# QUIZ APPLICATION

## A MINI PROJECT REPORT

***Submitted by***

## JAY AGRAVAT

### 92100103260

## RUSHI LUKKA

## 92100103262

## HITESH RABADIA

## 92100103284

### BACHELOR OF ENGINEERING

***in***

### Computer Engineering

****

## Marwadi University, Rajkot

**December 2022**

CERTIFICATE

# 

This is to certify that the project report submitted along with the project entitled **QUIZ APPLICATION** has been carried out by **JAY AGRAVAT-92100103260,**

**RUSHI LUKKA-92100103262, HITESH RABADIA-92100103284** under my guidance in partial fulfillment for the degree of Bachelor of Technology in Computer Engineering, 3rd Semester of Marwadi University, Rajkot during the academic year 2022-23.

RAVIKUMAR NATARAJAN HARDIK DOSHI

Internal Guide Head of the Department

### 

## Marwadi University

**Rajkot**

# DECLARATION

We hereby declare that the Mini Project-I report submitted along with the Project entitled

**QUIZ APPLICATION** submitted in partial fulfilment for the degree of Bachelor of Technology in COMPUTER ENGINEERING to Marwadi University, Rajkot, is a bonafide record of original project work carried out by me / us at Marwadi University under the supervision of **RAVIKUMAR NATARAJAN** and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

Name of the Student Sign of Student

1. JAY AGRAVAT
2. RUSHI LUKKA
3. HITESH RABADIA

### List of Figures

### 

Fig 1.1 Use case / Procedure Diagram ........................................................................ 2

Fig 1.2 Activity / Process Diagram ............................................................................. 5

Fig 2.1 Use case / Procedure Diagram ...................................................................... 12

Fig 2.2 Activity / Process Diagram ........................................................................... 15

Fig 2.3 Use case / Procedure Diagram ...................................................................... 22

Fig 2.4 Activity / Process Diagram ........................................................................... 25

Fig 3.1 Use case / Procedure Diagram ...................................................................... 32

Fig 4.1 Activity / Process Diagram ........................................................................... 35

Fig 4.2 Use case / Procedure Diagram ...................................................................... 42

Fig 4.3 Activity / Process Diagram ........................................................................... 43

### List of Tables

Table 1.1 Popular Methods / Techniques ................................................................... 2

Table 1.2 User / Reading / Observation Table ............................................................ 5

Table 2.1 Popular Methods / Techniques .................................................................. 12

Table 2.2 User / Reading / Observation Table .......................................................... 15

Table 2.3 Popular Methods / Techniques .................................................................. 22

Table 2.4 User / Reading / Observation Table .......................................................... 25

Table 3.1 Popular Methods / Techniques .................................................................. 32

Table 4.1 User / Reading / Observation Table .......................................................... 35

Table 4.2 Popular Methods / Techniques .................................................................. 42

Table 4.3 User / Reading / Observation Table .......................................................... 43

|  |  |
| --- | --- |
|  | **Abbreviations** |
| ALU | Arithmetical & Logical Unit |
| SDLC | Software Development Life Cycle |
|  |  |
|  |  |
|  |  |

### Table of Contents

Acknowledgement................................................................................................................... i

Abstract .................................................................................................................................. ii

List of Figures ...................................................................................................................... iii

List of Tables ........................................................................................................................ iv

List of Abbreviations ............................................................................................................. v Table of Contents .................................................................................................................. vi

Chapter 1 ………………....................................................................................................... 1

1.1 Introduction to Java ......................................................................................................... 2

1.1.1 Benefits of Java.................................................................................................... 3

1.2 Add required topic............................................................................................................ 4

Chapter 2 ………………........................................................................................................ 8

2.1 Introduction to Project Topic............................................................................................ 9

2.1.1 How to do ......................................................................................................... 10

2.2 Drawbacks in Existing System ..................................................................................... 11 2.3 Advantages of Proposed System ................................................................................... 12

2.4 Functional Requirements…............................................................................................ 13

2.4.1 Tools .................................................................................................................. 14

2.4.2 Front End and Back End ................................................................................... 14

Chapter 3 ………………..................................................................................................... 15

3.1 Source code ………....................................................................................................... 14

Chapter 4 ………………..................................................................................................... 15

4.1 Screenshots ………........................................................................................................ 17

Chapter 5 ………………..................................................................................................... 15

4.1 Conclusion ………........................................................................................................ 17

4.2 Future Enhancement...................................................................................................... 17

References............................................................................................................................ 40

**CHAPTER 1**

**OVERVIEW OF JAVA**

**1.1 Introduction of Java**

**Java** is a general-purpose, concurrent, class-based, object-oriented computer programming

language that is specifically designed to have as few implementation dependencies as possible. It

is intended to let application developers "write once, run anywhere" **(WORA)**, meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.

**[1][2].** Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystem’ Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Java [3] can be used to write applications and applets. A Java application is similar to any other high-level language program: It can only be compiled and then run on the same machine. An applet is compiled on one machine, stored on a server in binary, and can be sent to another machine over the Internet to be interpreted by a Java-aware browser. Java comes with a large library of ready-made classes and objects. The key difference between Java 1.0 and 1.1 was in this library. Similarly, Java 2.0 has a very much larger library for handling user interfaces (Swing by name) but only small changes to the core of the language.

**1.2** Object-oriented Programming

Java supports object-oriented programming techniques that are based on a hierarchy of classes

and well-defined and cooperating objects .

**Classes:** A class is a structure that defines the data and the methods to work on that data. When you write programs in Java, all program data is wrapped in a class, whether it is a class you write or a class you use from the Java API libraries. Classes in the Java API libraries define a set of

objects that share a common structure and behavior.

**Objects:** An instance is a synonym for object. A newly created instance has data members and methods as defined by the class for that instance.

Well-Defined Boundaries and Cooperation: Class definitions must allow objects to cooperate

during execution.

**CHAPTER 2**

**OVERVIEW OF PROJECT**

**2.1 Introduction of Quiz Generator**

[2.1.1]We are implementing basic concepts of java and data file handling.

[2.1.2]There are two levels of Quiz.

[2.1.3]Records can be saved with respect to students GR.No. and can be search afterwards.

[2.1.4]Records are conserved in text files.

[2.1.5]Records can be retrieved from files.

**2.2 Drawbacks of Existing System**

[2.2.1]No proper storing of data

[2.2.2]No proper method to see records

**2.3 Advantages of Proposed System**

[2.3.1]Data is stored in a file.

[2.3.2]User can search their marks using Enrollment Number

**2.4 Functional Requirements**

**Conduct Quiz:**

User need to enter G.R. number.After that user will directed to quiz program.

If user passes level 1 he will be given option to attempt level 2 or return to home.

**Search Record:**

User can search the marks by entering G.R. number.

**Fetch all:**

User can see all the records.

**2.4.1 Tools**

[2.4.1.1] Netbeans IDE

[2.4.1.2] Text file.

**2.4.2 Front End and Back End**

[2.4.2.1] File Handling

**CHAPTER 3**

**PROJECT SOURCE CODE**

**3.1 Source Code**

package oopproject;

import java.io.\*;

import java.util.\*;

public class Oopproject{

static int correct1;

static int correct2;

static String CreateLine(int gno,int m){

return gno+" "+m+"\n";

}

static void inputData(String s1){

try{

File file = new File("main.txt");

FileWriter fr = new FileWriter(file, true);

fr.write(s1);

fr.close();

}

catch(Exception e){

System.out.println(e);}

}

static void quiz(){

Scanner s = new Scanner(System.in);

int a;

System.out.print("Enter GR no.(3 digit) : ");

a=s.nextInt();

int y = quiz\_1();

String s1 = CreateLine(a,y);//y -> marks

inputData(s1);//to file

}

static int quiz\_1(){

Scanner s = new Scanner(System.in);

correct1 = 0;

int questions = 10;

System.out.println ("Welcome to the Multiple Choice Quiz! \n");

String[][]Ques\_Ans1 ={//i=no.of question j=0,question j=1,options,j=2,coreect answer

{"Which method can be used to find the length of a string in JAVA?",

"\n A. length() \n B. getlength() \n C. len \n D. getsize()\n","A"},

{"Which class is available to all the class automatically?",

"\n A. Swing \n B. Applet \n C. Object \n D. ActionEvent\n","C"},

{"Which package is directly available to our class without importing it?",

"\n A. Swing \n B. Applet \n C. net \n D. Lang\n","D"},

{"String class is defined in which package?",

"\n A. Lang \n B. Swing \n C. Applet \n D. awt\n","A"},

{"Which institute is best for java coaching?",

"\n A. Utek \n B. Aptech \n C. SSS IT \n D. jtek\n","C"},

{" Which one among these is not a keyword?",

"\n A. Class \n B. int \n C. get \n D. if\n","C"},

{"Which one among these is not a class?",

"\n A. Swing \n B. ActionPerformed \n C. ActionEvent\n D. Button\n","B"},

{"which one among these is not a function of Object class?",

"\n A. toString \n B. finalize \n C. equals \n D. getDocumentBase\n","D"},

{"which function is not present in Applet class?",

"\n A. init \n B. main \n C. start \n D. destroy\n","B"},

{"Which one among these is not a valid component?",

"\n A. JButton \n B. JList \n C. JBUttonGroup \n D. JTextArea\n","C"}

};

String[]user\_ans1 = new String[(int) questions];

int i = 0;

do{

System.out.print((i + 1) + ". " + Ques\_Ans1[i][0] + " " +Ques\_Ans1[i][1]);

System.out.print("Enter your Answer:");

user\_ans1[i] = String.valueOf (s.next().charAt (0));

user\_ans1[i]=user\_ans1[i].toUpperCase();

if (Ques\_Ans1[i][2].equals (user\_ans1[i])){

System.out.println ("\nCorrect!");

correct1++;

}

else{

System.out.println ("\n Incorrect. The correct answer is " +Ques\_Ans1[i][2]);

}

System.out.print ("\n");

i++;

}

while (i < questions);

System.out.println ("\nYour Score in level 1 is: " + correct1);

if(correct1 > 4){

System.out.println("\nCongratulations!!!!\nYou are eligible for level 2 quiz");

System.out.println("Do you want to Continue??\n1.Press 1 for continue\n2.Any other key for exit");

System.out.print("Enter your Choice: ");

int ch=s.nextInt();

switch(ch){

case 1:

correct2 = quiz\_2();

int e = correct1 + correct2;

System.out.println("Total score : "+e);

return e;

default:

return correct1;

}

}

return correct1;

}

static int quiz\_2(){

Scanner s = new Scanner(System.in);

int u = 0;

int questions = 5;

System.out.println ("Welcome to the Quiz-2 \n");

String[][]Ques\_Ans2 ={//i=no.of question j=0,question j=1,options,j=2,coreect answer

{"What is the size of float variable? ",

"\n A. 8 bit \n B. 16 bit \n C. 32 bit \n D. 64 bit\n", "C"},

{"Which one of these is not a primitive data type? ",

"\n A. integer \n B. char \n C. float \n D. double\n", "B"},

{"Which data type is used to create a variable that should store text? ",

"\n A. myString \n B. string \n C. character \n D. String64\n","D"},

{"How do you create a variable with the numeric value 5? ",

"\n A. num x = 5; \n B. float x = 5; \n C. x = 5; \n D. int x = 5;\n","A"},

{"Which method can be used to find the length of a string in JAVA?",

"\n A. length() \n B. getlength() \n C. len \n D. getsize()\n","A"}};

String[]user\_ans2 = new String[(int) questions];

int i = 0;

do{

System.out.print ("" + (i + 1) + ". " + Ques\_Ans2[i][0] + " " +Ques\_Ans2[i][1]);

System.out.print("Enter your Answer:");

user\_ans2[i] = String.valueOf (s.next().charAt (0));

user\_ans2[i]=user\_ans2[i].toUpperCase();

if (Ques\_Ans2[i][2].equals (user\_ans2[i])){

System.out.println ("\n Correct!");

u++;

}

else{

System.out.println ("\n Incorrect. The correct answer is " +Ques\_Ans2[i][2]);

}

System.out.print ("\n");

i++;

}

while (i < questions);

return u;

}

static void search(){

//get roll no getByRollNo

int p =0;

Scanner s = new Scanner(System.in);

int a;

System.out.print("Enter GR no.(3 digit) : ");

a=s.nextInt();

String x = Integer.toString(a);

try{

File f1= new File("Main.txt");

FileReader fr = new FileReader(f1);

BufferedReader br=new BufferedReader(fr);

String l=null;

System.out.println("Previous Record/s : ");

System.out.println("GR no. | marks");

while((l=br.readLine()) !=null)

{ char r[] = new char[3];

r[0] =l.charAt(0); r[1] =l.charAt(1); r[2] =l.charAt(2);

char c[] = new char[3];

c[0] =x.charAt(0); c[1] =x.charAt(1); c[2] =x.charAt(2);

if(r[0] == c[0] && r[1] == c[1] && r[2] == c[2]){

System.out.println(l);

p=1;

}

}

if(p == 0){System.out.println("No record found ");}

br.close();

}

catch(Exception e){

System.out.println(e);;}

}

static void fetchAll(){

try{

File f1= new File("Main.txt");

FileReader fr = new FileReader(f1);

BufferedReader br=new BufferedReader(fr);

String l=null;

System.out.println("GR no. | marks");

while((l=br.readLine()) !=null)

{ System.out.println(l);

}

br.close();

}

catch(Exception e){

System.out.println(e);;}

}

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int ch;

System.out.println("-\_-\_-\_-\_-\_ONLINE QUIZ GENERATOR\_-\_-\_-\_-\_-\n");

while(true){

System.out.println("--------------------------------");

System.out.println("1 to Conduct Quiz\n2 to search record");

System.out.println("3 to fetch all records\n4 to exit\n");

System.out.print("Enter choice :");

ch=s.nextInt();

switch(ch){

case 1: quiz();break;

case 2: search();break;

case 3: fetchAll();break;

case 4: System.exit(0);

default:System.out.println("Enter valid choice!!!\n");

}

}

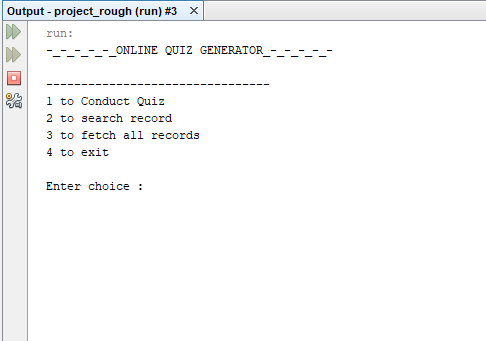
}

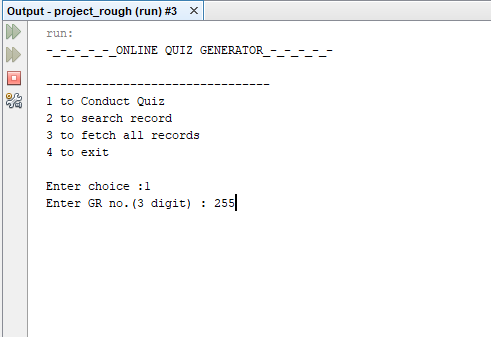
}

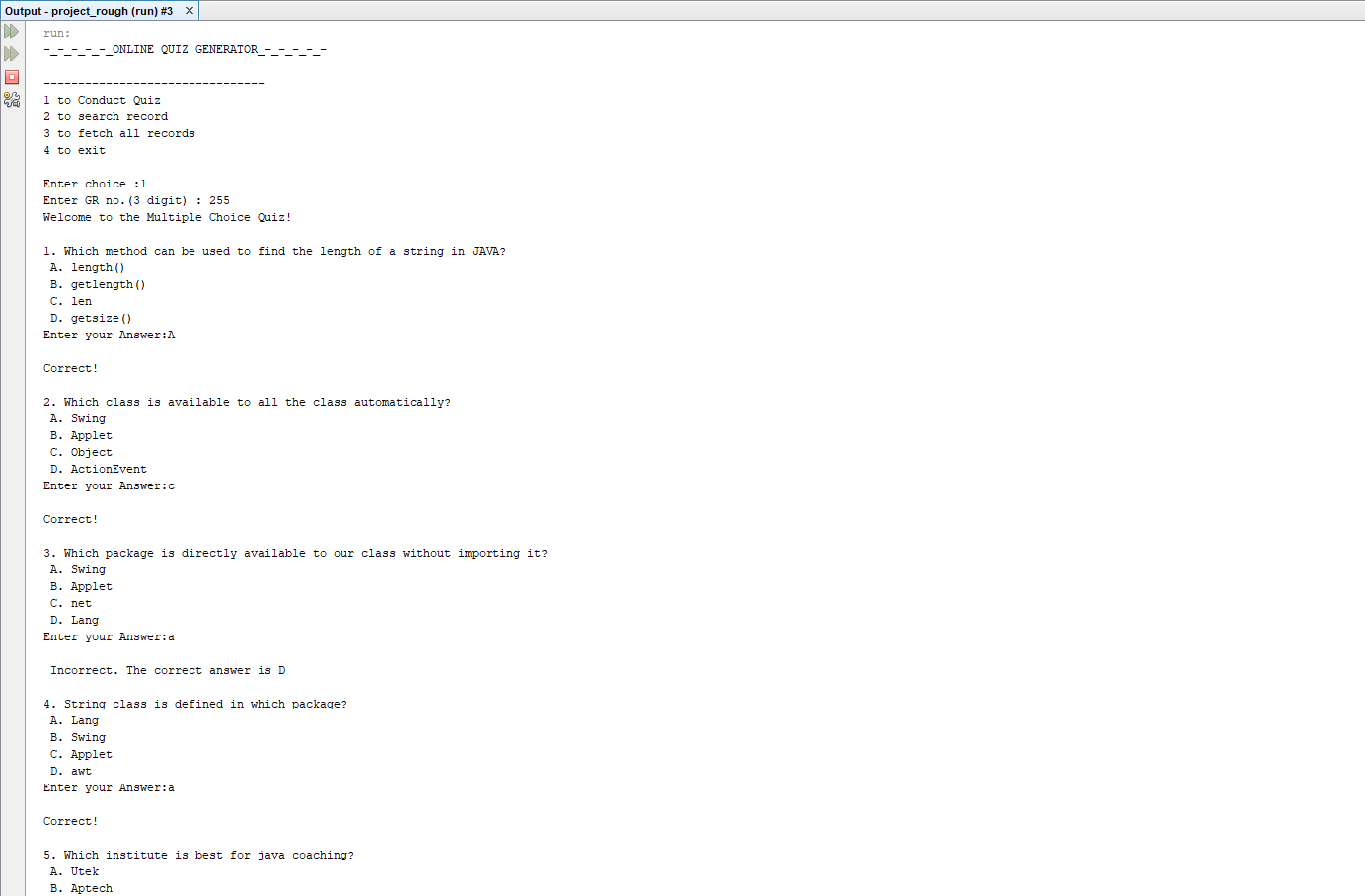
**CHAPTER 4**

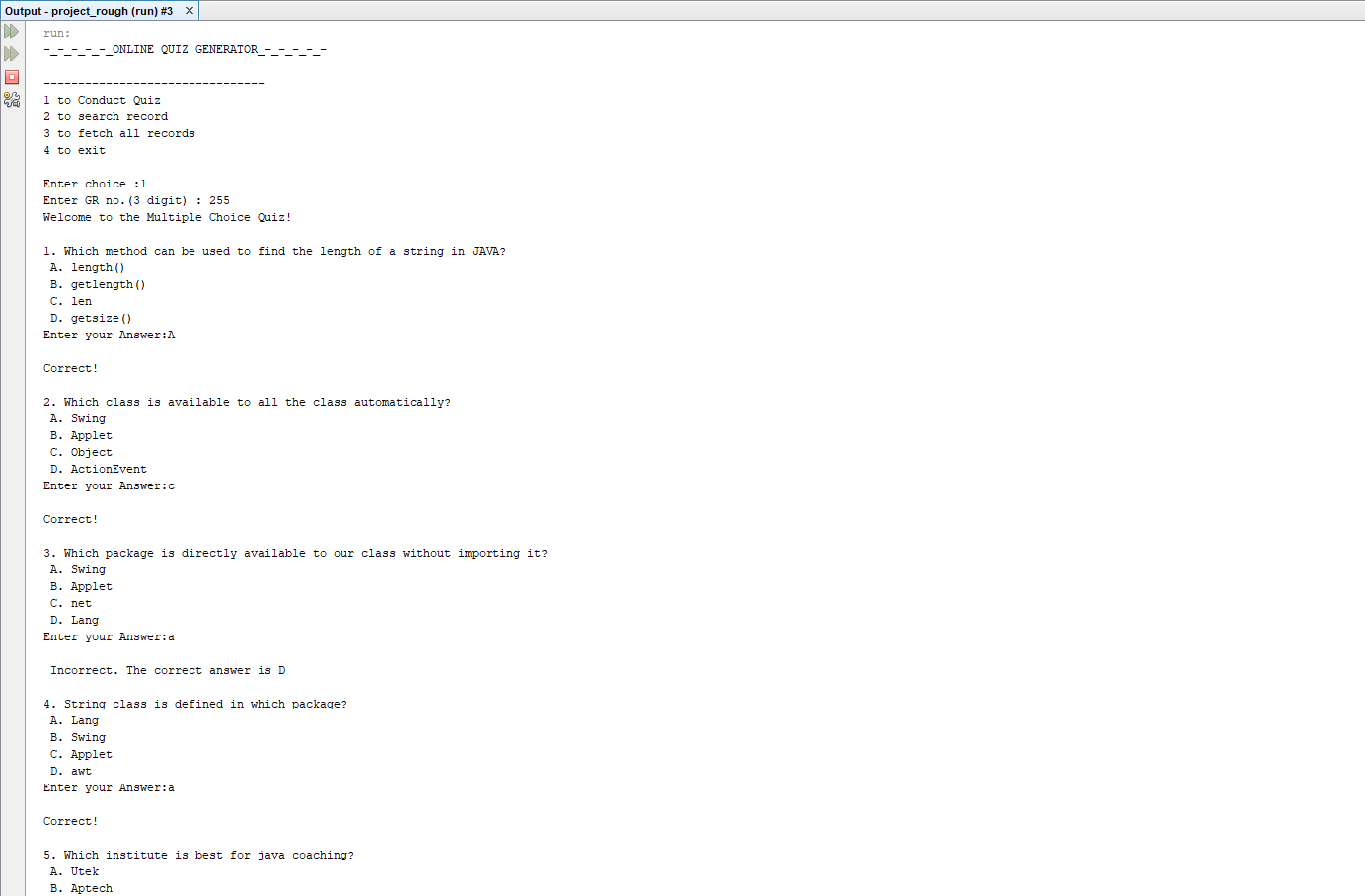
**SCREENSHOTS**

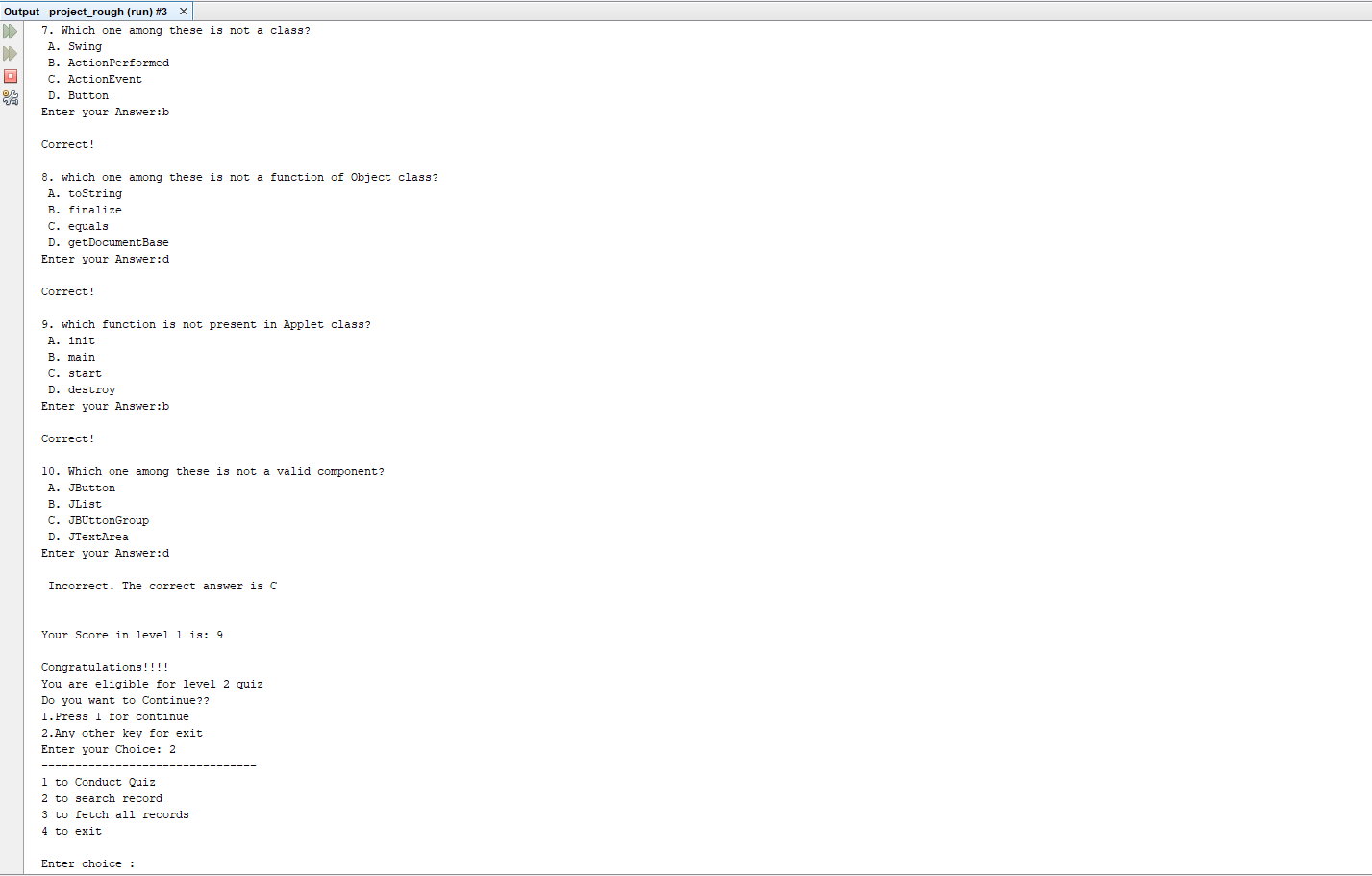
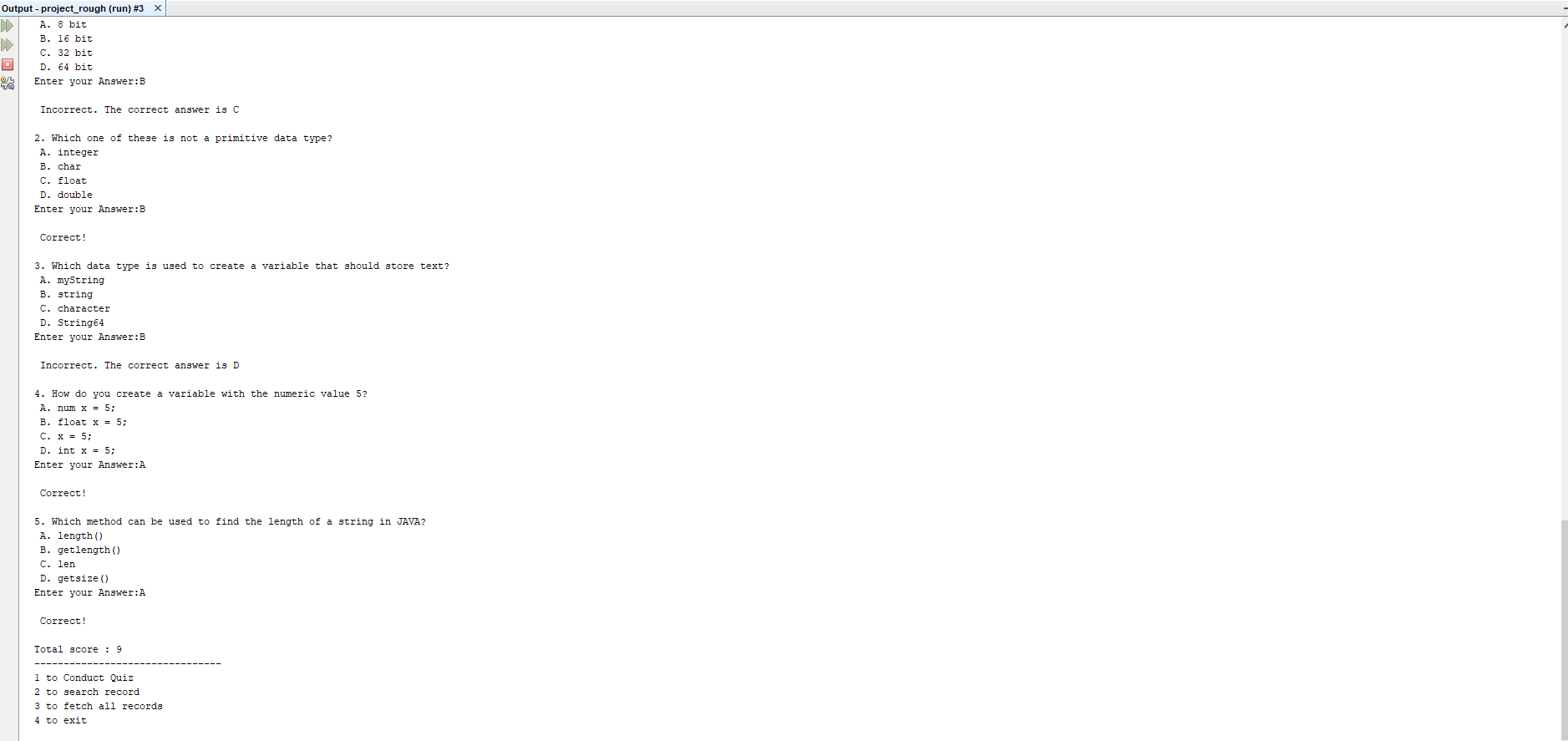
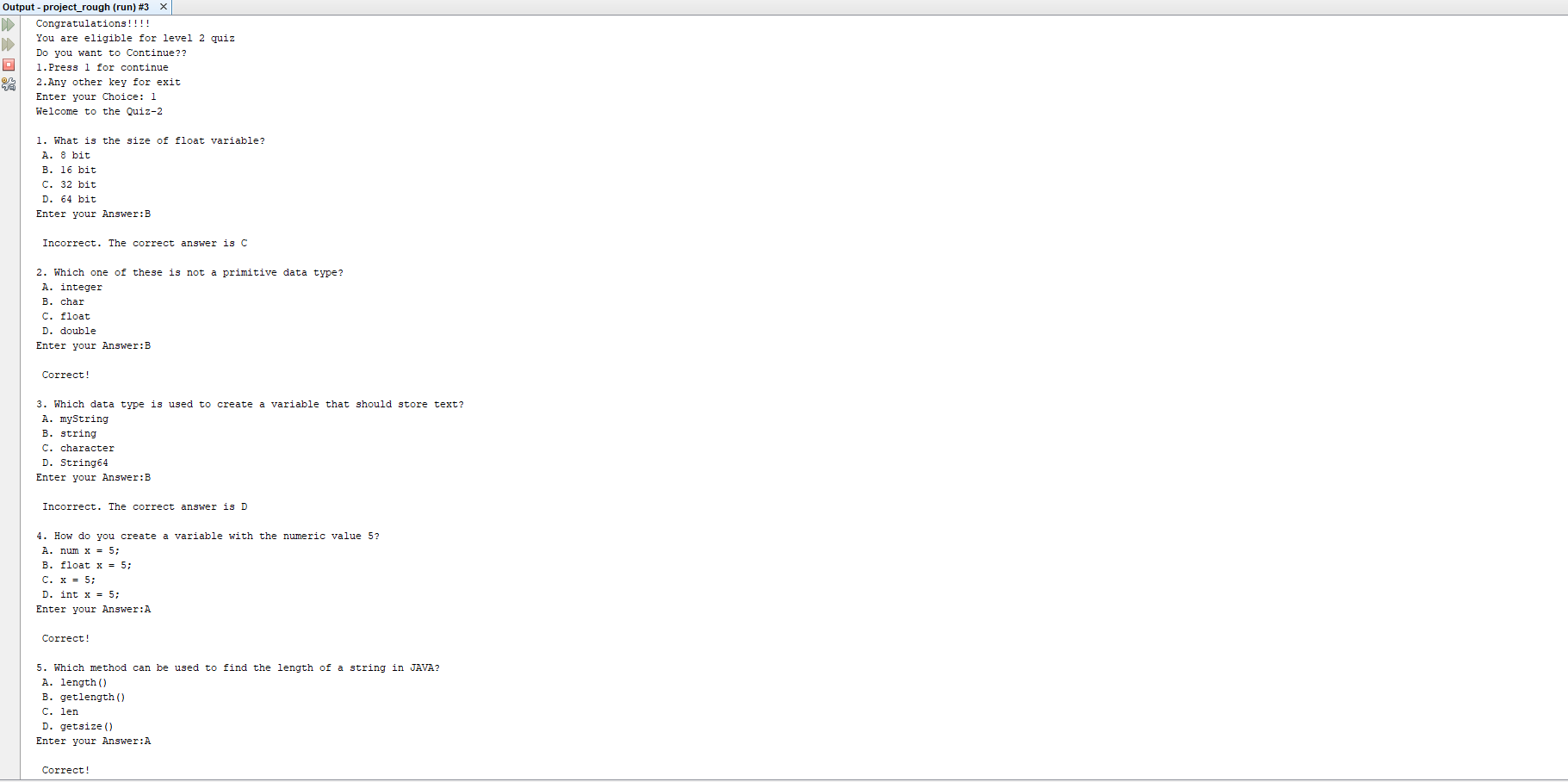
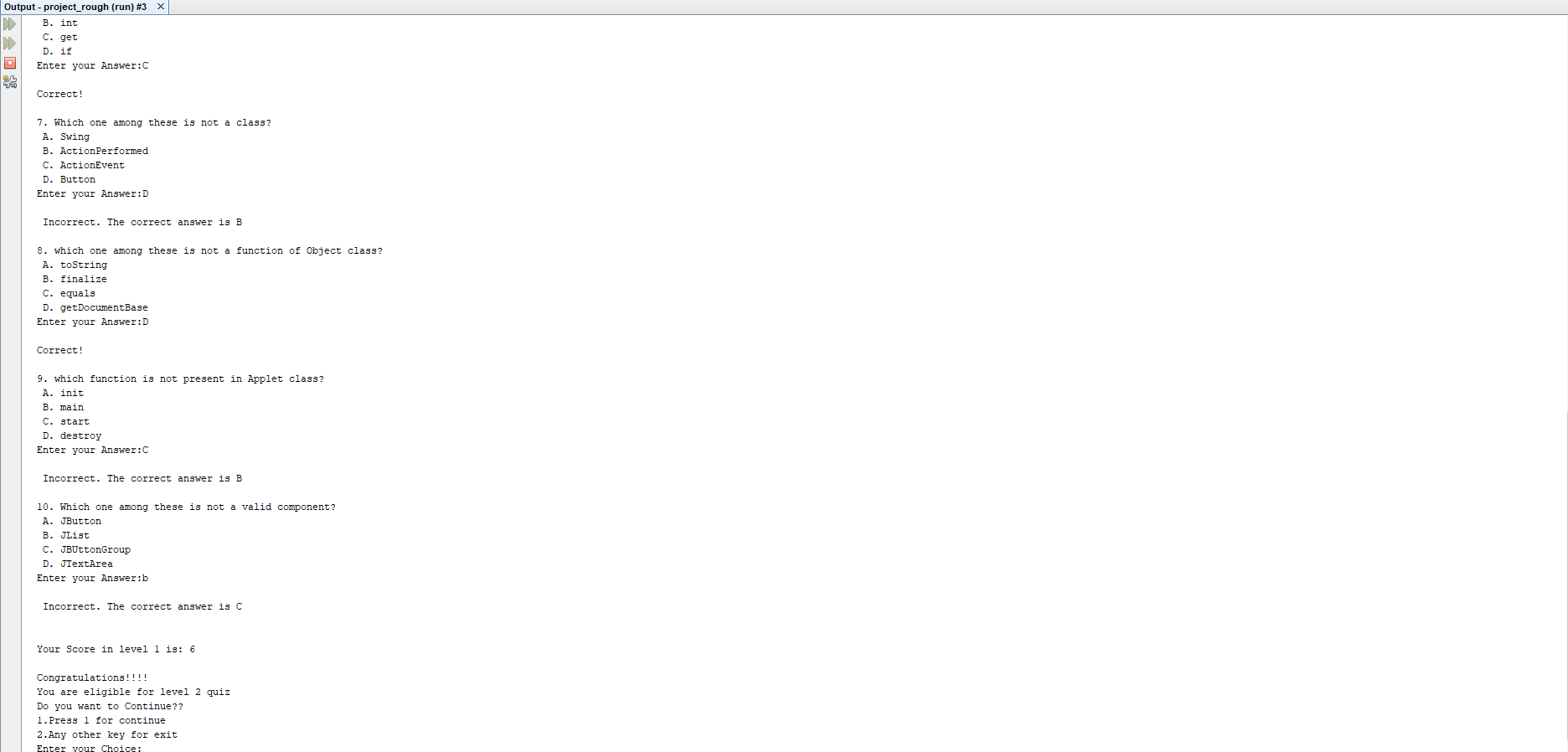
**4.1 Admin Screenshot**

****

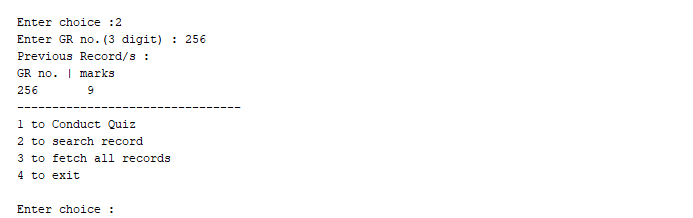
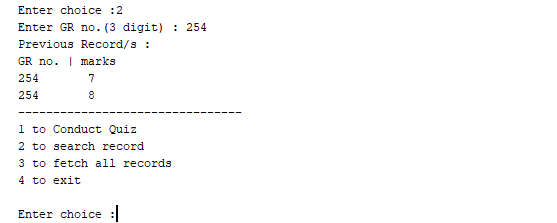
****

****

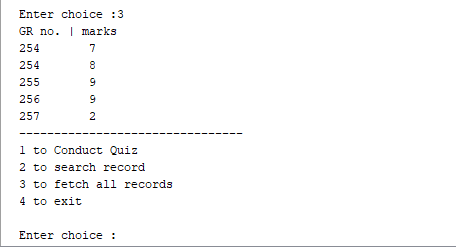
****

****

**Search:**

****

**Display all:**

****

**CHAPTER 5**

**CONCLUSION AND FUTURE ENHANCEMENTS**

**5.1 Conclusion**

This report presents an introduction to Java and how basic concepts of java and data file handling used to build Quiz generator.This was an introduction to the main goal of our report that presented that Quiz generator project used to take quiz and save it in text files and can be used to display marks to user.

**5.2 Future Enhancements**

[5.2.1] Provide option to delete records.

[5.2.2]Add option to add more MCQ by user.

**REFERENCES**

[1] Axelrod, CW (2013). Managing the Risks of Cyber-Physical Systems. In 2013 Systems,

Applications and Technology Conf. (LISAT), pp. 1–6. IEEE: Long Island.

[2] www.researchgate.com/doc/library\_management