JAVA PROJECT

Submitted by,

Lohith kumar -70

Arya Bhaskaran A- 67

Pooja P Bhaskar -72

Nihal Ali K -48

INTRODUCTION

The Quiz Application is an interactive Java application designed to make learning fun and engaging. It allows users to play a quiz by answering multiple-choice questions and also gives them the option to create their own quiz questions and answers. This feature makes the game more flexible and personalized, as users can design quizzes based on any topic of their choice.

The main goal of this project is to provide an easy way for students and learners to test their knowledge while practicing programming concepts. The application uses basic Java features such as loops, arrays, conditional statements, and user input handling to display questions, accept answers, and calculate scores.

This project not only helps users improve their general knowledge but also assists beginners in understanding how Java can be used to build interactive console-based applications. With its simple design and customizable question feature, the Simple Quiz Game serves as a fun learning tool for both students and new programmers.

OBJECTIVES:

- To design a simple and user-friendly quiz game using Java.
- To enable users to add their own questions and answers to the quiz.To apply fundamental Java concepts such as classes, arrays, loops, and conditional statements.
- To display questions, collect user responses, and provide instant feedback.Storing the questions into mysql database.

- To calculate and display the user’s total score at the end of the quiz.To promote learning and creativity by allowing users to design quizzes on topics of their interest.

ABSTRACT

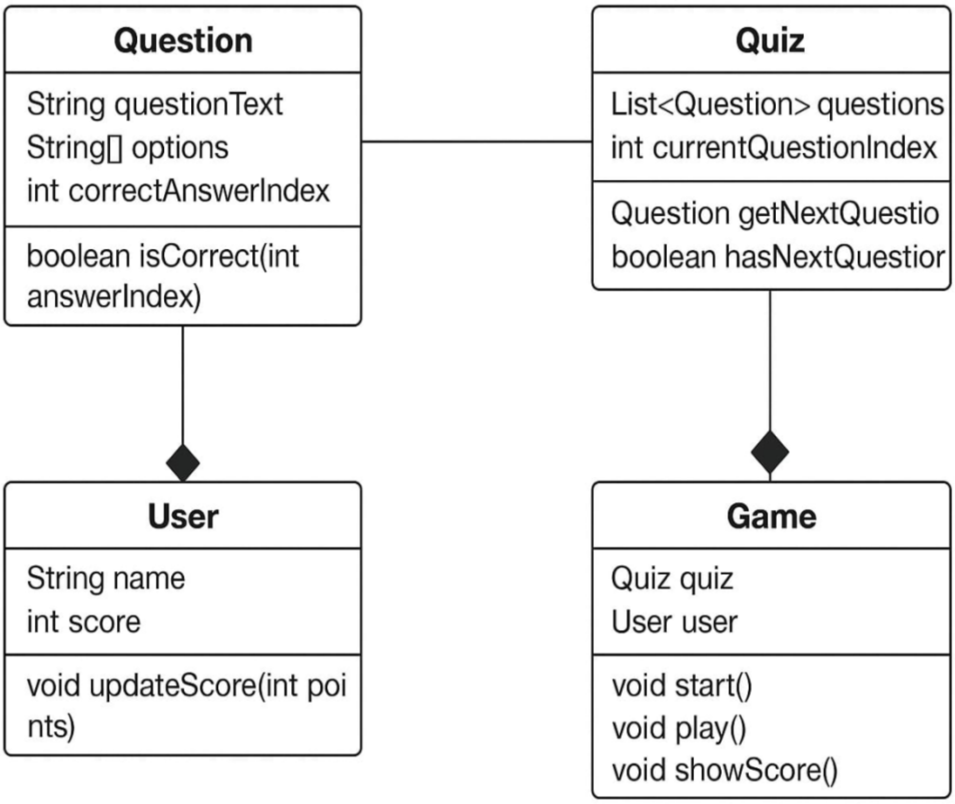
The Quiz Aplication is an interactive Java-based application designed to provide an engaging and educational experience for students and learners. The game presents a series of multiple-choice questions to the user, collects responses, evaluates answers, and calculates the final score. A unique feature of this project is that users can create their own quiz questions and answers, allowing the quiz to be customized according to individual preferences, subjects, or learning objectives.

The project demonstrates the practical application of fundamental Java programming concepts such as classes, arrays, loops, conditional statements, and user input handling. It offers a simple and user-friendly interface, making it accessible for beginners and students. This project not only reinforces knowledge in various subjects but also enhances logical thinking, problem-solving skills, and familiarity with programming logic. The Simple Quiz application can be extended in the future with features like timers, difficulty levels, and score tracking, making it a versatile educational tool.

This project not only helps users improve their general knowledge but also assists beginners in understanding how Java can be used to build interactive console-based applications. With its simple design and customizable question feature, the Simple Quiz application serves as a fun learning tool for both students and new programmers.

SYSTEM DESIGN

➢UML DESIGN



➢ USE CASES:

1.select subject:User chooses a subject for the quiz.

2.Set target of marks:User decides how many marks are needed to pass/finish the quiz.

3.Add question:User can add their own questions and ansers to the quiz databases.

4. Start quiz:Begins the quiz session.

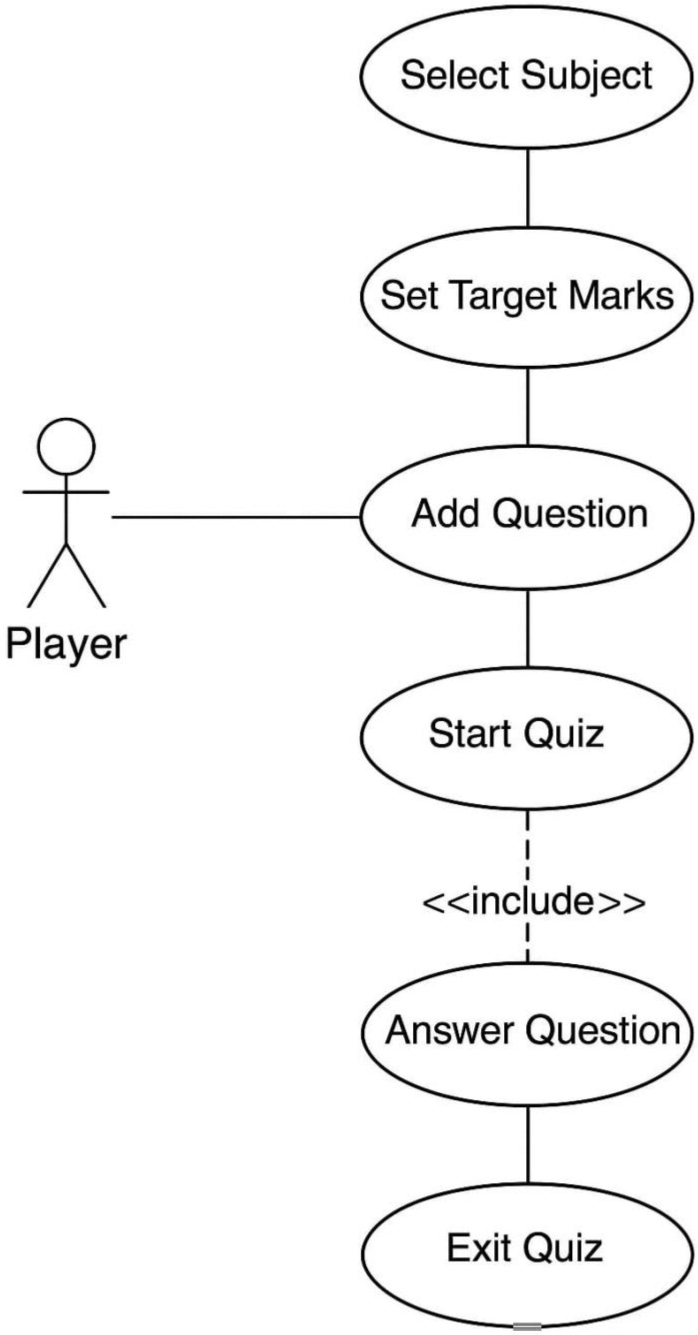
5.Answer question:User answers each question.

only proceeds to the next question if the answer is correct.

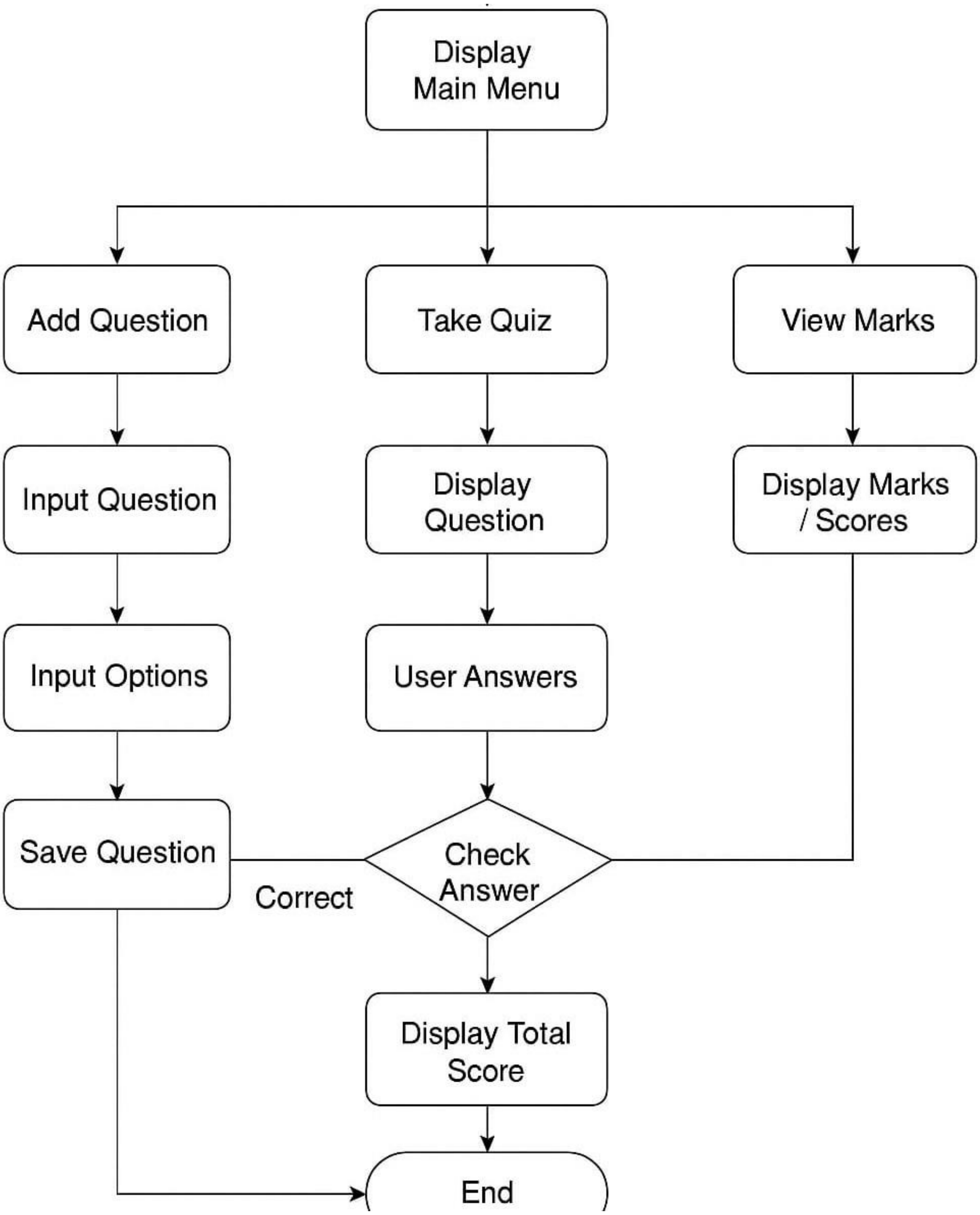
6.View marks:User can see their current score/marks.

7.Exit quiz:User can quit anytime.

- **USECASE DIAGRAM:**



➢ Activity Diagram



➢ Quiz Game Algorithm

Step 1: Initialize the Game
Create empty data structures for:
questions_list → store questions
answers_list → store correct answers
user_answers_list → store user responses
wrong_answers_list → store questions answered incorrectly
Connect to MySQL database for storing questions (optional).
Step 2: Choose Quiz Category
Display available categories (e.g., Math, Science, History).
Input category_choice from user.
Filter questions based on selected category if retrieving from database.
Step 3: Input Number of Questions
Ask user: "How many questions do you want to attempt?"
Input num_questions.
Validate input: it must be a positive integer and ≤ total available questions.
Step 4: Create Your Own Questions (Optional)
Ask user if they want to add new questions.
If yes:
Input question text.
Input correct answer.
Assign to the chosen category.
Store into MySQL database for future use.
Step 5: Quiz Loop
For i = 1 to num_questions:
1. Display question i.
2. Input user_answer.
3. Validate input (non-empty, matches expected type).

4. Compare with correct_answer:

If correct:

Add to right_answers_list.

Increment score.

Display “Correct!”

If incorrect:

Add to wrong_answers_list.

Display “Wrong!”

Step 6: Display Results

Print total score 🏆.

Show all correct answers

Show all wrong answers

Step 7: Optional Features

Store questions/answers in MySQL 💾:

INSERT new questions into the quiz_questions table.

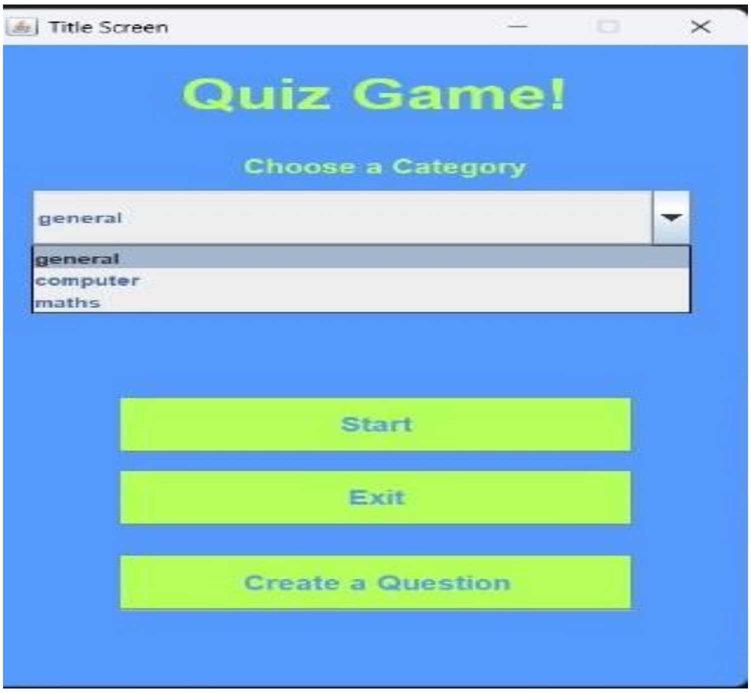
Validate input :

Ensure question text is not empty.

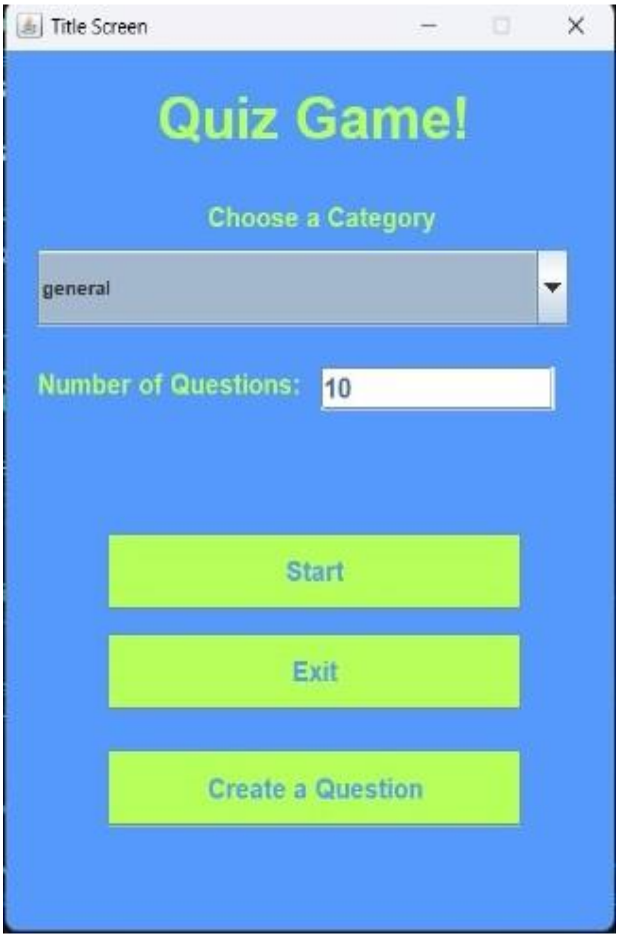
Ensure answer is not empty.

Ensure number of questions selected ≤ available questions.

➢ SCREENSHOTS

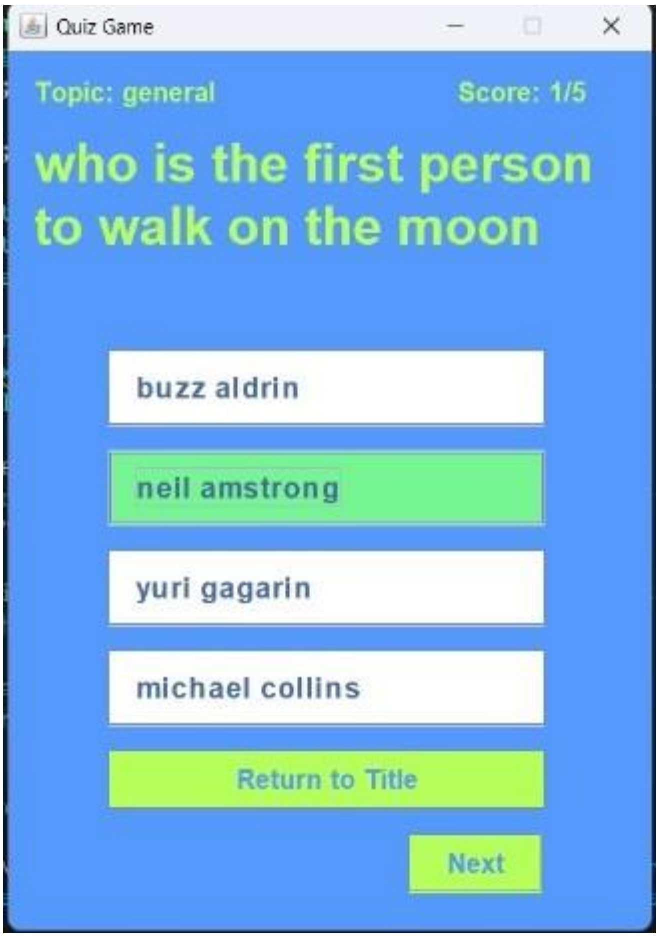


Choose a Category: general



Choose a Category: general

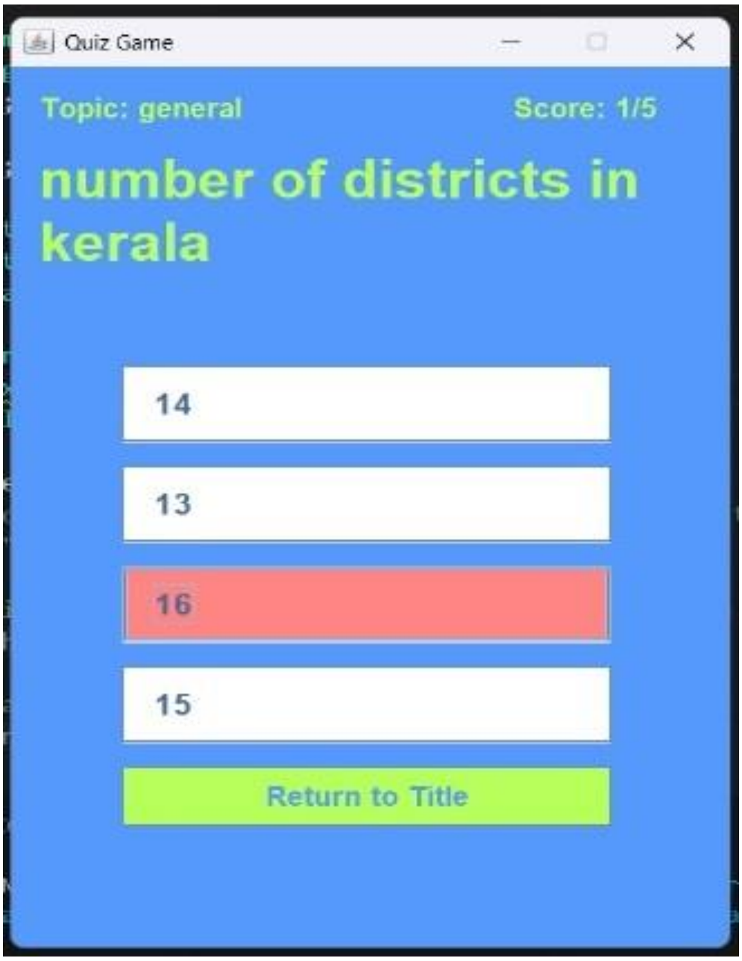
Number of Questions: 10



Topic: general

Score: 1/5

who is the first person to walk on the moon

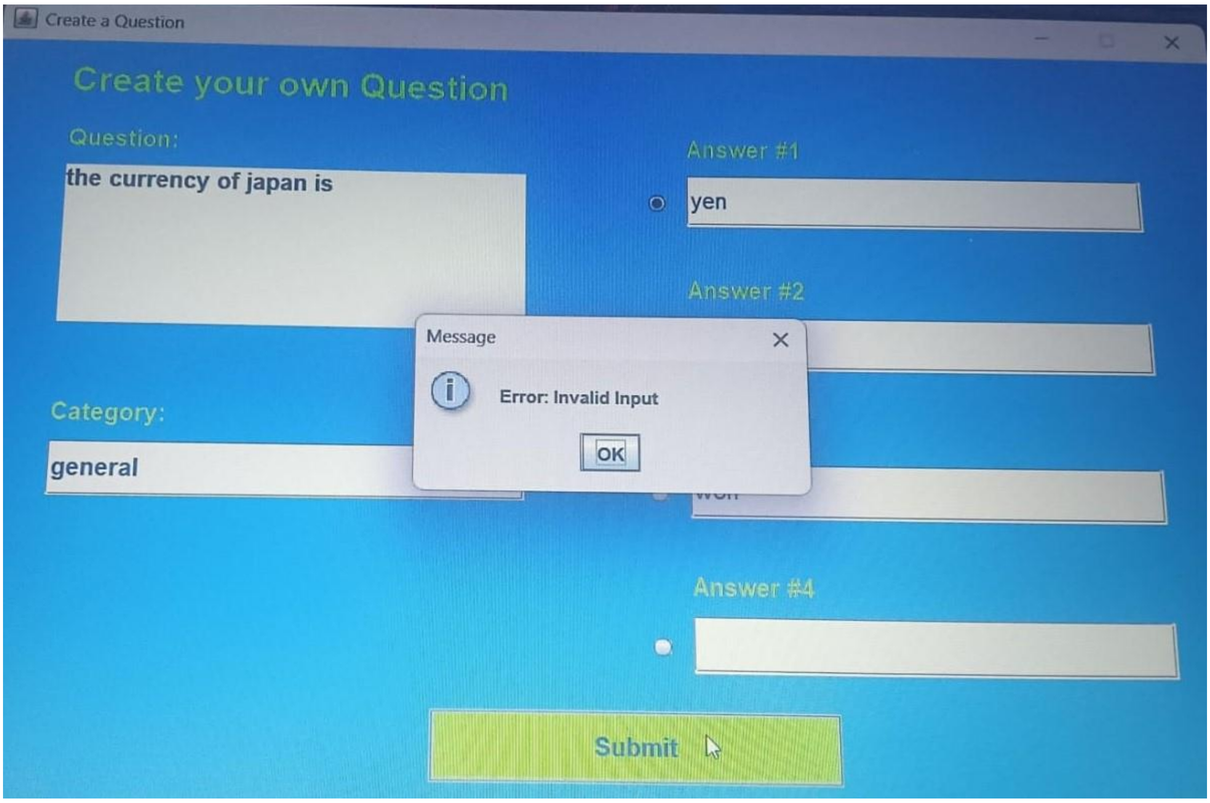


Quiz Game

Topic: general

Score: 1/5

number of districts in kerala



Question: the currency of japan is

Answer #1: ● yen

Category: general

won

Error: Invalid Input

CONCLUSION

The **Quiz Application** project demonstrates how Java can be used to develop an interactive and educational application. It provides a platform for students and beginners to test their knowledge, practice problem-solving, and enhance logical thinking. The added feature of allowing users to **create their own questions and answers** makes the game flexible, personalized, and engaging.

Through this project, fundamental Java concepts such as classes**,** objects, arrays, loops, conditional statements, and user input handling are applied practically, helping learners understand their real-world usage. The game is simple, user-friendly, and can easily be extended with advanced features like timers, difficulty levels, categories, and score tracking.

Overall, this project successfully combines education and entertainment, serving as a foundation for more advanced quiz-based applications while improving programming skills and knowledge retention among students.