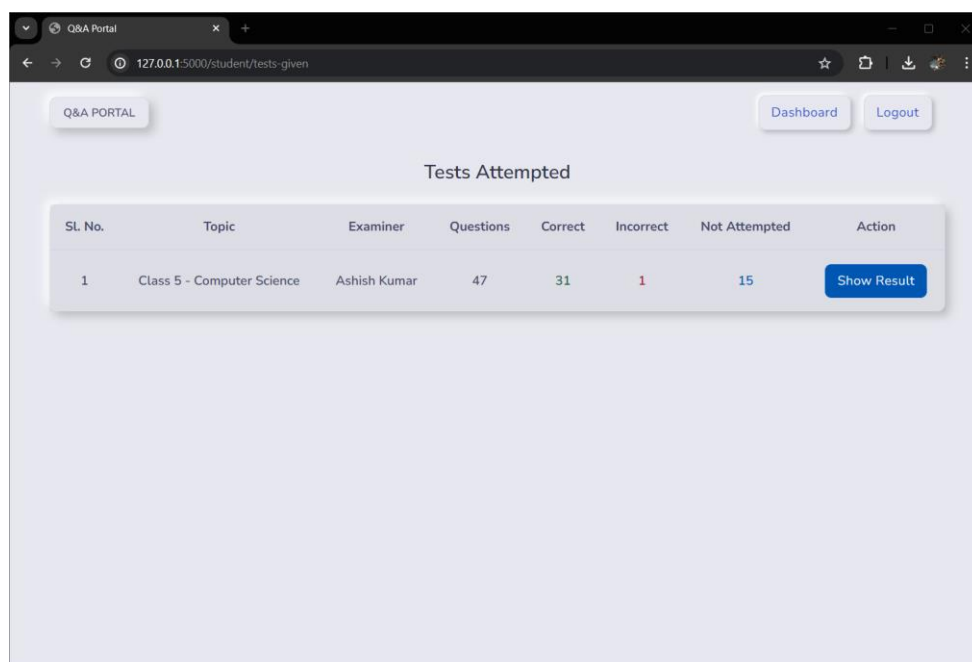


Figure 4.16: Tests Attempted Page

Now, on the Tests Attempted Page [Figure 4.16], we can see a table with the tests we have attempted.

Clicking on the "Show Result" button will let us view the detailed results.

## 4.2 Evaluation of Students' Attempted Tests

After clicking the Attempt button next to the desired test on the Give Test Page, the student is taken to the Attempting Test Page. Here, the test begins, and questions are presented one after another, as shown in Figure 4.17.

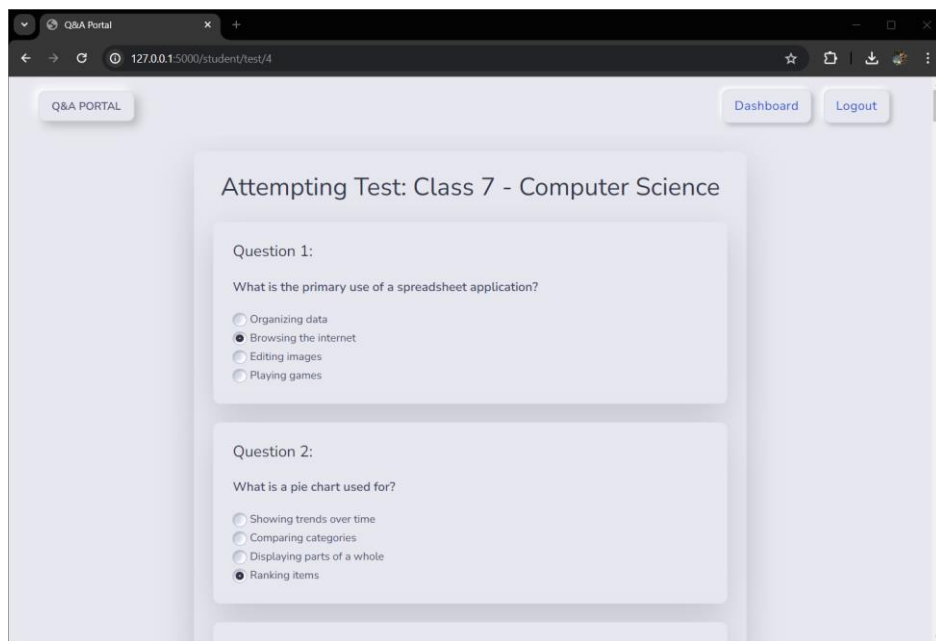


Figure 4.17: Student Attempting Test

Students need to select the correct options for the test questions. Once they have answered the questions to the best of their ability, they complete the process by clicking on the Finish Test button. This action submits their responses and concludes the test-taking session, as shown in Figure 4.18.

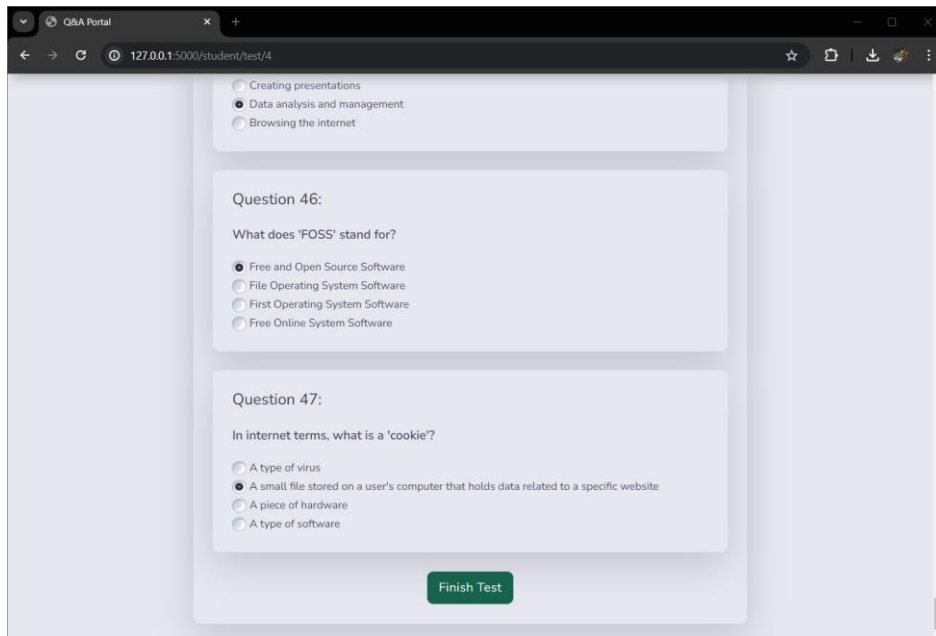


Figure 4.18: Student Submitting Test

Upon submission, students are redirected to their dashboards with a confirmation message. The evaluated results of the students' attempted tests can be viewed by clicking on the Results button, where they can observe their performance in terms of number of correct, incorrect, and unattempted questions, as shown in Figure 4.19.

| SL No. | Topic                      | Examiner     | Questions | Correct | Incorrect | Not Attempted | Action      |
|--------|----------------------------|--------------|-----------|---------|-----------|---------------|-------------|
| 1      | Class 7 - Computer Science | Ashish Kumar | 47        | 6       | 3         | 38            | Show Result |

Figure 4.19: Student Viewing Evaluated Test Results

The detailed report of the evaluated result can be viewed by clicking on the Show Result button, as shown in Figure 4.20.

| Sl. No. | Question                                                                                                       | Option 1                 | Option 2              | Option 3                    | Option 4       | Correct Option | Chosen Option | Result        |
|---------|----------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------|-----------------------------|----------------|----------------|---------------|---------------|
| 1       | What is the primary use of a spreadsheet application?                                                          | Organizing data          | Browsing the internet | Editing images              | Playing games  | 1              | 1             | Correct       |
| 2       | What is a pie chart used for?                                                                                  | Showing trends over time | Comparing categories  | Displaying parts of a whole | Ranking items  | 3              | 1             | Incorrect     |
| 3       | In BASIC programming, which loop structure is used to repeat a set of instructions a specific number of times? | While loop               | For loop              | Do...Until loop             | If...Else loop | 2              | -             | Not Attempted |
| 4       | What is the correct syntax to start a FOR loop in BASIC?                                                       | FOR i TO 10              | FOR i=1 TO 10         | FOR (i=1 to 10)             | FOR i 1 to 10  | 2              | 1             | Incorrect     |

Figure 4.20: Student Viewing Detailed Results

## 4.3 Analysis of Question-Answer Generation

We used a well-structured and specific prompt in ChatGPT and Claude AI, containing clear instructions and a defined correct option format.

The prompt provided to ChatGPT and Claude AI is as follows:

Generate 50 multiple-choice questions from the provided pdf.

The output should be in CSV format.

This CSV should have 6 columns:

Column 1 name should be question\_text which is the question text of the question,

Columns 2, 3, 4, and 5 should be option\_1, option\_2, option\_3, option\_4 respectively, which are the four options of the question,

Column 6 name should be "correct\_option" which will contain the correct option, like 1, 2, 3, or 4.

Keep the content of each column inside quotation marks.

Below is a sample generated response:

```
"question_text","option_1","option_2","option_3","option_4","correct_option"  
"What is the unique column that identifies a record in a table?","Field  
name","Primary key","Record","Cell","2"  
"In a database, what is each row called?","Field","Primary key","Record","Cell","3"  
"What is a 'query' in a database?","A special view of a table","A database update","A  
new record","A primary key","1"  
"What is the main function of DBMS?","Create data","Manage database","Maintain  
records","All of the above","4"  
...
```

As we can see, integrating these advanced AI tools with reliable educational resources from NCERT and IIT Bombay's Computer Masti series enabled us to create high-quality and accurate sets of question-answers. Following this process, we have generated questions covering various topics related to Computer Science, spanning from Class 5 to Class 12 standards. Each standard has been carefully addressed with 50 questions, ensuring comprehensive coverage of the curriculum. Examiners using our platform for tests and evaluations can benefit greatly from these sets of questions. This will let them create question-answers for their tests efficiently.

# Chapter 5: Conclusion and Future Scope

Developing this project has been a rewarding journey. The process of building a functional system that enables examiners to create and manage tests, while also enabling students to attempt these tests and view their results, has been both challenging and fulfilling. Making it accessible to both students and examiners has been a key focus throughout the process.

While the current system provides a solid base, there are still a few areas that could be enhanced and expanded. These improvements will improve the user experience in addition to addressing current shortcomings.

As of now, there is no functionality for users to recover their password if they forget it. We will need to implement a password recovery system, with email verification, to help users regain access to their accounts easily. The user interface can also be more intuitive. The current design is a bit plain and not very engaging. We will need to redesign it to be more user-friendly. Also, the current implementation does not fully address all security concerns. We will need to implement more proper input validation checks and protection against SQL injection and cross-site scripting (XSS).

Addressing these issues will improve the system's robustness, security, and usability, resulting in a better experience for both students and examiners. Future updates will guarantee that the platform remains relevant and valuable in the changing context of educational technology.

## References

- [1] Google Forms: <https://docs.google.com/forms>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [2] Microsoft Forms: <https://forms.microsoft.com/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [3] Moodle: <https://moodle.org/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [4] Kahoot!: <https://kahoot.com/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [5] Neumorphism UI Bootstrap by Themesberg:  
<https://themesberg.com/product/ui-kit/neumorphism-ui-kit-bootstrap>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [6] Wenyin, Liu, et al. "A web-based platform for user-interactive question-answering." *World Wide Web* 12 (2009): 107-124.
- [7] Al-Hakeem, Mazin S., and Mohammad Salim Abdulrahman. "Developing a new e-exam platform to enhance the university academic examinations: The case of Lebanese French University." *International Journal of Modern Education and Computer Science* 9.5 (2017): 9.
- [8] Python 3 documentation: <https://docs.python.org/3/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [9] Flask 3.0.x documentation: <https://flask.palletsprojects.com/en/3.0.x/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [10] The Flask Mega-Tutorial by Miguel Grinberg:  
<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [11] Flask Tutorials by Corey Schafer: <https://www.youtube.com/playlist?list=PLosiE80TeTs4UjLw5MM6OjgkjFeUxCYH>  
(last accessed on May 31, 2024, 08:35 AM IST)

- [12] Jinja 3.1.x documentation: <https://jinja.palletsprojects.com/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [13] MySQL 8.3 Reference Manual: <https://dev.mysql.com/doc/refman/8.3/en/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [14] SQL Tutorial by W3Schools: <https://www.w3schools.com/sql/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [15] CS50x 2024 - Lecture 7 - SQL: <https://youtu.be/1RCMYG8RUSE>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [16] HTML: HyperText Markup Language - MDN Web Docs:  
<https://developer.mozilla.org/en-US/docs/Web/HTML>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [17] CSS: Cascading Style Sheets - MDN Web Docs:  
<https://developer.mozilla.org/en-US/docs/Web/CSS>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [18] JavaScript - MDN Web Docs - Mozilla:  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [19] Bootstrap 5.3 documentation:  
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>  
(last accessed on May 31, 2024, 08:35 AM IST)
- [20] CS50x 2024 - Lecture 8 - HTML, CSS, JavaScript:  
<https://youtu.be/ciz2UaifaNM>  
(last accessed on May 31, 2024, 08:35 AM IST)