

A PROJECT REPORT ON

“ Questionnaire Exam System ”

By

1. Shubham Gundekar

shubhamgundekar262002@gmail.com

2. Nutan Pawar

nutanpawar2223@gmail.com

3. Fouzan Sahibole

en20138727@git-india.edu.in

ABSTRACT

The Questionnaire Exam System is a web application for to take online quiz in an efficient manner and no time wasting for checking the paper. The main objective of Questionnaire Exam System is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. Teachers give papers for students according to their convenience and time and there is no need of using extra thing like paper, pen etc. This can be used in educational institutions as well as in corporate world. Can be used anywhere any time as it is a web based application (user location doesn't matter). No restriction that examiner has to be present when the candidate takes the test. This Web Application provides facility to conduct online examination worldwide. It saves time as it allows number of students to give the exam at a time and displays the results as the test gets over, so no need to wait for the result. It is automatically generated by the server. Administrator has a privilege to create, modify and delete account for teachers. Teachers can create different tests and contexts for students. Students can register, login and give the test with his specific id, and can see the results as well. Teachers can also evaluate different contexts. Students can also see who is currently online and also the highest scores. The entire quiz is divided into different sections, students are allowed to move to a higher section, if there is a minimum pass mark. It provides a competitive platform, where a student not only judges their knowledge/skill but also they can improve their knowledge/skill at the same time.

ACKNOWLEDGMENT

It is a matter of great pleasure for us to get this opportunity to express our sincere sense of gratitude for following. This project is all about “**Questionnaire Exam System**” and our guide **Chakradhar Shinde** helps us to know and make this project successful. We understand the design structure and development of our project under their guidance.

Declaration

We, hereby declare that this project report titled "**Questionnaire Exam System**" is our own work and has been prepared in accordance with the academic guidelines and standards. All the information and data used in this report have been properly cited and referenced.

we further declare that:

- Any external sources of information and material used in this report have been duly acknowledged with proper citations.
- We have not used any previously submitted work, either in full or in part, to fulfill the requirements of this project report.
- This report has not been submitted to any other institution or university for assessment or examination purposes.
- Any contributions made by others to this project report are duly acknowledged.

We understand that any violation of the above declaration may result in academic penalties.

Date: _____

CONTENT

	Page No
1 Introduction	
1.1 Introduction	5
1.2 Problem Statement	6
1.3 Objective	6
2 Features & Functionality	
2.1 Features & Functionality	7
3 Hardware & Software Requirements	
3.1 Hardware requirements	8
3.2 Software requirements	8
4 Diagrams	
4.1 Login & Registration Usecase Diagram	9
4.2 Dataflow Diagram	10
4.3 Class Diagram	10
5 Source Code	
5.1 Source Code	11
6 Output	
6.1 Login	15
6.2 Admin Dashboard	15
6.3 Result	16
7 Conclusion and Future Work	
7.1 Conclusion	17
7.2 Future Work	17
8 References	

8.1 Work Cited	18
8.2 Web Resource	18

1. Introduction

1. 1 Introduction

This initiative focuses on e-learning and the use of online technologies. E-learning platforms are promoted nowadays since they reduce the amount of manual labour required and save time. These platforms are simple to use for anybody with an internet connection, wherever in the globe. Anyone with access to the internet may use it for a variety of purposes, not just in this industry, including online shopping, bill payment, reading newspapers and books, cinema ticket booking, bus and train reservation, and many more. Even while the project isn't really about teaching or studying, it does include assessing students' understanding of a certain subject through the use of objective type questions with choices. Although a set of questions cannot fully assess a person's knowledge, this project aims to assist students in self-evaluation so that, after studying a particular topic, they can correlate their concepts and any concepts they missed can be presented to them in the form of questions by their teacher. As a result, the instructor can also be aware of the ideas on which the pupils are more likely to provide incorrect responses.

JSP stands for Java Server Pages, which is a server-side technology for programming that allows the creation of dynamic (to be built at run time) and platform independent methods for building various Web-based applications. With the help of JSP we can access a wide variety of Java APIs, including JDBC API to access databases. JSP have extension.jsp. Like Servlet, JSP files also need to be in the.jsp format.

1.2 Problem Statement

A user-friendly platform is made possible by the online question answer application by minimising manual work. In the past test were completed manually, but as technology advanced, we were able to conduct them and produce scores automatically. Use of such programmes is greatly valued, particularly at this time when the pandemic is affecting schools and colleges so severely.

1.3 Objective

The main goal of this Questionnaire Exam System application is to offer a platform where teachers and students can interact online. This way, teachers can create questions for a variety of tests or assignments, and students can evaluate themselves after learning a concept by answering the questions that the teachers have uploaded. This reduces the amount of manual paperwork that needs to be completed.

The project has the following characteristics:

- Making of a questions
- Taking of test
- Registration of teachers and students.
- MCQ based questions.

2. Features & Functionality

2.1 Features & Functionality

The Questionnaire Exam System is an extensive software application designed to provide a dynamic and interactive platform for conducting tests and assessments. This project encompasses a wide range of features and functionalities to cater to the diverse needs of educational and corporate settings.

- **User Authentication:**

Users can create accounts, log in, and manage their profiles. Administrators can have special privileges for creation and management of questions and answer

- **Question Creation:**

Administrators can create question, add correct and incorrect options, specify time limits.

- **Question Types:**

Support for various question types including multiple-choice, true/false, fill in the blanks, and essay questions.

- **Management:** Administrators can edit, delete, modify questions.

- **Randomization:** Questions and answer choices can be randomized to prevent cheating.

- **Score Calculation:** Automated scoring and immediate feedback on correct/incorrect answers.

- **User-Friendly Interface:** An intuitive and responsive user interface for both administrators and participants.

- **Result Summary:** Participants can view their results, correct answers, and their score.

3. Hardware & Software Requirements

3.1 Hardware Requirements

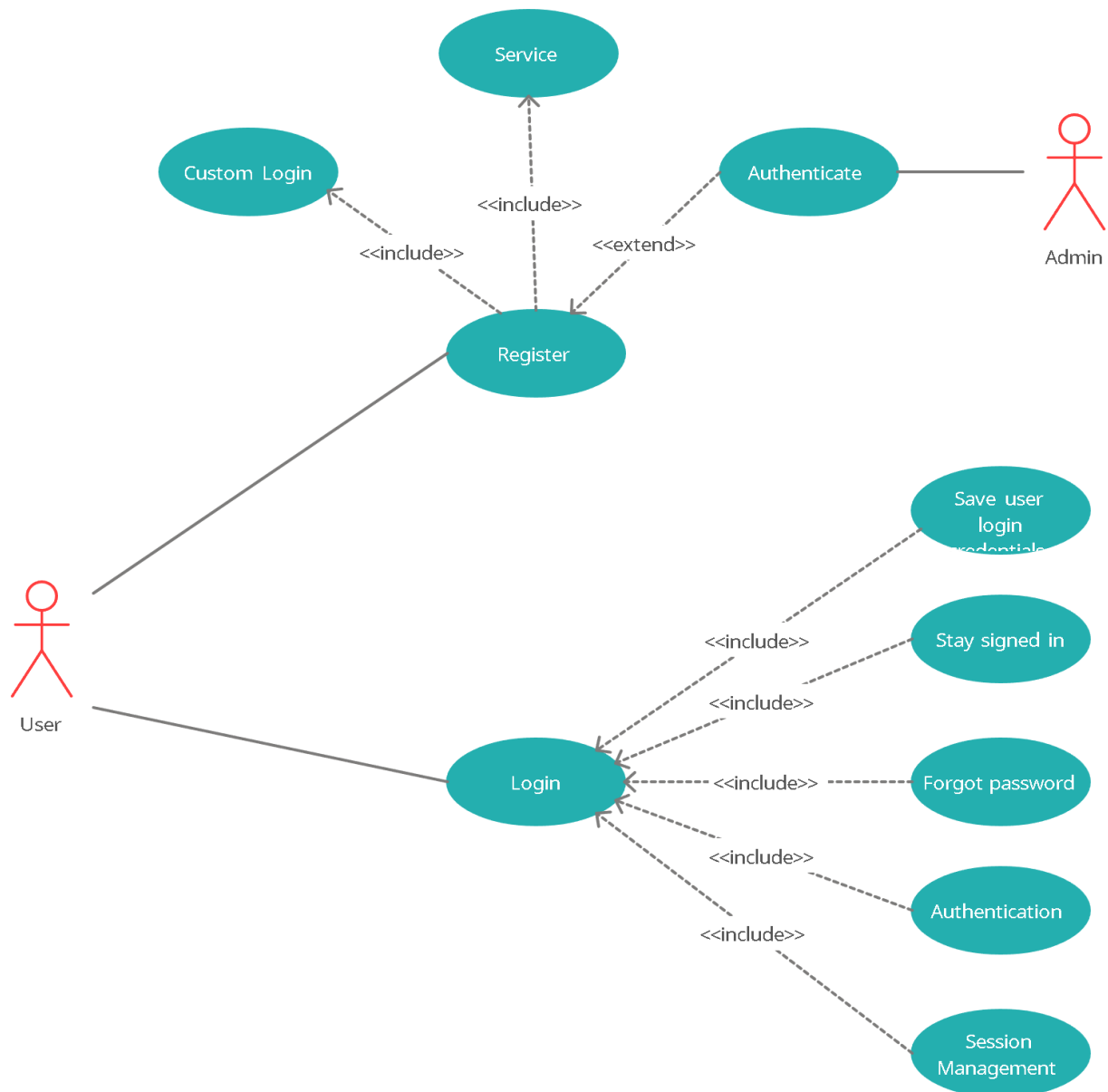
- **Processor:** x86 or x64.
- **RAM :** 512 MB (minimum), 1 GB (recommended)
- **Hard disk:** up to 300 MB of available space may be required. However, 90 MB free space is required in boot drive even if you are installing in other drive.

3.2 Software Requirements

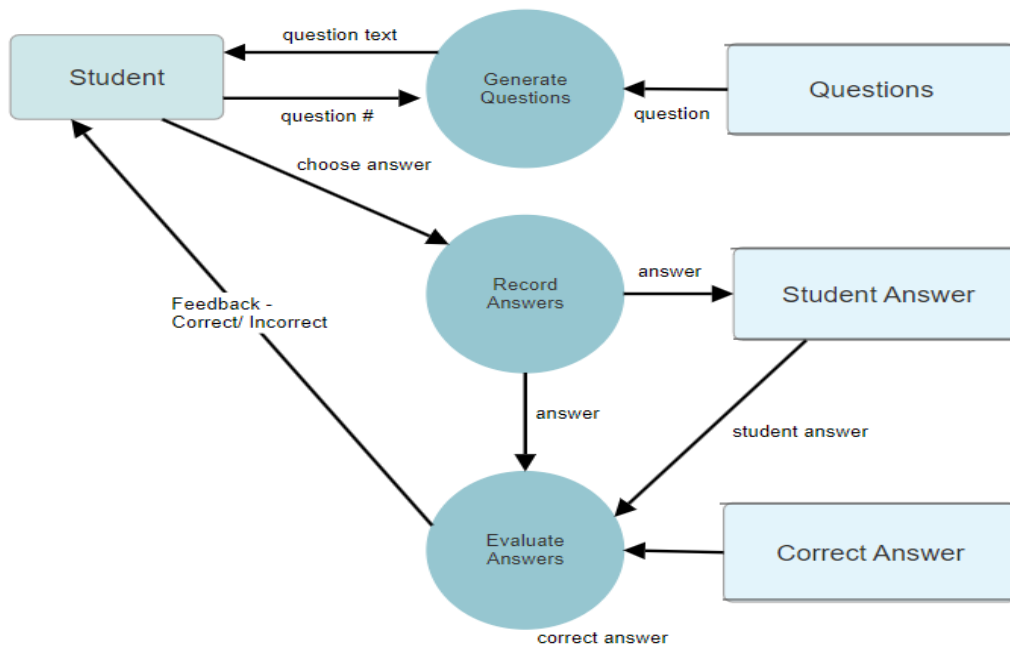
- Java Development Kit (JDK), Java Runtime Environment (JRE) version (1.7 or 1.8) needs to be installed and configured.
- Any web servers (Apache Tomcat 9) that's supports the Java Servlet, JSP specification. Also, ensure that the web servers need to be tested and configured with the IDE.
- Eclipse IDE for Enterprise developers with J2EE support – (Lune, Kepler, Mars, or later) or NetBeans IDE with Java2E support.

4. Diagrams

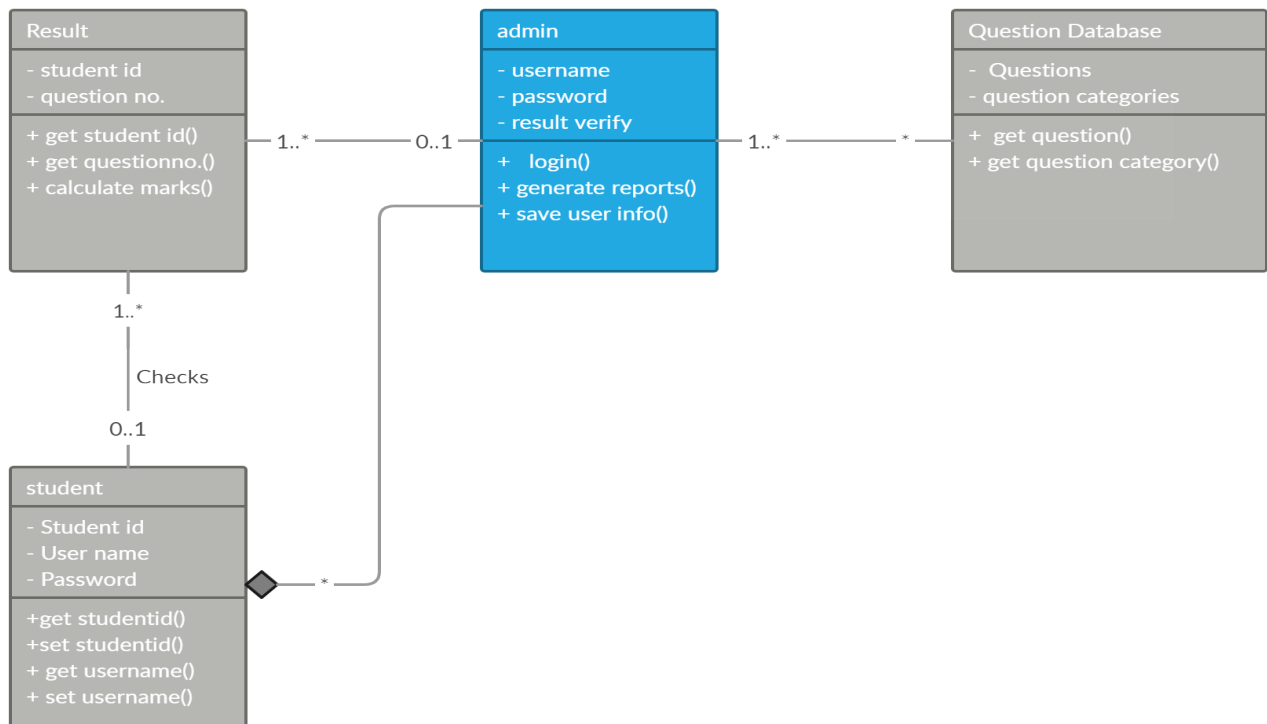
4.1 Login & Registration



4.2 Dataflow Diagram



4.3 Class Diagram



5. Source Code

```

import java.util.*;
public class QuizGame {
    private static Map<String, Quiz> quizzes = new HashMap<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("Enter a command: (create, take, view, list,
exit)");
            String command = scanner.nextLine();
            if (command.equals("create")) {
                createQuiz(scanner);
            } else if (command.equals("take")) {
                takeQuiz(scanner);
            } else if (command.equals("view")) {
                viewQuiz(scanner);
            } else if (command.equals("list")) {

```

```

        listQuizzes();
    } else if (command.equals("exit")) {
        break;
    } else {
        System.out.println("Invalid command.");
    }
}

private static void createQuiz(Scanner scanner) {
    System.out.println("Enter the name of the quiz:");
    String quizName = scanner.nextLine();
    Quiz quiz = new Quiz(quizName);

    System.out.println("Enter the number of questions:");

    int numQuestions = Integer.parseInt(scanner.nextLine());
    for(int i = 0; i < numQuestions; i++)
    {
        System.out.println("Enter the question: ");
        String question = scanner.nextLine();
        System.out.println("Enter the number of choices: ");

        int numChoices = Integer.parseInt(scanner.nextLine());
        List<String> choices = new ArrayList<>();
        for(int j = 0; j < numChoices; j++)
        {
            System.out.println("Enter choice " + (j+1) + ":");
            String choice = scanner.nextLine();
            choices.add(choice);
        }

        System.out.println("Enter the index of the correct choice: ");
        int correctChoice = Integer.parseInt(scanner.nextLine()) - 1;
        quiz.addQuestion (new Question
            (question, choices, correctChoice));
    }
    quizzes.put(quizName, quiz);
    System.out.println("Quiz created.");
}

private static void takeQuiz (Scanner scanner) {
    System.out.println("Enter the name of the quiz:");
    String quizName = scanner.nextLine();

```

```

Quiz quiz = quizzes.get(quizName);
if (quiz == null) {
System.out.println("Quiz not found.");
return;
}
int score = 0;
for (int i = 0; i < quiz.getNumQuestions(); i++) {
    Question question=quiz.getQuestion(i);

System.out.println("Question " + (i + 1) + ":"
+ question.getQuestion());
List<String> choices = question.getChoices();
for (int j = 0; j < choices.size(); j++) {
    System.out.println((j + 1) + ": " + choices.get(j));
}
System.out.println("Enter your answer: ");
int userAnswer = Integer.parseInt(scanner.nextLine())- 1;
if (userAnswer == question.getCorrectChoice())
{ System.out.println("Correct!");
score++;
} else {
System.out.println("Incorrect. The correct answer is "
+ (question.getCorrectChoice() + 1) + ".");
}}
System.out.println("Your score is " + score + " out of " + quiz.getNumQuestions() +
".");
}

private static void viewQuiz (Scanner scanner) {
System.out.println("Enter the name of the quiz:");
String quizName = scanner.nextLine();
Quiz quiz=quizzes.get(quizName);
if (quiz == null) {
System.out.println("Quiz not found.");
return;
}

System.out.println("Quiz: " + quiz.getName());
for (int i = 0; i < quiz.getNumQuestions(); i++) {
Question question = quiz.getQuestion(i);
System.out.println("Question " + (i + 1) + ": " +
question.getQuestion());
List<String> choices = question.getChoices();
for (int j = 0; j < choices.size(); j++) {

```

```

        System.out.println((j + 1) + ": " + choices.get(j));
    }
    System.out.println("Answer: " +
(question.getCorrectChoice() + 1));
}
}

private static void listQuizzes() {
    System.out.println("Quizzes:");
    for (String quizName : quizzes.keySet()) { System.out.println("- " + quizName);
    }
    }
}

class Quiz {
    private String name;
    private List<Question> questions = new ArrayList<>();
    public Quiz(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void addQuestion (Question question) { questions.add(question);
    }

    public Question getQuestion(int index) { return questions.get(index);
    }

    public int getNumQuestions() {
        return questions.size();
    }
}

class Question{
    private String question;
    private List<String> choices;
    private int correctChoice;
    public Question (String question, List<String> choices, int correctChoice) {
        this.question = question;
        this.choices = choices;
        this.correctChoice = correctChoice;
    }

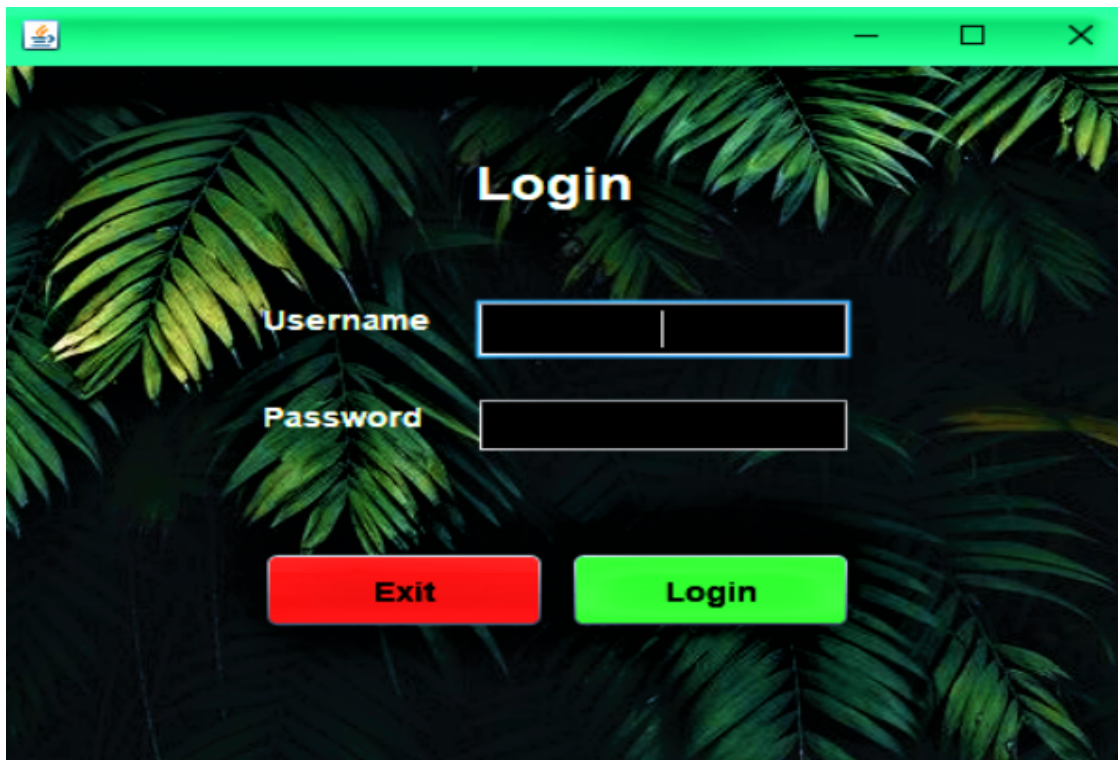
    public String getQuestion() { return question;
    }
}

```

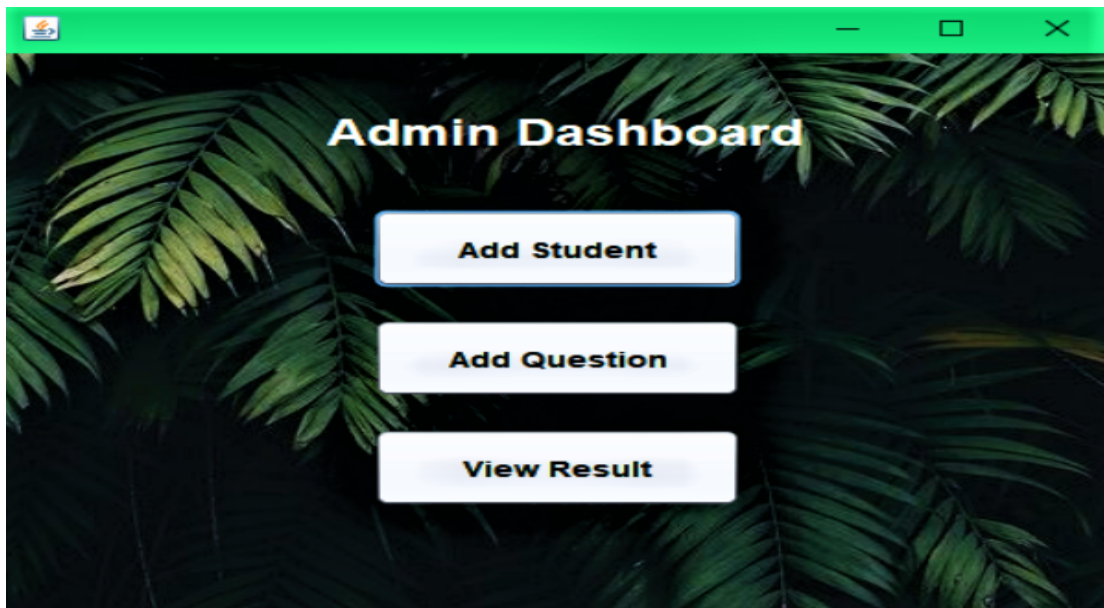
```
public List<String> getChoices() { return choices;
}
public int getCorrectChoice() { return correctChoice;
}
}
```

6. Output

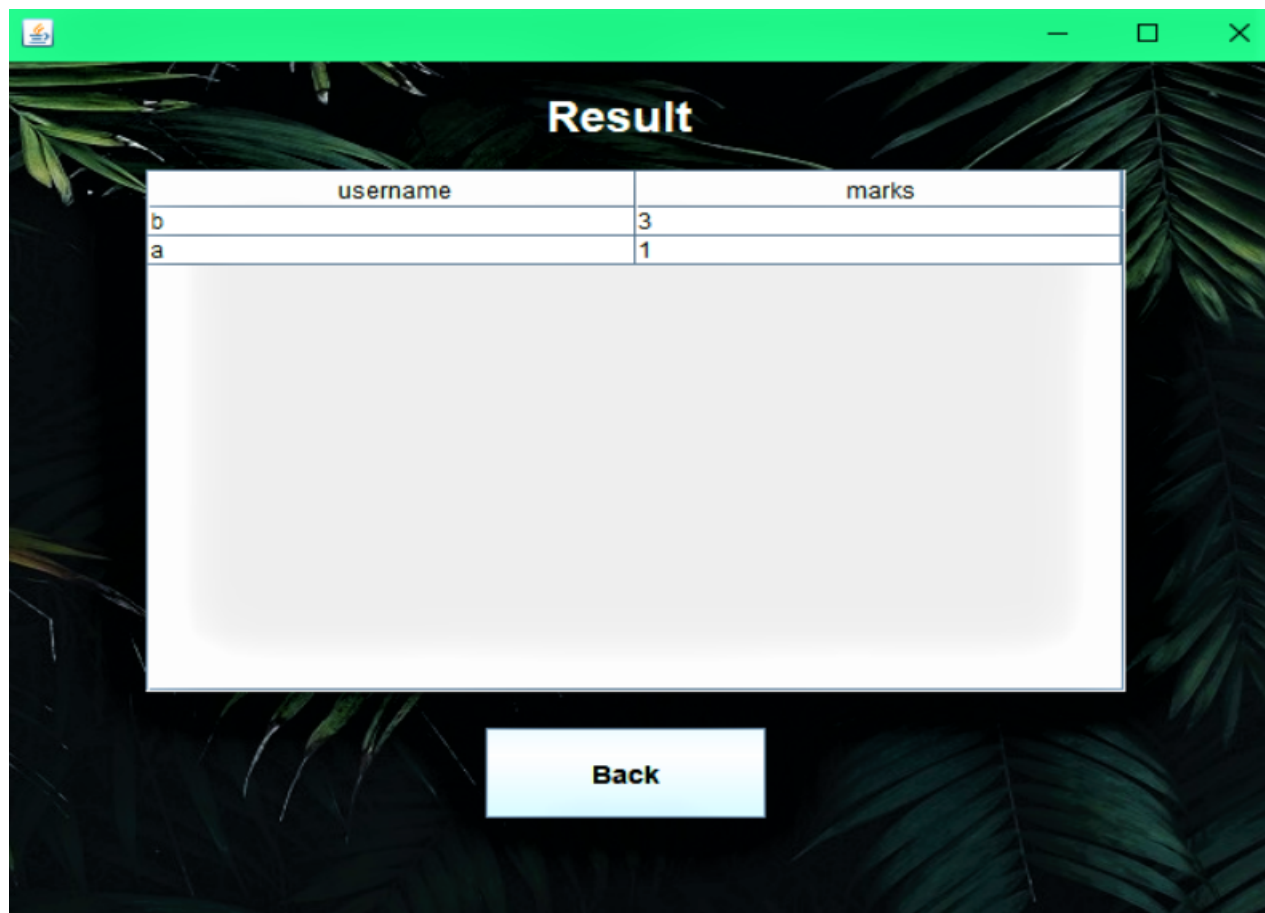
6.1 Login



6.2 Admin Dashboard



6.3 Result



7. Conclusion and Future Work

7.1 Conclusion

Following the completion of this project, we decided that this web application meets the needs and requirements of the client and is user friendly. This tiny project also helped me grasp the design, coding, and implementation stages that are involved in the creation of any project. Many concepts were updated, and many were learned for the first time while developing this online application.

7.2 Future Work

This project has a broad scope because many more features can be added to it. In first place, each question will have a timer attached. Because there is no time constraint for the test, students can get answers to the questions anyplace else, resulting in an untrue assessment of their performance. To prevent this, many other features, particularly for students, can be introduced to the user side, such as barring any form of tab switching in the browser and introducing a webcam for more honest evaluation of the students. By adding an administrative component to the system, a more creative manner of displaying results and deleting tests is possible. User addition and deletion can be done under the administrative section.

8. References

8.1 Works Cited

1. Developing dynamic website using an online website builder that is Weebly for Viking Fortune [Md. Shamsul Arafin Yi Jiang]- Laurea University of Applied Sciences
2. Web Usability: A Literature Review [Giselle Joy Esmeria¹ and Rosemary R. Seva] Industrial Engineering Department, De La Salle University Science and Technology Complex

8.2 Web Sources

1. <https://www.javatpoint.com/jsp-architecture>
2. <https://www.tutorialspoint.com/servlets-tutorial> <https://www.edureka.com/jsp/>
3. <https://www.researchgate.net/publication/>
4. <https://www.geeksforgeeks.com/jsp-architecture/>
5. <https://guru99.com/servlets/>