

**PROJECT REPORT**

**PROJECT TITLE: MATH QUIZ APPLICATION**

**PRESENTED BY: Sagar Dhondge**

**INTERN ID: VN-JD-4W289**

**Math Quiz App Report**

# 1. Introduction

The Math Quiz App is a desktop application designed to help users practice and improve their math skills. The app offers a variety of math questions, tracks user progress, and stores user data in a MySQL database.

# 2. Technologies Used

* **Java**: The core programming language used for the application logic.
* **JavaFX**: Used for creating the graphical user interface (GUI).
* **MySQL**: A relational database management system to store user data and quiz results.

# 3. Application Features

* **Quiz Categories**: Various categories such as Addition, Subtraction, Multiplication, and Division.
* **Question Generation**: Randomly generated questions based on the selected category and difficulty level.
* **Score Tracking**: Keeps track of the user's score for each quiz session.
* **Database Integration**: User information and scores are stored in a MySQL database for persistence.
* **Leaderboard**: Displays top scores to encourage competition among users.

# 4. System Design

**4.1. Architecture**

* **Model-View-Controller (MVC)**: The app follows the MVC pattern to separate concerns and organize the codebase.
* **Database Schema**: Includes tables for Users, Quizzes, Questions, and Scores.

**4.2. Class Diagram**

* **User**: Represents a user with attributes like username, password, and email.
* **Question**: Represents a math question with attributes like question text, answer, and difficulty level.
* **Quiz**: Represents a quiz session with attributes like category and score.
* **DatabaseHelper**: A utility class for handling database connections and queries.

# 5. Implementation

**5.1. User Interface**

* **Screen**: Allows users to do various mathematical operations.
* **Main Menu**: Provides options to select quiz categories such as addition Subtraction Multiplication and division to view the result.
* **Quiz Screen**: Displays the questions and records user answers.
* **Result Screen**: Shows the final score and provides an option to retry.

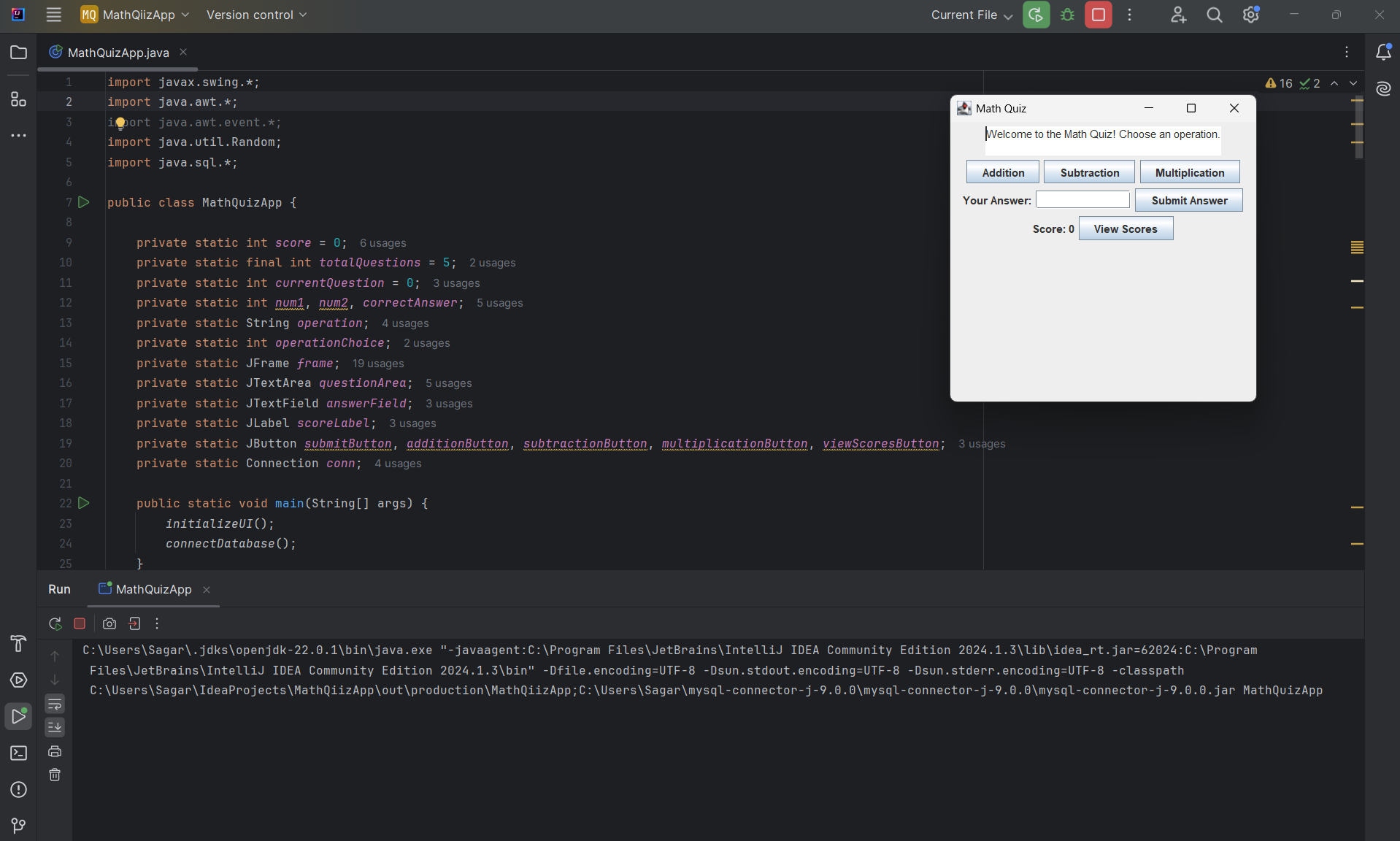
**5.2. Database Integration**

* **Connection**: Uses JDBC (Java Database Connectivity) to connect to the MySQL database.
* **CRUD Operations**: Includes operations to create, read, update, and delete user and score data.

# 6. Code.

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.util.Random;  
import java.sql.\*;  
  
public class MathQuizApp {  
  
 private static int *score* = 0;  
 private static final int *totalQuestions* = 5;  
 private static int *currentQuestion* = 0;  
 private static int *num1*, *num2*, *correctAnswer*;  
 private static String *operation*;  
 private static int *operationChoice*;  
 private static JFrame *frame*;  
 private static JTextArea *questionArea*;  
 private static JTextField *answerField*;  
 private static JLabel *scoreLabel*;  
 private static JButton *submitButton*, *additionButton*, *subtractionButton*, *multiplicationButton*, *viewScoresButton*;  
 private static Connection *conn*;  
  
 public static void main(String[] args) {  
 *initializeUI*();  
 *connectDatabase*();  
 }  
  
 private static void initializeUI() {  
 *frame* = new JFrame("Math Quiz");  
 *frame*.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 *frame*.setLayout(new FlowLayout());  
  
 *questionArea* = new JTextArea(2, 20);  
 *questionArea*.setEditable(false);  
 *questionArea*.setText("Welcome to the Math Quiz! Choose an operation.");  
  
 *scoreLabel* = new JLabel("Score: 0");  
 *answerField* = new JTextField(10);  
 *submitButton* = new JButton("Submit Answer");  
 *additionButton* = new JButton("Addition");  
 *subtractionButton* = new JButton("Subtraction");  
 *multiplicationButton* = new JButton("Multiplication");  
 *viewScoresButton* = new JButton("View Scores");  
  
 *frame*.add(*questionArea*);  
 *frame*.add(*additionButton*);  
 *frame*.add(*subtractionButton*);  
 *frame*.add(*multiplicationButton*);  
 *frame*.add(new JLabel("Your Answer:"));  
 *frame*.add(*answerField*);  
 *frame*.add(*submitButton*);  
 *frame*.add(*scoreLabel*);  
 *frame*.add(*viewScoresButton*);  
  
 *frame*.setSize(350, 350);  
 *frame*.setVisible(true);  
  
 *additionButton*.addActionListener(e -> *startQuiz*(1));  
 *subtractionButton*.addActionListener(e -> *startQuiz*(2));  
 *multiplicationButton*.addActionListener(e -> *startQuiz*(3));  
 *submitButton*.addActionListener(e -> *checkAnswer*());  
 *viewScoresButton*.addActionListener(e -> *displayHistoricalScores*());  
 }  
  
 private static void connectDatabase() {  
 try {  
 // Load MySQL JDBC driver  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
  
 // Establish connection  
 String url = "jdbc:mysql://localhost:3306/quiz"; // Replace with your DB name  
 String user = "root"; // Replace with your username  
 String password = "Sagar@9075"; // Replace with your password  
 *conn* = DriverManager.*getConnection*(url, user, password);  
  
 // Create table if it doesn't exist  
 Statement stmt = *conn*.createStatement();  
 String createTableQuery = "CREATE TABLE IF NOT EXISTS scores (" +  
 "id INT AUTO\_INCREMENT PRIMARY KEY, " +  
 "score INT)";  
 stmt.execute(createTableQuery);  
 } catch (SQLException | ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private static void startQuiz(int operationChoiceSelected) {  
 *operationChoice* = operationChoiceSelected;  
 *currentQuestion* = 0;  
 *score* = 0;  
 *updateScore*();  
 *nextQuestion*();  
 }  
  
 private static void nextQuestion() {  
 Random random = new Random();  
 *num1* = random.nextInt(100) + 1;  
 *num2* = random.nextInt(100) + 1;  
  
 switch (*operationChoice*) {  
 case 1:  
 *correctAnswer* = *num1* + *num2*;  
 *operation* = "+";  
 break;  
 case 2:  
 *correctAnswer* = *num1* - *num2*;  
 *operation* = "-";  
 break;  
 case 3:  
 *correctAnswer* = *num1* \* *num2*;  
 *operation* = "\*";  
 break;  
 }  
  
 *questionArea*.setText("What is " + *num1* + " " + *operation* + " " + *num2* + "?");  
 }  
  
 private static void checkAnswer() {  
 try {  
 int userAnswer = Integer.*parseInt*(*answerField*.getText());  
 if (userAnswer == *correctAnswer*) {  
 *score*++;  
 JOptionPane.*showMessageDialog*(*frame*, " Awesome Correct!", "Answer", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(*frame*, " Opps Incorrect Try Again !" + *correctAnswer*, "Answer", JOptionPane.*ERROR\_MESSAGE*);  
 }  
  
 *currentQuestion*++;  
 *updateScore*();  
  
 if (*currentQuestion* < *totalQuestions*) {  
 *nextQuestion*();  
 } else {  
 *endQuiz*();  
 }  
 } catch (NumberFormatException e) {  
 JOptionPane.*showMessageDialog*(*frame*, "Please enter a valid number.", "Input Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
  
 private static void updateScore() {  
 *scoreLabel*.setText("Score: " + *score*);  
 }  
  
 private static void endQuiz() {  
 JOptionPane.*showMessageDialog*(*frame*, "Quiz Finished! Final Score: " + *score* + "/" + *totalQuestions*, "Quiz Over", JOptionPane.*INFORMATION\_MESSAGE*);  
 *saveScore*();  
 *score* = 0;  
 *updateScore*();  
 }  
  
 private static void saveScore() {  
 try {  
 String query = "INSERT INTO scores (score) VALUES (?)";  
 PreparedStatement pstmt = *conn*.prepareStatement(query);  
 pstmt.setInt(1, *score*);  
 pstmt.executeUpdate();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static void displayHistoricalScores() {  
 try {  
 Statement stmt = *conn*.createStatement();  
 ResultSet rs = stmt.executeQuery("SELECT \* FROM scores");  
 StringBuilder sb = new StringBuilder("Historical Scores:\n");  
 while (rs.next()) {  
 sb.append("Score: ").append(rs.getInt("score")).append("\n");  
 }  
 JOptionPane.*showMessageDialog*(*frame*, sb.toString(), "Past Scores", JOptionPane.*INFORMATION\_MESSAGE*);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
}

# 7.Result.



# **7. Testing**

**Unit Testing**: Conducted for individual components like QuestionGenerator and DatabaseHelper.

**Integration Testing**: Ensured smooth interaction between the app and the database.

**User Testing**: Feedback from users helped refine the UI and improve the user experience.

# **8. Conclusion**

The Math Quiz App provides an interactive platform for users to enhance their math skills. With a robust backend and an intuitive frontend, the app offers a seamless experience for users of all ages.