

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



## An Internship Project Report on

## *Quiz Application*

Submitted in partial fulfillment of the requirements for the VIII Semester of  
degree of **Bachelor of Engineering in Information Science and Engineering** of  
Visvesvaraya Technological University, Belagavi

by

**Amritha Kandiyl**

**1RN18IS015**

Under the Guidance of

**Dr. Rajkumar R**

Associate Professor

Department of ISE



ESTD: 2001

*An Institute with a Difference*

**Department of Information Science and Engineering**

**RNS Institute of Technology**

**Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,  
Channasandra, Bengaluru-560098**

**2021-2022**

# RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,  
Channasandra, Bengaluru - 560098

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



### CERTIFICATE

Certified that the Internship work entitled **Quiz Application** has been successfully completed by **Amritha Kandiyl (1RN18IS015)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8<sup>th</sup> semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

---

**Dr. Rajkumar R**  
Internship Guide  
Associate Professor  
Department of ISE

---

**Dr. Suresh L**  
Professor and HoD  
Department of ISE  
RNSIT

---

**Dr. M K Venkatesha**  
Principal  
RNSIT

**Name of the Examiners**

**External Viva**

**Signature with Date**

1. \_\_\_\_\_

1. \_\_\_\_\_

2. \_\_\_\_\_

2. \_\_\_\_\_

# DECLARATION

I, **AMRITHA KANDIYIL** [USN: **1RN18IS015**], student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Quiz Application*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

**AMRITHA KANDIYIL**

**(1RN18IS015)**

# **Abstract**

Manually conducting exams/tests and checking students' scores is one of the key areas where a lot of time is wasted. This time could have otherwise been used effectively for a productive study time.

Replacing these manual tasks of conducting exams, correcting the answer scripts and giving scores, with some comprehensive apps can save a lot of precious teaching time. Also, since all these tasks are being performed by humans, the chances of errors occurring are more.

Quiz application is developed so that students can prepare for an examination or can have a quick revision of whatever they have learnt. As soon as an answer is selected, the computer displays if it is true or false and scores will be counted automatically. An option will be given to reset the test if required.

# ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha**, for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mrs. Vinutha G K**, Assistant Professor , Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R, Co-Founder & CEO, Enmaz Engineering Services Pvt.Ltd**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. Rajkumar R**, Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

**Amritha  
Kandiyil**

# TABLE OF CONTENTS

<b>Abstract</b>	<b>i</b>
<b>Acknowledgment</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of figures</b>	<b>v</b>
<b>List of Abbreviations</b>	<b>vi</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Background	1
1.2 Existing System	1
1.3 Proposed System	1
<b>2. Literature Review</b>	<b>2</b>
<b>3. System Design</b>	<b>4</b>
3.1 Widget Tree	4
3.1.1 Questions Screen	4
<b>4. Implementation</b>	<b>5</b>
4.1 Requirement specification	5
4.1.1 Hardware Requirements	5
4.1.2 Software Requirements	5
4.1.3 Flutter	5
4.1.3.1 Dart Platform	5
4.1.3.2 Flutter Engine	6
4.1.3.3 Foundation Library	6
4.1.3.4 Design Specific Widgets	6
4.2 Discussion of Code Segments	6
4.2.1 main.dart	6
4.2.2 home.dart	7
4.2.3 answer.dart	13

<b>5. Testing</b>	16
5.1 Introduction	16
5.2 Levels of Testing	16
5.2.1 Unit testing	16
5.2.2 Integration Testing	17
5.2.3 System Testing	17
5.2.4 Validation Testing	17
5.2.5 Output Testing	17
5.2.6 User Acceptance Testing	17
<b>6. Results</b>	18
6.1 Questions Screen	18
6.2 Evaluation of answers – true and false	19
6.3 Displaying the final score	21
<b>7. Conclusion and future work</b>	23
7.1 Conclusion	23
7.2 Future work	23
<b>References</b>	24

## List of Figures

<b>Fig. no.</b>	<b>Fig. description</b>	<b>Page number</b>
3.1.1	Widget tree for questions screen	4
6.1	Questions screen	18
6.2.1	Correct answer	19
6.2.2	Wrong answer	19
6.3	Displaying the final score	21
6.4	Restart quiz	22



## **List of Abbreviations**

DSDEPHR	Distributed Storage design for Encrypted Personal Health Record
HD	Hadoop Database
SRS	Software Requirement Specification

# **1. Introduction**

## **1.1 Background**

Quiz application is an application developed for learning and improving knowledge in the educational area. This application can be used in schools, colleges and also by students themselves before a test or examination to evaluate themselves or even for a quick revision.

## **1.2 Existing System**

In the existing system, the students have to manually contact the teacher to conduct examinations and also, a single examination for a single student cannot be done. It will be tedious for the teacher when a large number of students want to check their scores. The human effort is more here.

## **1.3 Proposed System**

To overcome the drawbacks mentioned above, the proposed system has been developed. This project has been developed so that a student can easily test themselves and have a quick revision. The interface is easy to understand and is extremely user friendly.

## **2. Literature Review**

### **2.1 Quiz app**

Attending exams and monitoring the same are important in order to track students' performance in class. Quizzes can be used for a quick revision before a school/college exam, competitive exam or just for general knowledge sake. Students can use this to see where they stand compared to their peers and improve their scores thereof. Therefore, any problems related to students' academics should be identified as early as possible so that appropriate measures can be taken to address them. However, conducting exams, if done manually, can be tedious and time consuming, especially for classes with large numbers of students.

The system automates most of the steps involved in the quiz app. Students can answer each question and if it is correct, it will be shown in green and if it is wrong, it will be shown in red. Scores are counted automatically. After answering each question, a symbol for the correct or wrong question will be displayed above. At the end of the quiz, a "Restart Quiz" option will be shown. If it is clicked, the score goes back to zero and the questions will be displayed once again.

#### **2.1.1 Introduction**

Manually conducting exams/tests and checking students' scores is one of the key areas where a lot of time is wasted. This time could have otherwise been used effectively for a productive study time.

Replacing these manual tasks of conducting exams, correcting the answer scripts and giving scores, with some comprehensive apps can save a lot of precious teaching time. Also, since all these tasks are being performed by humans, the chances of errors occurring are more.

Quiz app is developed so that students can prepare for an examination or can have a quick revision of whatever they have learnt. As soon as an answer is selected, the computer displays if it is true or false and scores will be counted automatically. An option will be given to reset the test if required.

### **2.1.2 Proposed System**

Quiz application is an application developed for learning and improving knowledge in the educational area. This application can be used in schools, colleges and also by students themselves before a test or examination to evaluate themselves or even for a quick revision.

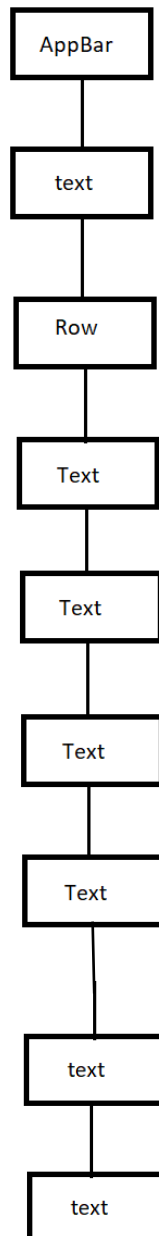
This application is:

- User friendly - the graphical user interface provided in the proposed system, makes the system very user friendly and easy to operate.
- Very less paper work: The proposed system requires very less paper work. All the data is fed into the computer immediately and scores are generated through computers. Moreover, work becomes very easy because there is no need to keep data on papers.
- Computer operator control: Computer operator control will be there so no chance of errors. Also, work can be done speedily and in time.

## 3. System Design

### 3.1 Widget Tree

#### 3.1.1 Questions Screen



## 4. Implementation

### 4.1 Requirement Specifications

#### 4.1.1 Hardware Requirements

- CPU: Pentium processor and above
- RAM: 4 GB
- HDD: 40 GB

#### 4.1.2 Software Requirements

- **Operating System:** Windows 8 and above
- **Front-end Design:** Visual Studio Code
- **Front-end Language:** Dart

#### 4.1.3 Flutter

**Flutter** is an open-source UI, software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, Web Platform and the web from a single codebase.

The major components of Flutter include:

- Dart Platform
- Flutter Engine
- Foundation Library
- Design-specific widgets
- Flutter Development Tools (DevTools)

##### 4.1.3.1 Dart Platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses

Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful, hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

#### 4.1.3.2 Flutter Engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform Specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plug-in architecture, and a Dart runtime and compile toolchain.

#### 4.1.3.3 Foundation Library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

#### 4.1.3.4 Design Specific Widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and *Cupertino* widgets implement Apple's iOS Human Interface Guidelines.

## 4.2 Discussion of Code Segments

### 4.2.1 main.dart

```
import 'package:flutter/material.dart';
import 'package:quiz_app/home.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {

  @override
```

```

Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Quiz app',
    theme: ThemeData(
      primaryColor: Color(0xFF55C1EF),
    ),
    home: Home(),
  );
}

```

## 4.2.2 home.dart

```

import 'package:flutter/material.dart';
import 'package:quiz_app/answer.dart';

class Home extends StatefulWidget {

  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {

  List<Icon> _scoreTracker = [];
  int _questionIndex = 0;
  int _totalScore = 0;
  bool answerWasSelected = false;
  bool endOfQuiz = false;
  bool correctAnswerSelected = false;

  void _questionAnswered(bool answerScore) {
    setState(() {
      // answer was selected
      answerWasSelected = true;

      // check if answer was correct
      if(answerScore) {

```



```

        _totalScore++;
        correctAnswerSelected = true;
    }
    // adding score tracker on top
    _scoreTracker.add(
        answerScore ? Icon(
            Icons.check_circle,
            color: Colors.green,
        ):Icon(
            Icons.clear,
            color: Colors.red,
        ),
    );
    //when the quiz ends
    if(_questionIndex + 1 == _questions.length){
        endOfQuiz = true;
    }

});
}

void _nextQuestion() {
    setState(() {
        _questionIndex++;
        answerWasSelected = false;
        correctAnswerSelected = false;
    });
    // what happens at the end of the quiz
    if(_questionIndex >= _questions.length) {
        _resetQuiz();
    }
}

void _resetQuiz() {
    setState(() {
        _questionIndex = 0;
        _totalScore = 0;
        _scoreTracker = [];
        endOfQuiz = false;
    });
}

```

```

    });
}

@override
Widget build(BuildContext context) {
  return Scaffold(

    appBar: AppBar(
      title: Text(
        'Quiz App',
        style: TextStyle(color: Colors.white),
      ),
      centerTitle: true,
    ),

    body: Center(
      child:
        Column(
          children: [
            Row(
              children: [
                if(_scoreTracker.length == 0)
                  SizedBox(
                    height: 25.0,
                  ), //SizedBox
                if(_scoreTracker.length > 0) ..._scoreTracker
              ], //children
            ), //row
            Container(
              width: double.infinity,
              height: 130.0,
              margin: EdgeInsets.only(bottom: 10.0, left: 30.0, right: 30.0),
              padding: EdgeInsets.symmetric(horizontal: 50.0, vertical: 20.0),
              decoration: BoxDecoration(
                color: Colors.deepOrange,
                borderRadius: BorderRadius.circular(10.0),
              ), //Container
              child: Center(
                child: Text(
                  _questions[_questionIndex]['question'],

```

```

        textAlign: TextAlign.center,
        style: TextStyle(
            fontSize: 20.0,
            color: Colors.white,
            fontWeight: FontWeight.bold,
        ),
    ),
),
),

...(_questions[_questionIndex]['answers']
as List<Map<String, Object>>)
.map(
    (answer) => Answer(
        answerText: answer['answerText'],
        answerColor: answerWasSelected
        ? answer['score']
            ? Colors.green
            : Colors.red
        : null,
        answerTap: () {
            // if answer was already selected, then nothing happens on tap
            if(answerWasSelected) {
                return;
            }
            // answer is being selected
            _questionAnswered(answer['score']);
        },
    ),
),

SizedBox(height: 20.0),
ElevatedButton(
    style: ElevatedButton.styleFrom(
        minimumSize: Size(double.infinity, 40.0),
    ),

    onPressed: () {
        if(!answerWasSelected) {

```

```

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text(
            'Please select an answer before proceeding to the next
question!')),
          ));
        return;
      }
      _nextQuestion();
    },
    child: Text(endOfQuiz ? 'Restart quiz' : 'Next question'),
  ), // ElevatedButton
  Container(
    padding: EdgeInsets.all(20.0),
    child: Text(
      '${_totalScore.toString()}/${_questions.length}',
      style: TextStyle(fontSize: 40.0, fontWeight: FontWeight.bold),
    ),
  ),

],

),
),

);
}
}

```

```

final _questions = const [
  {
    'question': 'How long is New Zealand's Ninety Mile Beach?',
    'answers': [
      {'answerText': '88km, so 55 miles long.', 'score': true},
      {'answerText': '55km, so 34 miles long.', 'score': false},
      {'answerText': '90km, so 56 miles long.', 'score': false},
    ],
  },
  {

```

```

    'question':
      'In which month does the German festival of Oktoberfest mostly take
place?',
    'answers': [
      {'answerText': 'January', 'score': false},
      {'answerText': 'October', 'score': false},
      {'answerText': 'September', 'score': true},
    ],
  },
  {
    'question': 'Who composed the music for Sonic the Hedgehog 3?',
    'answers': [
      {'answerText': 'Britney Spears', 'score': false},
      {'answerText': 'Timbaland', 'score': false},
      {'answerText': 'Michael Jackson', 'score': true},
    ],
  },
  {
    'question': 'In Georgia (the state), it's illegal to eat what with a
fork?',
    'answers': [
      {'answerText': 'Hamburgers', 'score': false},
      {'answerText': 'Fried chicken', 'score': true},
      {'answerText': 'Pizza', 'score': false},
    ],
  },
  {
    'question':
      'Which part of his body did musician Gene Simmons from Kiss insure for
one million dollars?',
    'answers': [
      {'answerText': 'His tongue', 'score': true},
      {'answerText': 'His leg', 'score': false},
      {'answerText': 'His butt', 'score': false},
    ],
  },
  {
    'question': 'In which country are Panama hats made?',
    'answers': [
      {'answerText': 'Ecuador', 'score': true},

```

```

        {'answerText': 'Panama (duh)', 'score': false},
        {'answerText': 'Portugal', 'score': false},
    ],
},
{
    'question': 'From which country do French fries originate?',
    'answers': [
        {'answerText': 'Belgium', 'score': true},
        {'answerText': 'France (duh)', 'score': false},
        {'answerText': 'Switzerland', 'score': false},
    ],
},
{
    'question': 'Which sea creature has three hearts?',
    'answers': [
        {'answerText': 'Great White Sharks', 'score': false},
        {'answerText': 'Killer Whales', 'score': false},
        {'answerText': 'The Octopus', 'score': true},
    ],
},
{
    'question': 'Which European country eats the most chocolate per capita?',
    'answers': [
        {'answerText': 'Belgium', 'score': false},
        {'answerText': 'The Netherlands', 'score': false},
        {'answerText': 'Switzerland', 'score': true},
    ],
},
];

```

#### 4.2.3 answer.dart

```

import 'package:flutter/material.dart';

```

```

class Answer extends StatelessWidget {

  final String answerText;

  Answer({this.answerText, this.answerColor, this.answerTap});

  final Color answerColor;

  final Function answerTap;

  @override

  Widget build(BuildContext context) {

    return InkWell(

      onTap: answerTap,

      child: Container(

        width: double.infinity,

        margin: EdgeInsets.symmetric(vertical: 5.0, horizontal: 30.0),

        padding: EdgeInsets.all(15.0),

        decoration: BoxDecoration(

          color: answerColor,

          border: Border.all(color: Colors.blue),

          borderRadius: BorderRadius.circular(20.0),

        ),

        child: Text(

          answerText,

```

```
        style: TextStyle(
          fontSize: 15.0,
        ),
      ),
    ),
  );
}
```



# 5. Testing

## 5.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

- A software configuration that includes a software requirement specification, a design specification and source code.
- A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

## 5.2 Levels of Testing

### 5.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working correct or not. In this part the main intention is to check the each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program is taking the inputs in the correct format or not.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

### **5.2.2 Integration Testing**

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs.

### **5.2.3 System Testing**

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

### **5.2.4 Validation Testing**

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

### **5.2.5 Output Testing**

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

### **5.2.6 User Acceptance Testing**

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment.

UAT is done in the final phase of testing.

## 6. Results

### 6.1 Questions Screen

Quiz app

localhost:50793/#/

Quiz App

How long is New Zealand's Ninety Mile Beach?

☐ 88km, so 55 miles long.

☐ 55km, so 34 miles long.

☐ 90km, so 56 miles long.

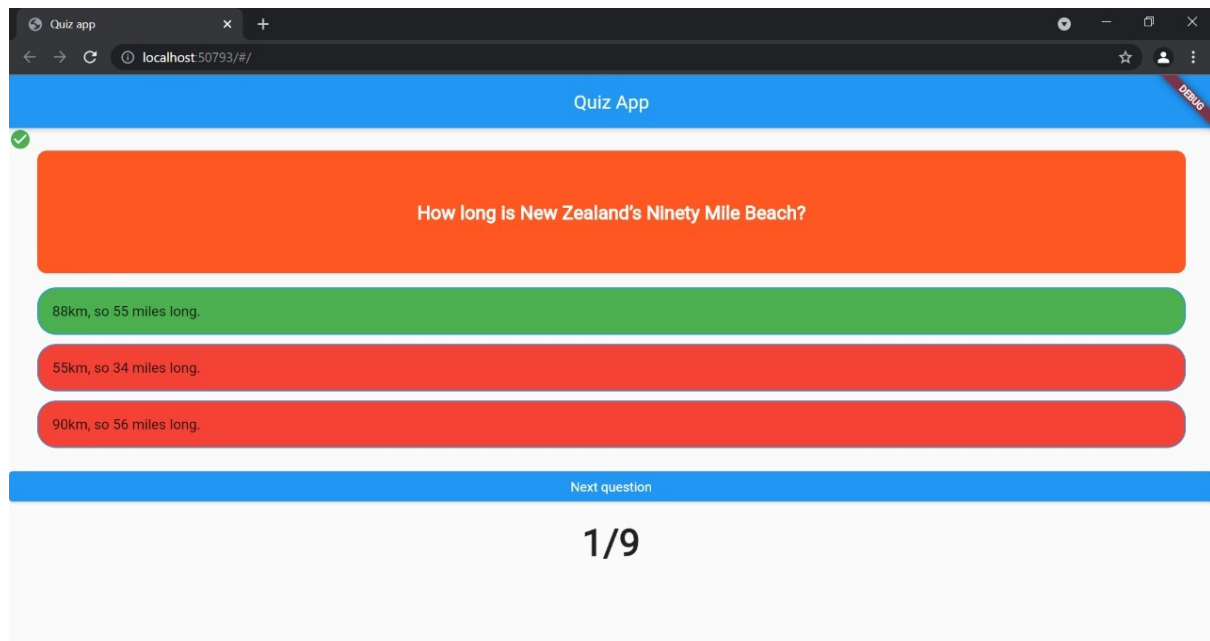
Next question

0/9

6.1 Questions will be shown one-by-one. Score is initially zero. Click on “Next question” to go to the next question.

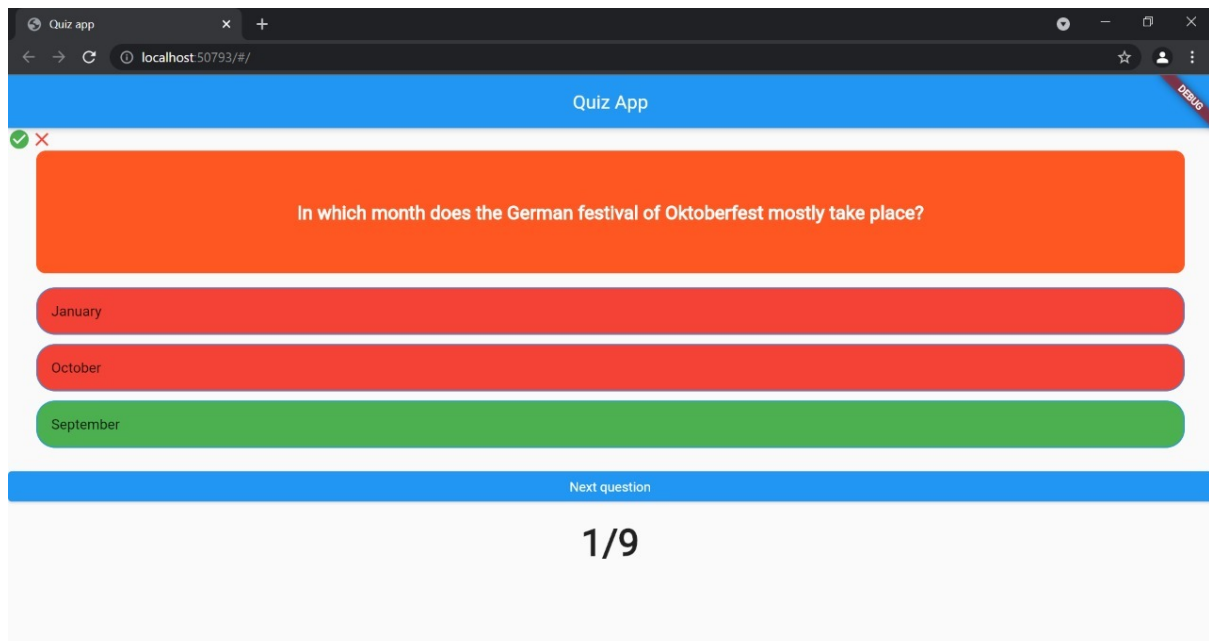
## 6.2 Evaluation of answers – true and false

### 6.2.1 Correct answer



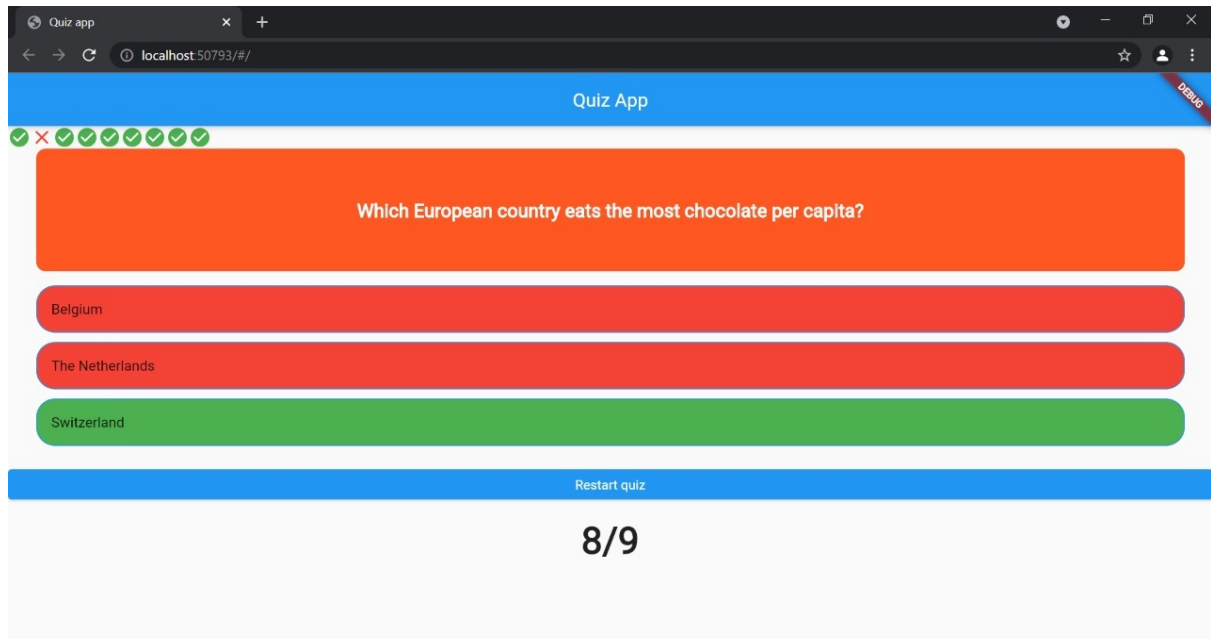
6.2.1 Selected answer is correct and is shown in green. A tick mark is displayed above to show that the selected answer is correct. Scores are updated automatically.

## 6.2.2 Wrong answer



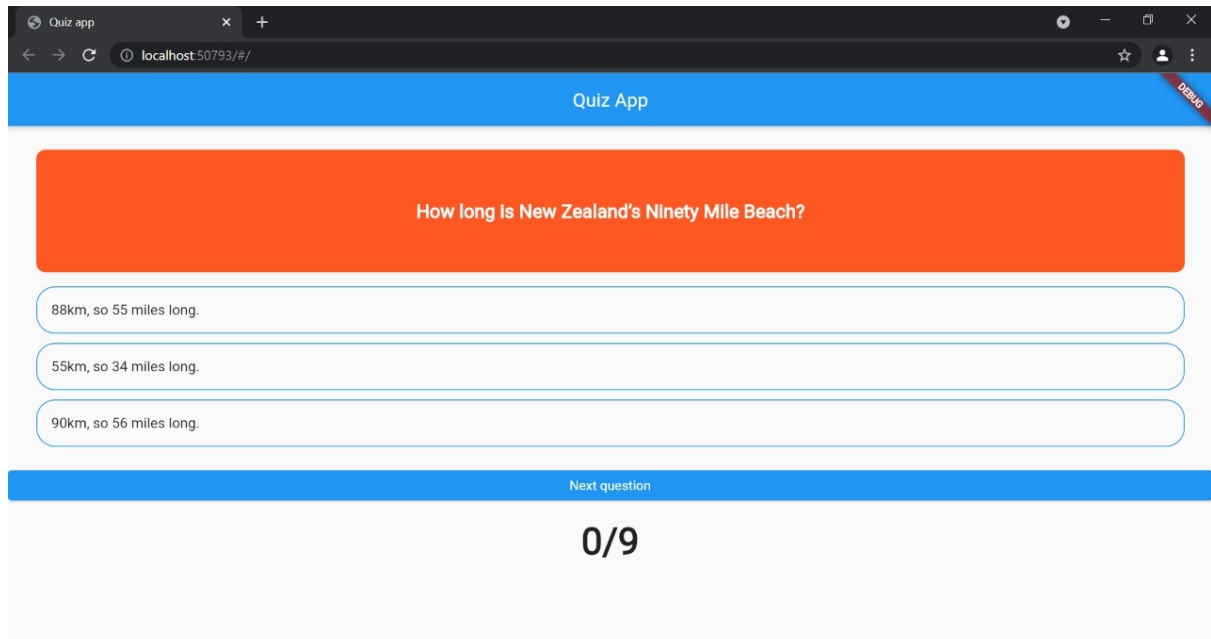
6.2.2 Selected answer is wrong and is shown in red. An 'x' mark is displayed above to show that the selected answer is correct. Scores will not be updated for wrong answers.

## 6.3 Displaying the final score



6.3 Final score is displayed. “Restart quiz” option is shown. The evaluated answers’ symbols are displayed above.

## 6.4 Restart quiz



6.4 Once “Restart quiz” is selected, score is reverted back to zero, all the symbols are deleted and the questions are displayed once again.

## **7. Conclusion and Future work**

### **7.1 Conclusion**

The main purpose of our project is to develop an application that offers new aspects of learning and improving knowledge in the educational area.

Nowadays, most of the students have smart phones and tablets, which can support applications (apps) for mobile devices. Therefore, to make a profitable use of them in order to review the concepts covered in class we have implemented this application.

The main objective of this research is to study the use of mobile apps as supporting tools in the education context

We have learned a lot about Android based applications and also about user's behaviour. We have found that the development process is hard and time-consuming, but it can be managed by a team work. We hope that other developers will take advantage from our experience/from our development.

### **7.2 Future enhancements**

Future enhancements include:

- **HINT button:** - When the user taps/clicks on "Hint", it will show hints on the question by providing some additional information regarding the topic which will help the user to answer the question correctly.
- **ELIMINATE:** - It will help the user to answer question whenever he/she have a confusion on either of the two options. It helps by disabling two extra options from four options, then user can easily select an option from the two.
- **SKIP button:** - Skip options can be used as per user's choice. This function will simply skip the current question and proceed to the next one.
- **TIMER:** - The timer can be added so that user can practice to answer a particular question within the set time for each and every question. When the user wants some time to answer a question, then he/she will use this function to pause the quiz. As a result, the clock will be stopped for 30 sec and this time is not be added in the total time.



## References

- <https://flutter.dev/>
- <https://developers.google.com/learn/pathways/intro-to-flutter>
- <https://www.youtube.com/>
- Beginning Flutter: A Hands On Guide to App Development by Marco L Napoli