# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Mind Matrix app is innovative educational platforms that leverage technology to provide high-quality and personalized learning experiences to students of all ages. These apps offer a wide range of educational content, including video lectures, interactive quizzes, practice tests, and supplementary study materials, covering various subjects and academic levels . Key features of these apps include adaptive learning algorithms, which tailor the content to individual learning needs, and the ability to learn at one's own pace. Gamification elements and engaging multimedia content make the learning process enjoyable and effective. Additionally, some apps incorporate live tutoring sessions, ensuring students receive real-time assistance when needed. Mind Matrix app have transformed the traditional education landscape, making learning more accessible ,interactive, and effective."

In the realm of interactive learning and engagement, the Java-based Quiz Application emerges as a versatile and user-friendly tool designed to facilitate seamless quiz creation, administration, and participation. Leveraging the power of Java programming language, this application provides educators, businesses, and enthusiasts with a robust platform to design, deploy, and manage quizzes effortlessly.

Technology in the field of education is constantly evolving, growing and this increase will continually offer new and extraordinary advances in the field of education. Traditional E-learning systems developed for laptop and desktops were based on standalone application and web based application architecture. These applications will have some limitations to be used effectively or we cannot use them easily .since these applications need a computing device and network connectivity.

With the advancement in the mobile technology and availability of smart mobile devices and the networks we can create a system which can be used to check knowledge levels of students. Since mobile network is available at maximum areas so this can be used for the knowledge testing of any person specially candidates of software company who need a specific skill for job

# 1. INTRODUCTION

## 1.1 Introduction to application

Mind Matrix is a comprehensive online educational platform that provides a comprehensive selection of educational materials for learners of all educational levels, from elementary school to high school, as well as competitive examinations and professional development. It is designed to facilitate engaging, interactive learning through the use of video tutorials, quizzes, adaptive learning technologies, and other features.

Online quizzes and learning apps have revolutionized the traditional education landscape, bringing about a paradigm shift in the way individuals acquire knowledge and skills. In recent years, the integration of technology into education has given rise to a plethora of innovative platforms designed to enhance the learning experience. Online quizzes and learning apps, in particular, have become instrumental in fostering interactive and engaging educational environments.

Moreover, the gamification elements incorporated into many online quizzes and learning apps add an element of fun and competition to the learning process. Achievements, badges, and leaderboards motivate learners to stay engaged and strive for improvement, creating a more enjoyable and rewarding educational journey.

In conclusion, the advent of online quizzes and learning apps has ushered in a new era of education characterized by flexibility, accessibility, and interactivity. These tools have become integral in meeting the diverse needs of learners, providing a platform for continuous skill development and knowledge acquisition in the ever-evolving digital age. As technology continues to advance, online learning platforms are likely to play an increasingly crucial role in shaping the future of education.

## 1.2 Objectives:

Another key objective is to facilitate personalized learning experiences. Online quizzes and learning platforms leverage adaptive technologies to tailor content delivery to individual learner needs, accommodating various learning styles and paces. This personalization aims to optimize comprehension and retention, ensuring that educational content is presented in a manner that resonates most effectively with each learner.

Efficiency in assessment is a crucial goal as well. Online quizzes aim to provide timely and constructive feedback to learners, enabling them to gauge their understanding of the material and identify areas for improvement. The automation of assessment processes not only streamlines evaluation but also allows for the quick adaptation of content based on individual performance.

Furthermore, the integration of interactive elements, multimedia resources, and gamification techniques within online learning platforms serves to enhance engagement. By creating dynamic and stimulating learning environments, these tools aim to captivate learners' interest, encouraging active participation and sustained attention throughout the learning process.

## 1.3 Goal of Project:

The goal of  the project are as follows:

### 1.Enhanced Learning Experience

- Facilitate an engaging and interactive learning experience for students or participants.
- Provide multimedia content such as videos, images, and simulations to enhance understanding.

### 2.Flexibility and Accessibility

- Enable learners to access educational materials and quizzes from anywhere at any time.
- Support different learning styles by offering a variety of resources.

### 3.Assessment and Feedback

- Provide a platform for creating and delivering quizzes to assess learners' understanding.
- Offer instant feedback to learners to help them understand their strengths and areas for improvement.

### 4.Progress Tracking

- Implement features that allow learners and instructors to track progress over time.
- Generate reports and analytics to identify patterns and trends in learning outcomes.

### 5.Personalization

- Incorporate adaptive learning techniques to tailor content and quizzes based on individual learner needs.
- Allow learners to set their own pace and choose topics of interest.

### 6.Collaboration and Social Learning

- Foster collaboration among learners through discussion forums, group projects, or peer assessments.

- Utilize social learning features to encourage knowledge sharing and community building.

## 7.Integration with Learning Management Systems (LMS)

- Ensure seamless integration with existing learning management systems for easy administration and tracking.
- Support interoperability standards to facilitate data exchange with other educational tools and platforms.

## 8.Security and Integrity

- Implement robust security measures to protect the integrity of quizzes and prevent cheating.
- Safeguard user data and ensure compliance with privacy regulations**.**

# 2.LITERATURE SURVEY

A literature survey is the research of the apps and websites that are similar to this application and the given are some of the drawbacks of that applications.

## 1. Gamification Pitfalls

Issue: While gamification can enhance engagement, it may also lead to a focus on winning rather than learning.

Implication: Students may prioritize earning points over understanding concepts, potentially defeating the purpose of educational quizzes.

## Limited Content Coverage

Issue: Some quiz apps may only cover specific topics or use a narrow range of question types.

Implication: This limitation might not provide a comprehensive learning experience and may neglect important aspects of a subject.

## Cheating and Collaboration`

Issue: Users can easily cheat by looking up answers or collaborating with others during quizzes.

Implication: The assessment may not accurately reflect individual understanding, compromising the integrity of the educational process.

## Accessibility and Equity

Issue: Access to quiz apps may be limited by technological barriers or socio-economic factors.

Implication: This can contribute to educational inequalities, disadvantaging students who do not have access to the necessary devices or internet connectivity.

## Overemphasis on Speed

Issue: Some quiz apps reward quick answers, which may prioritize speed over accuracy.

Implication: Students might rush through questions, leading to errors and a lack of focus on understanding the material thoroughly.

## Inadequate Feedback:

Issue: Some quiz apps may provide limited or generic feedback on incorrect answers.

Implication: Students may miss opportunities for learning from mistakes, as they are not given detailed explanations of correct answers.

## Lack of Personalization:

Issue: Quiz apps may not adapt to individual learning styles or pace.

Implication: Some students may feel overwhelmed or bored if the content is not tailored to their needs, affecting their engagement and motivation.

**Privacy Concerns:**

Issue: Data privacy and security concerns may arise, especially when quiz apps collect and store personal information.

Implication: Users may be hesitant to use these apps if they are concerned about the misus their data.

## 2.1 Existing System

The existing system of online Mind Matrix is a dynamic and technology-driven platform that allows individuals to assess their knowledge and skills in various subjects or topics through a digital interface. Typically, these quizzes are accessible through websites or mobile applications, providing users with the flexibility to participate from anywhere with an internet connection. The system is designed to cater to a wide range of users, including students, professionals, and enthusiasts seeking to test and enhance their understanding of specific subjects.

Key features of the existing online quiz system include a user-friendly interface, diverse question formats (multiple-choice, true/false, fill in the blanks, etc.), and immediate feedback upon completion. Participants often have the option to select the difficulty level of the quiz, enabling personalized learning experiences. Additionally, many systems incorporate time constraints for each question, adding an element of challenge and promoting quick thinking.

Behind the scenes, these systems typically include a database to store questions, user profiles, and results. Some platforms also employ algorithms to generate randomized questions, reducing the likelihood of cheating and enhancing the overall integrity of the assessment process. Moreover, integration with learning management systems (LMS) or educational platforms may be present to facilitate seamless tracking of progress and performance.

## 2.2 Problem Identification

Quiz apps, while popular in educational settings, face several challenges that can impact their effectiveness. One prominent issue is the tendency to promote shallow learning. These apps may inadvertently encourage memorization of facts without fostering a deeper understanding of the subject matter, potentially hindering the development of critical thinking skills among users. Another concern is the potential overemphasis on gamification elements, such as points and leaderboards, which may overshadow the primary educational objectives. This can lead to a disconnect between the game-like features and the substantive content, with users prioritizing winning over genuine learning.

Additionally, the limited diversity of question types in some quiz apps poses a challenge. A narrow set of questions may not effectively assess a student's comprehensive understanding and problem-solving abilities. Cheating and collaboration present ongoing issues, as users can easily access external resources or collaborate with peers during assessments. This compromises the integrity of the evaluation process, resulting in outcomes that may not accurately reflect individual knowledge and skills.

Personalization issues also arise, as quiz apps may not adapt to individual learning styles, preferences, or pace. This lack of customization can result in disengagement or frustration among users who find the content either too challenging or too easy. Privacy and security concerns, such as the collection of sensitive user information, pose potential risks that may erode user trust in these platforms. Lastly, the limited interactivity of some quiz apps can lead to a less engaging educational experience compared to platforms that incorporate multimedia, simulations, or collaborative elements.

Addressing these challenges requires a comprehensive approach, involving improvements in app design, instructional strategies, and a keen awareness of the diverse needs of users.

# 3.RESEARCH METHODLOGY

## 3.1 Proposed Algorithm

**Algorithms** : Learning, linear regression are used in developing this application.

**User Authentication**:

Users should be able to register or log in to the system.

Store user credentials securely (use encryption and hashing).

**User Profile**:

Create a user profile to store user information, such as username, email, and scores.

**Quiz Creation**:

Allow quiz creators (teachers, administrators) to create quizzes.

Define quiz parameters, such as title, duration, and number of questions.

For each question, specify the question type (multiple choice, true/false, etc.) and provide options.

**Quiz Storage:**

Store quizzes in a database, associating each question with its respective quiz.

Include information about correct answers, scoring, and any multimedia components (images, videos).

**Quiz Assignment:**

Allow quiz creators to assign quizzes to specific users or groups.

**Quiz Taking**:

When a user starts a quiz, load the questions and options from the database.

Implement a timer for timed quizzes.

Allow users to select answers for each question.

**Answer Validation**:

Validate user-submitted answers against the correct answers stored in the database.Calculate and store the user's score.

**Result Tracking:**

Store quiz results in the database, associating them with the respective user and quiz.

Provide feedback to users on their performance, showing correct and incorrect answers.

**Leaderboards**:

Implement leaderboards to display top scores for users or groups.

**Security Measures**:

Implement security features to prevent cheating, such as randomizing question order and shuffling answer choices.

Protect against common web vulnerabilities (e.g., SQL injection, cross-site scripting).

**User Interface:**

Design an intuitive and user-friendly interface for quiz creation, taking, and result viewing.

Ensure responsiveness for various devices.

**Scalability:**

Design the system to scale by optimizing database queries and considering potential increases in user and quiz numbers.

**Notifications:**

Implement notifications for users to receive updates on assigned quizzes, completed quizzes, and scores.

**Review and Edit:**

Allow users to review and revisit completed quizzes.

Enable quiz creators to edit and update quizzes as needed.

**Logging and Analytics:**

Implement logging for user actions and errors for troubleshooting.

Integrate analytics to gather insights into user behavior and quiz performance.

# 4.REQUIREMENTS

## 4.1 Software Requirements

i. **Front-End :** XML, UI DESIGN, WORKSPACE

ii. **Back-End :** JAVA

iii. **Database :** JSON

iv. Swing Framework is used….

## 4.2 Hardware Requirements

**Database Server:**

i. **Processor:** Similar to the application server, a multi-core processor for handling database queries efficiently.

ii. **Storage:** SSD storage is preferable for faster data retrieval.

iii. **RAM:** Sufficient RAM for caching and handling database operations. Again, the amount depends on the size of the database.

## Backup System:

i. A reliable backup system to ensure data integrity and recovery in case of failures.

## Security Measures:

Ensure the implementation of security measures such as firewalls, encryption, and regular security audits to protect the application and user data.

# 5.DESIGN AND IMPLEMENTATION

## 5.1 Design

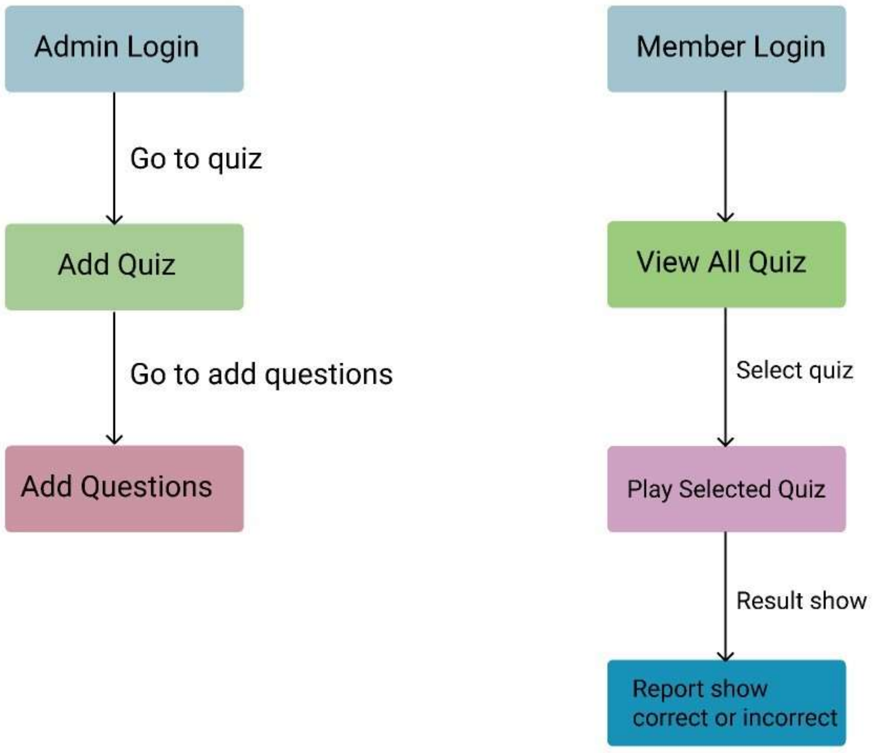


**Fig : 5.1**

### 1. Quiz Logic

- Implement logic to load questions from storage and initialize the `Quiz` object.
- Handle user input and validate answers.
- Keep track of the user's score and provide feedback after completing the quiz.

### 2. Scoring System

- Design a scoring system based on correct and incorrect answers.
- Consider additional features like time-based scoring or bonus points for consecutive correct answers.

## 3. Enhancements

- Allow users to customize the quiz settings, such as the number of questions or difficulty level.
- Implement a timer for each question or an overall time limit for the entire quiz.
- Add features like lifelines or hints for users.

## 4. Error Handling

- Implement robust error handling to handle unexpected user input or issues with data loading.
- Provide informative error messages to guide users in case of mistakes.

## 5. Testing

- Thoroughly test your application with various scenarios to ensure it behaves as expected.

## 6. User Feedback

- Consider adding feedback messages or a summary at the end of the quiz to inform users of their performance.

## 5.2 Implementation

### Source code

```
package models;
public class Question {
    public final String question;
    public final String optionA;
    public final String optionB;
    public final String optionC;
    public final String optionD;
    public final String answer;
 public Question(String question, String optionA, String optionB, String optionC, String
optionD, String answer) {
        this.question = question;
        this.optionA = optionA;
        this.optionB = optionB;
        this.optionC = optionC;
        this.optionD = optionD;
        this.answer = answer;
    }
}
package quizapp;
import quizapp.ui.Login;
public class Main {
    public Main() {
    }
    public static void main(String[] args) {
        (new Login()).setVisible(true);
    }
package repositories;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import services.AdminService;
import services.JsonFileService;
public class AdminRepository implements AdminService {
    privatefinalJsonFileService jfs = new JsonFileRepository("resources/databases/users.json");
    public AdminRepository() {
    }
    public boolean getLogin(String username, String password) {
        return username.equals("admin") && password.equals("admin");
    }
    public boolean createStudent(String username, String password) {
        try {
            Map<String, String> map = new HashMap();
            map.put("username", username);
            map.put("password", password);
            map.put("role", "student");
            map.put("marks", "0");
            boolean isAdded = this.jfs.createItemsInJsonFile(map);
            if (!isAdded) {
                System.out.println("Student Addition Failed");
            }

            return true;
        } catch (Exception var5) {
            System.out.println("Student Addition Failed. Exception occured");
            System.out.println(var5.getMessage());
            var5.printStackTrace();
            return false;
        }
    }
    public List<Map<String, String>> getStudentMarks() {
        try {
            List<String> keys = new ArrayList();
            keys.add("username");
            keys.add("marks");
            List<Map<String, String>> marks = this.jfs.getAllRow(keys);
            if (!marks.isEmpty()) {
                return marks;
            } else {
                System.out.println("No Student marks found");
                return null;
            }
        } catch (Exception var3) {
            System.out.println("Student marks retrieval Failed. Exception occured");
var3.printStackTrace();
            return null;
        }
    }
}
package services;
import java.util.List;
import java.util.Map;

public interface AdminService {
   boolean getLogin(String var1, String var2);

   boolean createStudent(String var1, String var2);

   List<Map<String, String>> getStudentMarks();
}
package services;
import java.util.List;
import java.util.Map;
public interface JsonFileService {
   boolean createItemsInJsonFile(Map<String, String> var1);
   List<Map<String, String>> getAllRow(List<String> var1);
   List<Map<String, String>> getRandomRows(List<String> var1, int var2);
   List<Map<String, String>> getAllRowByCondition(List<String> var1, Map<String, String> var2);
   boolean updateAllRowByCondition(List<String> var1, Map<String, String> var2, Map<String, String> var3);
}
package services;
import java.util.List;
import java.util.Map;
import models.Question;
public interface QuestionService {
   boolean addQuestion(Question var1);
List<Map<String, String>> getRandomQuestions(int var1);
}
package services;
import java.util.List;
import java.util.Map;
public interface StudentService {
   boolean getLogin(String var1, String var2);
```

```
    boolean addStudent(String var1, String var2);
    List<Map<String, String>> getAllStudent();
    boolean updateStudentMarks(String var1, String var2);
}
```

[{"question":"The cost incurred in first time reviews and testing is called ---------","answer":"c","optionC":"Blue money","optionD":"","optionA":"Red money","optionB":"Green money"},{"question":"Which of the following is not a core step of Six Sigma?","answer":"b","optionC":"Measure","optionD":"Define","optionA":"Analyse","optionB":"Control"},{"question":"What kind of quality cost is incurred when an error is detected in a product prior to shipment?","answer":"b","optionC":"External Failure","optionD":"Appraisal","optionA":"Prevention","optionB":"Internal Failure"},{"question":"Which of the following is not included in External failure costs?","answer":"a","optionC":"warrantwork","optionD":"complaint resolution","optionA":"testing","optionB":"help line support"},{"question":"Where the Alpha testing is done at\u2026","answer":"a","optionC":"User\u2019s end","optionD":"None of above","optionA":"Developer\u2019s end","optionB":"Developer\u2019s & User\u2019s end"},{"question":"Which of the following belong to the testing levels?","answer":"d","optionC":"Integration Testing","optionD":"All of above","optionA":"Unit Testing","optionB":"System Testing"},{"question":"To achieve the software Quality is easily adaptable only with programming skills?","answer":"b","optionC":"","optionD":"","optionA":"True","optionB":"False"},{"question":"Is the 100% Quality of a software is achieveable.","answer":"b","optionC":"depent on tester","optionD":"envirenment factor involve","optionA":"YES","optionB":"NO"},{"question":" -------- is not a Test Document.","answer":"a","optionC":"Test Policy","optionD":"Test Case","optionA":"PIN ( Project Initiation Note)","optionB":"RTM (requirement Traceability matrix)"},{"question":"SDLC stands for ________","answer":"a","optionC":"System development life cycle","optionD":"Software design life cycle","optionA":"Software development life cycle","optionB":"System design life cycle"},{"question":"When the Testing Principles are useful while building the Software Product?","answer":"d","optionC":"During testing","optionD":"Throughout life-cycle","optionA":"During execution","optionB":"During review"},{"question":"Behavioral testing is","answer":"b","optionC":"Grey box testing","optionD":"None of the mentioned","optionA":"White box testing","optionB":"Black box testing"},{"question":"Acceptance testing is also known as","answer":"d","optionC":"Grey box testing","optionD":"Beta testing","optionA":"Alpha Testing","optionB":"White box testing"},{"question":"Beta testing is done at","answer":"a","optionC":"User\u201Fs & Developer\u201Fs end","optionD":"Developer\u201Fs end","optionA":"User\u201Fs end","optionB":"None of the mentioned"},{"question":"Which testing is done by developer?","answer":"a","optionC":"Sanity Testing","optionD":"Regression Testing","optionA":"Unit Testing","optionB":"Integration Testing"},{"question":"SQA stands for:","answer":"a","optionC":"Simple Quotua Analysic","optionD":"None of the above","optionA":"Software Quality Assurance","optionB":"SQL Analysic"}]

{"password":"2","role":"student","marks":"3","username":"b"},{"password":"1","role":"student","marks":"3","username":"a"},{"password":"rr","role":"student","marks":"3","username":"rr"},{"password":"leena","role":"student","marks":"0","username":"leena"}]

```
package models;
public class Question {
   public final String question;
   public final String optionA;
   public final String optionB;
   public final String optionC;
   public final String optionD;
   public final String answer;
   public Question(String question, String optionA, String optionB, String optionC, String optionD, String answer) {
      this.question = question;
      this.optionA = optionA;
      this.optionB = optionB;
      this.optionC = optionC;
      this.optionD = optionD;
      this.answer = answer;
   }
}
//      a.add("1");
//       a.add("2");
//       a.add("3");
//        a.add("4");
//        j.getRandomRows(0, a);
    }

}
package repositories;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Random;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
import services.JsonFileService;
public class JsonFileRepository implements JsonFileService {
    private final String fileName;
    public JsonFileRepository(String fileName) {
        this.fileName = fileName;
    }
    private boolean isJsonFileNameValid() {
        boolean doesJsonExtensionExists = fileName.contains(".json");
        if (doesJsonExtensionExists) {
            return true;
        }
        System.out.println(".json extension is not present in file name");
        return false;
    }
    private boolean createJsonFileIfDoesnNotExist() {
        try {
            // check if file name is valid
            boolean isfileNameValid = isJsonFileNameValid();
            if (!isfileNameValid) {
                return false;
            }
            File = new File(fileName);
            // if file doesn't exist
            if (!file.exists()) {
                // create the file
                boolean isFileCreated = file.createNewFile();
                // if file creation failed, exit
                if (!isFileCreated) {
                    System.out.println("File creation failed");
                    return false;
                }
                // if file creation successful
                // initialize it with a blank json array
                FileWriter fr = new FileWriter(file);
                fr.write("[]");
                fr.close();
            }
FileWriter fw = new FileWriter(fileName);
            fw.write(jsonArray.toJSONString());
            fw.flush();
            fw.close();
            parser.reset();
            fr.close();
            return true;
        } catch (Exception ex) {
            System.out.println("row update failed");
            System.out.println(ex.getMessage());
            ex.printStackTrace();
            return false;
        }
    }
}
package repositories;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import models.Question;
import services.JsonFileService;
import services.QuestionService;
public class QuestionRepository implements QuestionService {
    private final JsonFileService jfs;
    public QuestionRepository() {
        jfs = new JsonFileRepository("resources/databases/questions.json");
    }
    @Override
    public boolean addQuestion(Question question) {
        try {
            Map<String, String> map = new HashMap<>();
            map.put("question", question.question);
            map.put("optionA", question.optionA);
            map.put("optionB", question.optionB);
            map.put("optionC", question.optionC);
            map.put("optionD", question.optionD);
            map.put("answer", question.answer);
            boolean isAdded = jfs.createItemsInJsonFile(map);
            if (isAdded == false) {
                System.out.println("Question Addition Failed");
            }
            return true;
        } catch (Exception ex) {
            System.out.println("Question Addition Failed. Exception occured");
            System.out.println(ex.getMessage());
            ex.printStackTrace();
            return false;
        }
    }
    public List<Map<String, String>> getRandomQuestions(int numberOfQuestions) {
        try {
            List<String> keys = new ArrayList<>();
            keys.add("question");
            keys.add("optionA");
            keys.add("optionB");
            keys.add("optionC");
            keys.add("optionD");
            keys.add("answer");
            List<Map<String, String>> questions = jfs.getRandomRows(keys, numberOfQuestions);
            if (!questions.isEmpty()) {
                return questions;
            }
            return null;
        } catch (Exception ex) {
            System.out.println("Random Question Retrival Failed. Exception occured");
            System.out.println(ex.getMessage());
            ex.printStackTrace();
            return null;
        }
    }
}
package services;
import java.util.List;
import java.util.Map;
import models.Question;
public interface QuestionService {
    public boolean addQuestion(Question question);
    public List<Map<String, String>> getRandomQuestions(int numberOfQuestions);
}
<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
  <component name="NewModuleRootManager" inherit-compiler-output="true">
    <exclude-output />
    <content url="file://$MODULE_DIR$">
      <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
    </content>
    <orderEntry type="inheritedJdk" />
    <orderEntry type="sourceFolder" forTests="false" />
    <orderEntry type="library" name="json-simple-1.1.1" level="project"   </component>
</module>
package repositories;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import services.JsonFileService;
import services.StudentService;

public class StudentRepository implements StudentService {
    privatefinalJsonFileService jfs = new JsonFileRepository("resources/databases/users.json");
      public StudentRepository() {
    }
    public boolean getLogin(final String username, final String password) {
        List<String> keys = Arrays.asList("username", "password");
        Map<String, String> conditions = new HashMap<String, String>() {
            {
                this.put("username", username);
                this.put("password", password);
            }
        };
        List<Map<String, String>> students = this.jfs.getAllRowByCondition(keys, conditions);
        if (!students.isEmpty() && students.size() >= 2) {
            Map<String, String> usernameMap = new HashMap<String, String>() {
                {
                    this.put("username", username);
                }
            };
            Map<String, String> passwordMap = new HashMap<String, String>() {
                {
                    this.put("password", password);
                }
            };
            if (students.contains(usernameMap) && students.contains(passwordMap)) {
                return true;
            }
        }

        return false;
    }

    public boolean addStudent(String username, String password) {
        try {
            Map<String, String> map = new HashMap();
            map.put("username", username);
            map.put("password", password);
            map.put("marks", "");
            map.put("role", "student");
            return true;
        } catch (Exception var4) {
            System.out.println("Student Addition Failed. Exception occured");
            System.out.println(var4.getMessage());
            var4.printStackTrace();
            return false;
        }
    }

    public List<Map<String, String>> getAllStudent() {
        try {
            new ArrayList();
            List<String> keys = new ArrayList();
            keys.add("username");
            keys.add("marks");
            List<Map<String, String>> list = this.jfs.getAllRow(keys);
            if (list.isEmpty()) {
                System.out.println("No student found");
                return null;
            } else {
                return list;
            }
        } catch (Exception var3) {
            System.out.println("Student Retrival Failed. Exception occured");
            System.out.println(var3.getMessage());
            var3.printStackTrace();
            return null;
        }
    }

    public boolean updateStudentMarks(String username, String marks) {
        try {
            List<String> keys = new ArrayList();
            keys.add("username");
            keys.add("marks");
            Map<String, String> conditions = new HashMap();
            conditions.put("username", username);
            Map<String, String> updatedValues = new HashMap();
            updatedValues.put("marks", marks);
            booleanisUpdated=this.jfs.updateAllRowByCondition(keys,conditions,
updatedValues);
            if (isUpdated) {
                System.out.println("Student's marks updated");
                return true;
            } else {
                return false;
            }
        } catch (Exception var7) {
            System.out.println("Student's marks update Failed. Exception occured");
            System.out.println(var7.getMessage());
            var7.printStackTrace();
            return false;
        }
    }
}
```

## Screenshot of Application

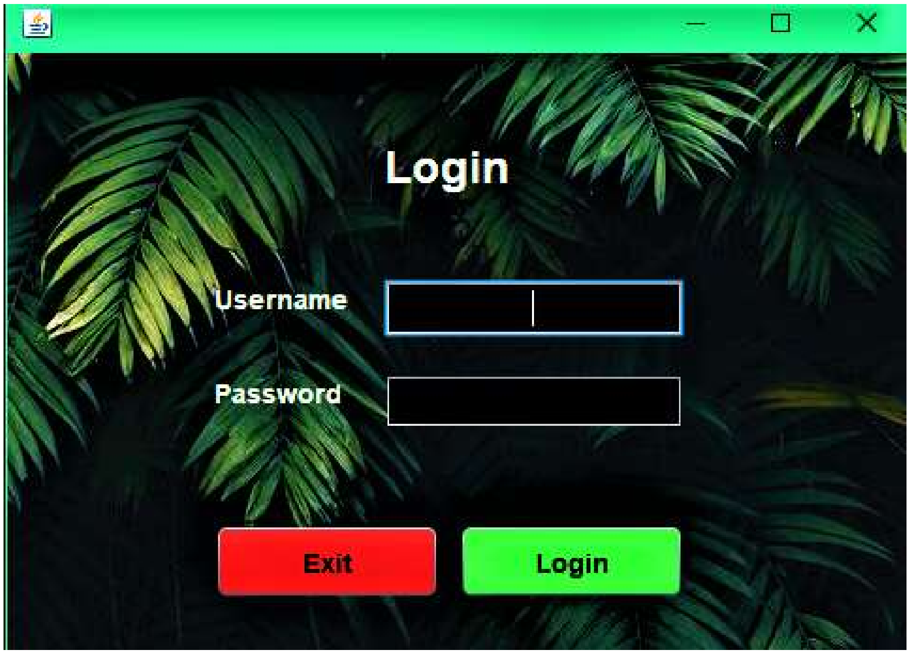


Fig 5.2 Login page

The credentials given by the admin are to be entered in login page to access the application.

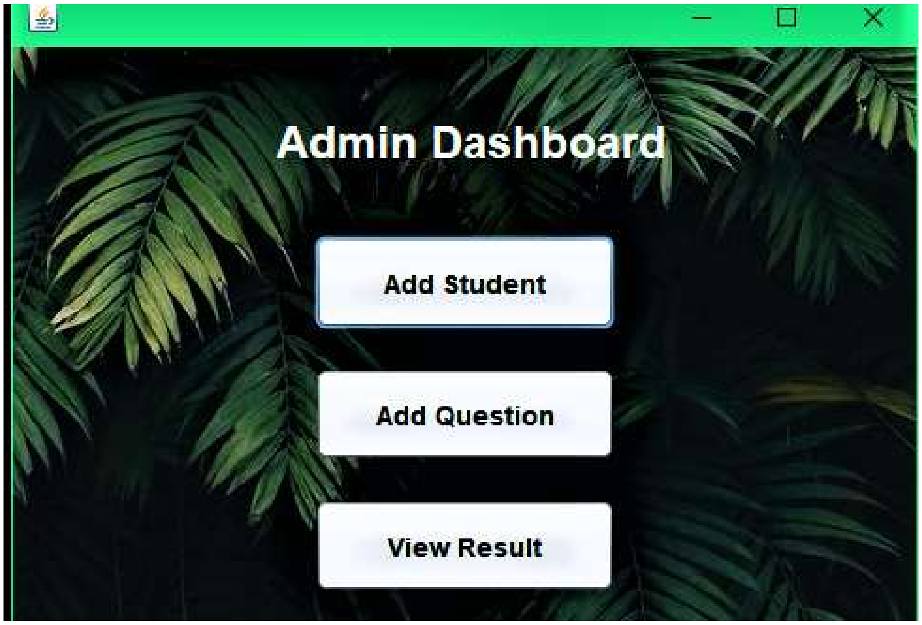
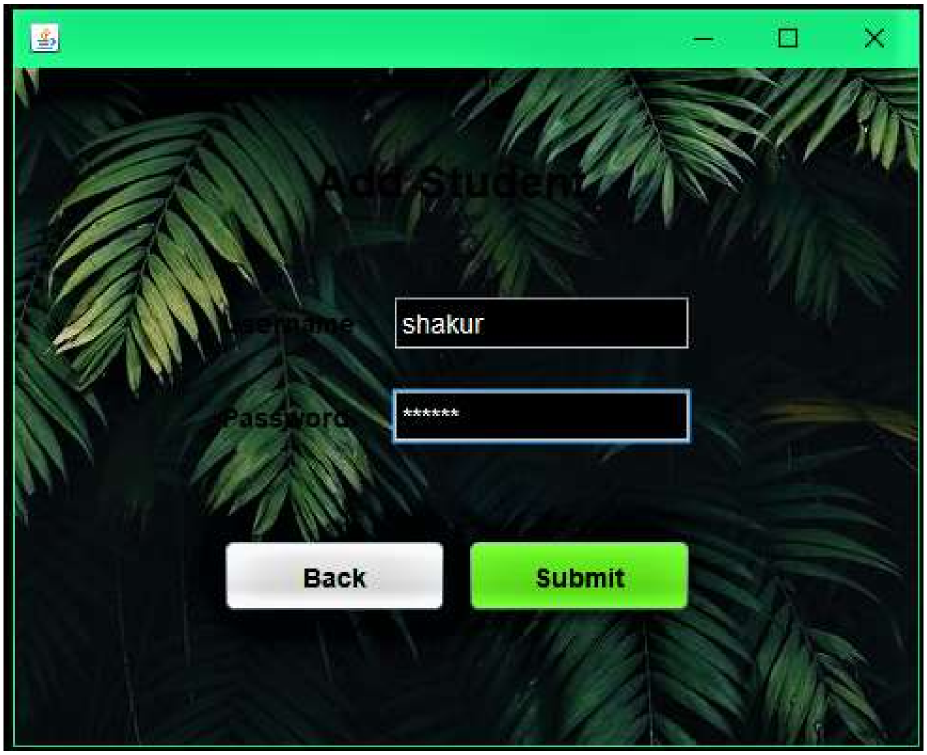


Fig 5.3 Admin Dashboard

If the admin is accessing the app using the provided credentials it directs to this page….

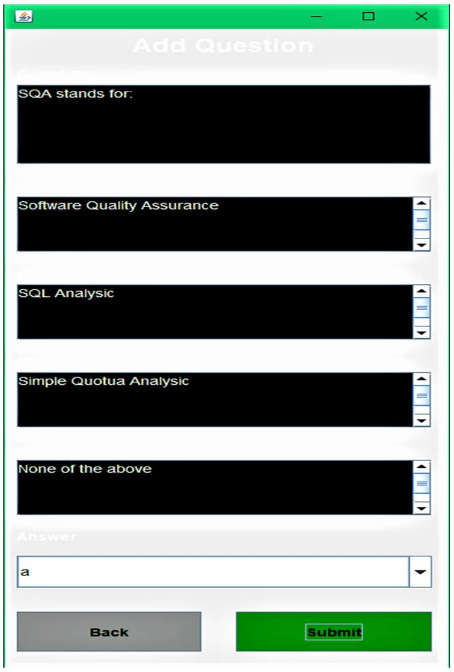


shakur

******

5.3 Adding a student

The admin has the access to add a student using the “Add the student” option provided



SQA stands for:

Software Quality Assurance

SQL Analysic

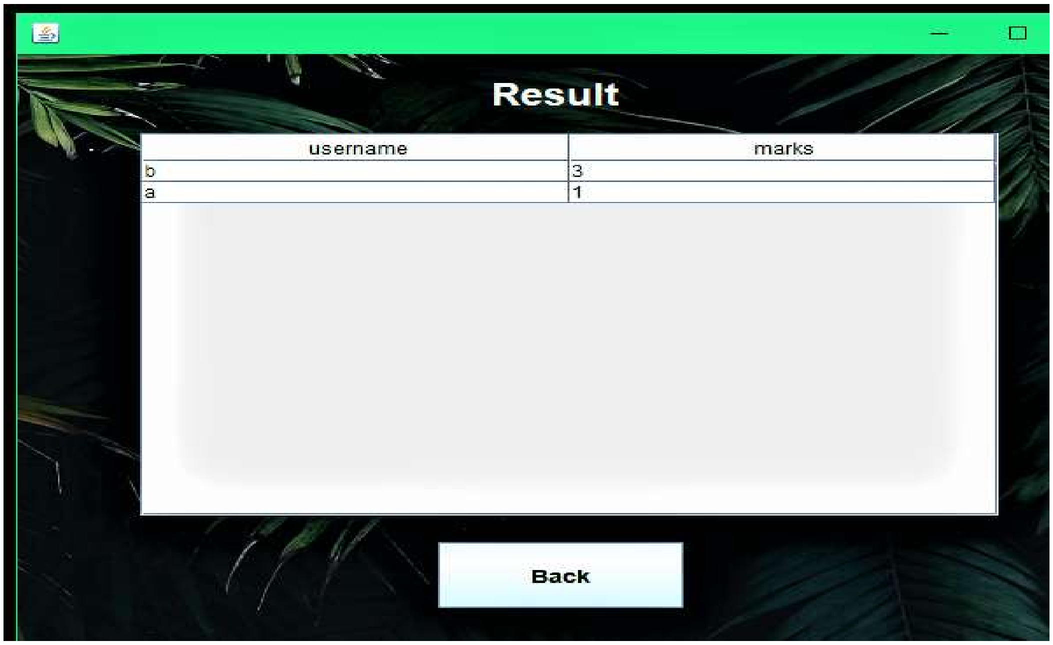
Simple Quotua Analysic

None of the above

a

5.4 Quiz questions

The admin can add the questions into the quiz “Add question” option provided.

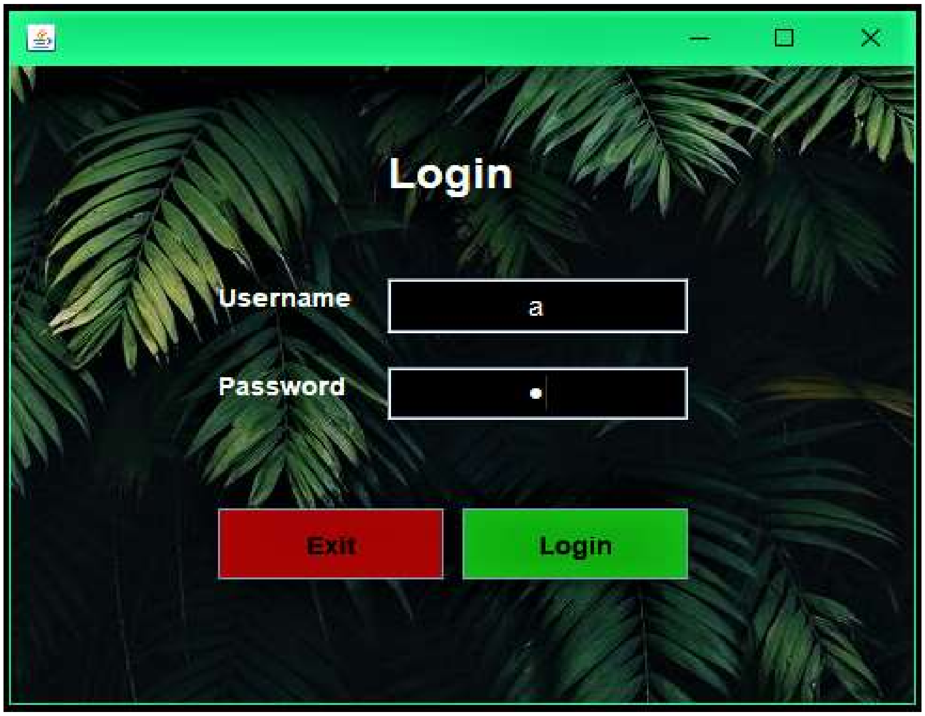


Result

| username | marks |
|---|---|
| b | 3 |
| a | 1 |

5.5 Result

The results are displayed after the quiz is attempted.This result can view by the admin



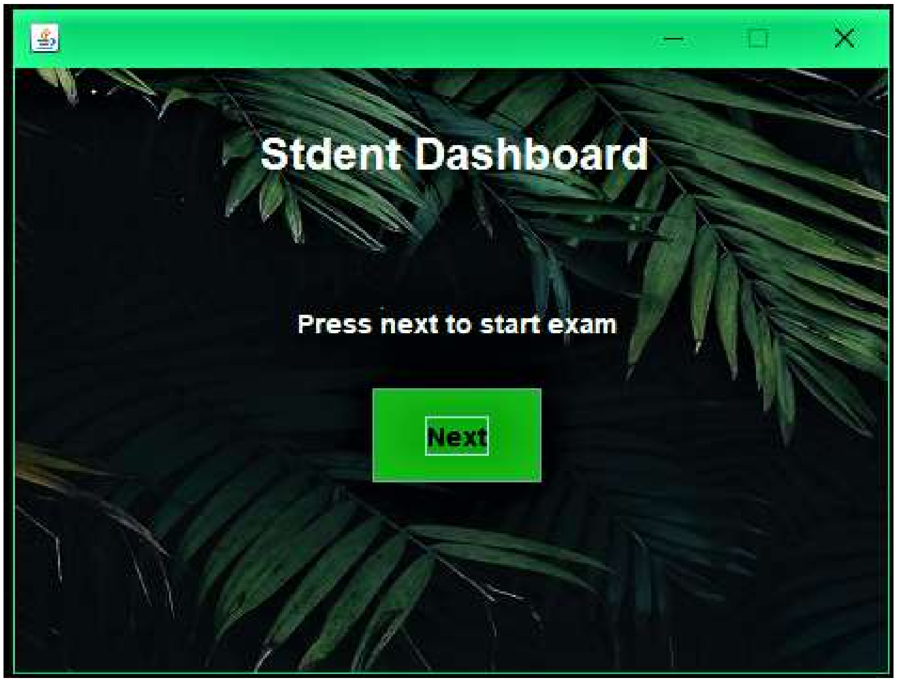
Login

Username: a

Password: •

5.6 Login page

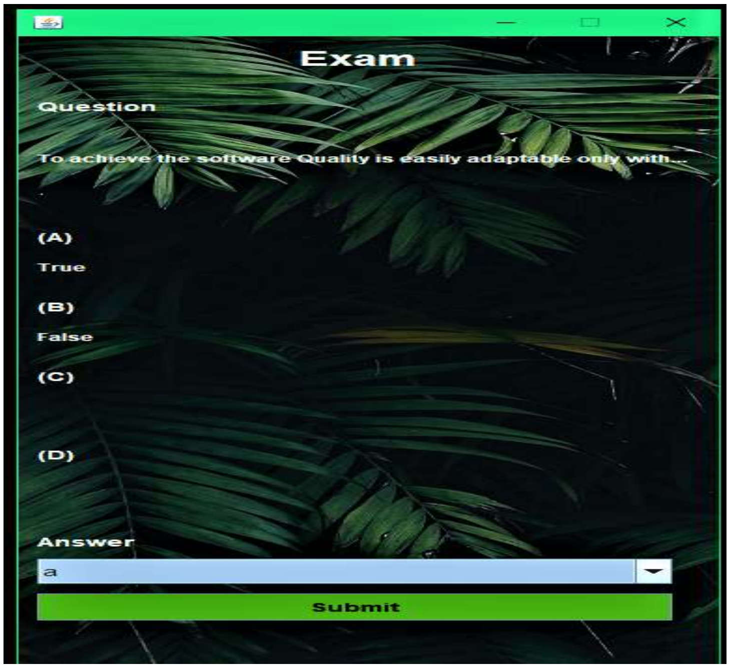
Using the page the users can access the app using the credentials provided by the admin


Stdent Dashboard

Press next to start exam

5.7 Student Dashboard

By clicking on next option we move into main app


Exam

Question

To achieve the software Quality is easily adaptable only with...

(A)

True

(B)

False

(C)

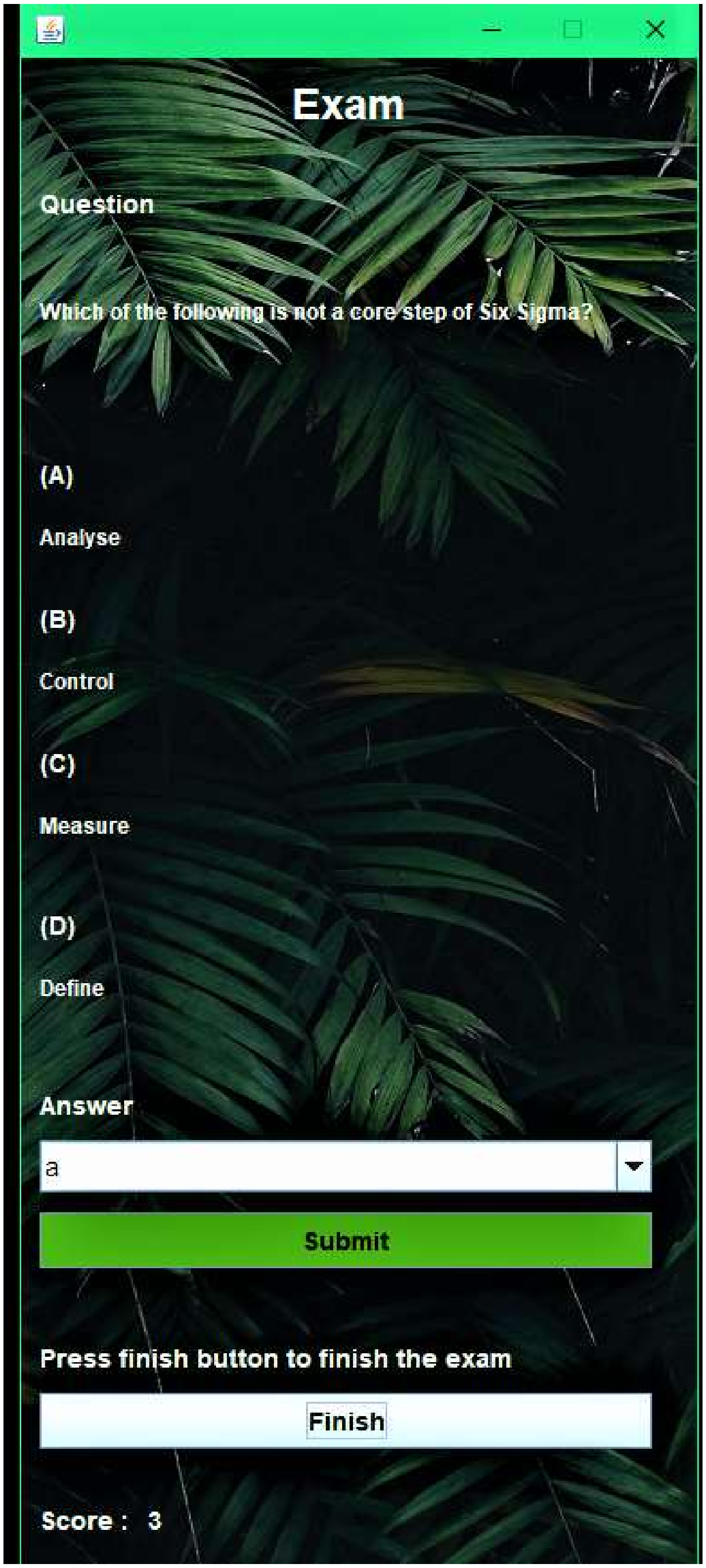
(D)

Answer

a

5.8 Submission of the answers

The questions are answered by users using the options bar provided



Exam

Question

Which of the following is not a core step of Six Sigma?

(A)

Analyse

(B)

Control

(C)

Measure

(D)

Define

Answer

a

Press finish button to finish the exam

Score : 3

5.9 Answering the questions

The answers for the questions provided are submitted and the result is displayed

# 6.RESULT & DISCUSSION

## 6.1 Result

The results for Mind Matrix  can vary based on several factors, including the platform.

| Criteria | Existing System | Proposed System | Percentage Difference |
|---|---|---|---|
| **Performance** | **5** | **9** | **80%** |
| **Functionality** | **7** | **8** | **14.3%** |
| **User Interface** | **4** | **9** | **125%** |
| **Security** | **6** | **8** | **33.3%** |
| **Scalability** | **5** | **10** | **100%** |
| **Average** | **-** | **-** | **70.92%** |

Table 6.1 Comparision

## 6.2 Discussions

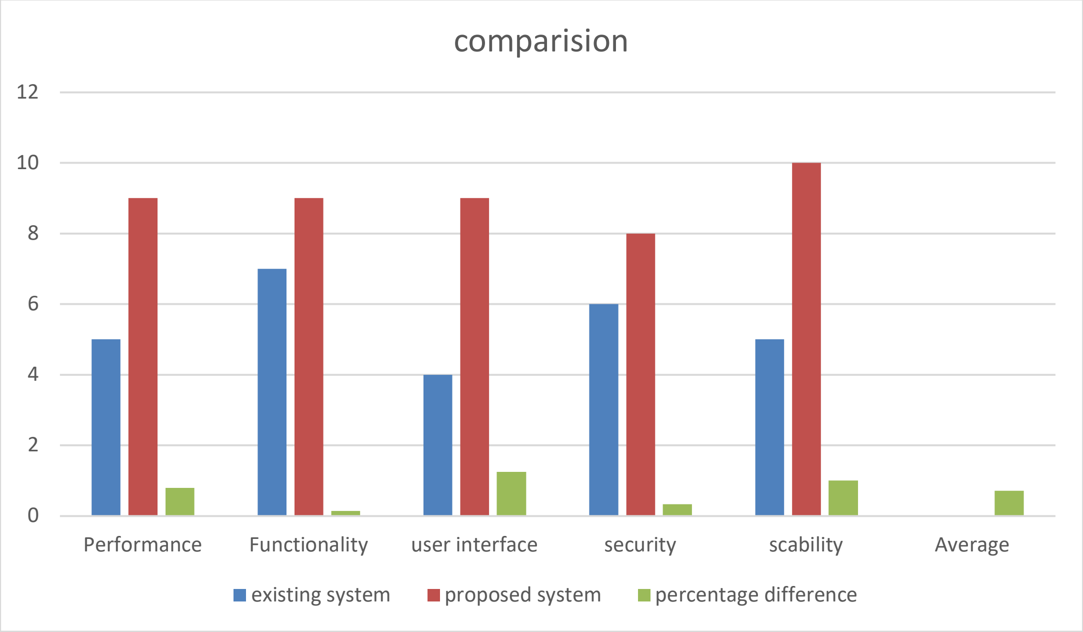


Fig no:6.2 Bar graph for the results obtained

# 7.CONCLUSION & FUTURE ENHANCEMENTS

## 7.1 Conclusion

In conclusion, online quizzes have become an integral part of modern education and assessment methods. They offer numerous advantages such as flexibility, accessibility, and immediate feedback, enhancing the overall learning experience for students. The digital format allows for a wide range of question types and interactive elements, promoting engagement and active participation.

While online quizzes provide educators with efficient tools for evaluating students' knowledge and understanding, it is essential to recognize the importance of well-designed assessments that align with learning objectives. Striking a balance between formative and summative assessments ensures a comprehensive evaluation of students' progress and encourages continuous improvement.

Furthermore, the adaptability of online quizzes caters to diverse learning styles and preferences, accommodating students with varying needs. The integration of technology not only facilitates the assessment process but also prepares students for the digital demands of the contemporary workforce.

As online learning continues to evolve, it is crucial to address challenges such as the potential for cheating and the need for equitable access to technology. Implementing robust security measures and considering inclusivity in assessment design will contribute to the credibility and fairness of online quizzes.

In essence, online quizzes serve as valuable tools for both educators and students, fostering a dynamic and responsive educational environment. As technology advances, further innovations in assessment methods are likely to emerge, shaping the future landscape of online learning and evaluation

## 7.2 Future enhancements:

### 1. Personalized Learning:

• Advances in artificial intelligence (AI) and machine learning will enable the creation of more personalized learning experiences. Adaptive quizzes and learning platforms can tailor content to individual needs, pacing, and learning styles.

## 2. Gamification

• Gamified elements, such as points, badges, and leaderboards, can enhance user engagement and motivation. Integrating game-like features into online quizzes and learning modules can make the experience more enjoyable and effective.

## 3._Virtual and Augmented Reality

• The integration of virtual and augmented reality technologies can provide immersive learning experiences. Imagine students taking quizzes in a virtual classroom or exploring historical events through augmented reality simulations.

## 4. Blockchain for Credentialing

• Blockchain technology can be used to secure and verify educational credentials. This ensures the integrity of online quiz results and certifications, making them more trustworthy in professional and academic settings.

## 5. Collaborative Learning

• Online quizzes and learning platforms will likely incorporate more collaborative features, allowing students to work together in real-time, share insights, and learn from one another, irrespective of geographical locations.

## 6. Mobile Learning

• With the increasing use of smartphones and tablets, mobile learning will continue to grow. Online quizzes and learning materials optimized for mobile devices provide learners with flexibility and accessibility.

## 7.Rich Question Types

Expand the range of question types beyond multiple-choice, including interactive drag-and-drop, matching, and scenario-based questions to assess a broader set of skills.

## 8.Personalized Recommendations

Offer personalized recommendations for additional study materials or courses based on quiz performance, helping users fill gaps in their knowledge.

# REFERENCES

1. Rashid A, Zubair MZ. An Intelligent Agent for a Vacuum Cleaner.
2. https://play.google.com/store?hl=en/ last accessed, 24 Nov 2015 [11] Saqib SM,Asghar MZ, Ahmad S, Ahmad B, Jan MA.“Framework for Customized-SOA Projects” .International Journal of Computer Science and Information Security. 2011 May1;9(5):240.
3. Asghar D, Zubair M, Asghar MJ. “Expert System For Online Diagnosis of Red-Eye Diseases”. International Journal of Computer Science & Emerging Technologies (IJCSET).2010;1(2):35-9
4. Saqib SM, Jan MA, Ahmad B, Ahmad S,Asghar MZ. “Custom Software under the Shade of Cloud Computing”. International Journal of Computer Science and Information Security. 2011 May 1;9(5):219
5. Hussain S, Asghar MZ, Ahmad B, Ahmad “A Step towards Software Corrective Maintenance Using RCM model”. ArXiv preprint arXiv:0909.0732. 2009 Sep 3.
6. Rashid A, Zubair MZ. An Intelligent Agent for a Vacuum Cleaner. International Journal of Digital Content Technology and its Applications. 2009;3(2):143-6.
7. Asghar MZ, Ahmad S, Marwat A, Kundi FM. “Sentiment Analysis on YouTube: A Brief Survey”. arXiv preprint arXiv:1511.09142.2015 Nov 30
8. James Steele, Nelson To, Shan Conder, Lauren Darcey, the Android Developer's Collection.
9. Jerome (J. F.) DiMarzio, Android – A Programmer’s guide, Tata McGraw-Hill Education.
10. Mark L. Murphy, The busy coder's guide to Android development , Commons Ware.