

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

By

**JEBIN EARNEST A
REG. NO. 39120046**



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **JEBIN EARNEST A.(Reg. No: 39120046)** who carried out the project entitled “**Quiz Application**” under my supervision from 03 -02-2022 to 06-04-2022.

Internal Guide

Dr.L.MARY GLADENCE M.E., Ph.D.,

Head of the Department

Dr.R.SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **JEBIN EARNEST A.** Thereby declare that the project report entitled **Quiz Application** done by me under the guidance of **Dr.L.MARY GLADENCE M.E.,Ph.D.**, is in partial fulfillment of the requirements for the award of Bachelor of Information Technology.

DATE: 21/04/2022

PLACE: Chennai

SIGNATURE OF THE CANDIDATE



ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, Dean, School of Computing, **Dr. R. Subhashini, M.E., Ph.D. Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L.MARY GLADENCE M.E.Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology**

TRAINING CERTIFICATE



CERTIFICATE OF APPRECIATION



*AlgoxFusion recognizes the hard work and
dedication of*

Je bin Earnest

*for the successful completion of an online course and a mini-project on
"Flutter App Development" in association with "PRIDE - SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY" from Feb 23, 2022 to April 6, 2022.*

PhanSant.P

CEO, ALGOXFUSION PVT. LTD

Certificate ID: 22SIST323

07/04/2022
ISSUED DATE

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO
1	INTRODUCTION	7
2	OBJECTIVES 2.1 MAIN GOAL 2.2 SCOPE OF THE PROJECT	8
3	FLUTTER APP DEVELOPMENT 3.1 WIDGETS 3.2 GLOBAL FLUTTER USE CASES	13
4	DART PROGRAMMING	13
5	QUIZ APPLICATION 5.1 ENVIRONMENT FRIENDLY 5.2 SELF ASSESSMENT TOOL 5.3 INSTANT GRADING	14
6	CONCLUSION	14
7	REFERENCES	15
8	APPENDIX A.SCREENSHOTS	16

ABSTRACT

Flutter API is an open source software development kit for building UI which is natively complied.

In this app we are going to have the features or modulus mentioned below:

Five multiple choice questions

Four selectable option for each question in the form of buttons except the last one

And based on the final score a remark will be shown at the end of the quiz in addition to the score and restart button.

There are two screens in the app home (where question will be shown) and the result screen(where score and remark will be shown)

CHAPTER 1

1.INTRODUCTION

A quiz is a form of game or mind sport in which players attempt to answer questions correctly about a certain or variety of subjects. According to OED, when it applied specifically to set questions used to evaluate a person's knowledge in an academic context, the word Quiz first appeared in print in 1867.

Quizzes can be used as a brief assessment in education and similar fields to measure growth in knowledge, abilities, or skills. In past days quiz application was conducted manually. To reduce the effort in attending the quiz manually it evolved gradually with the help of technology to quiz application and online Quiz Contest

Quiz Contest is an application developed to conduct a quiz based on time constraints. Quiz Contest system is accessed by entering the username and password which is added to the database.

Before the start of the quiz, the rules and regulations are displayed that includes a description of the time limit, number of questions to be answered and scoring methods.

Quiz is started by displaying one question with four options each based on computer

and general knowledge. If the answer is correct, 7 score of 7 is incremented by four and no negative marks for wrong answers . If the time exceeds 20 secs, the next question will come automatically after giving a few limited question's and the quiz application will finally direct you to the score page.

Final score will be displayed and updated in the database with username

2.OBJECTIVES

2.1 MAIN GOAL:

The main goal of the application is to enable users to practice for subjective tests conducted for admissions and recruitment, with focus on the Computer science field.

2.2 SCOPE OF PROJECT:

The scope of "Quiz Contest " is to facilitate a user-friendly environment for all users and reduce manual effort. In the past the quiz was conducted manually but with further resolution of the technology we are able to generate the score and pose the queries automatically.

3.FLUTTER APP DEVELOPMENT

Flutter is a free and open-source mobile UI framework created by Google and

released in May 2017. It is a popular frontend development framework from Google that enables developers to build beautiful frontends for any screen.

Flutter consists of two important parts:

An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).

A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

3.1 WIDGETS

Widgets are the building blocks of the interface of a Flutter app and define the structure of a UI element. A Flutter application interface is a combination of nested widget

StatelessWidget — A widget that depends only on its configuration, and the state doesn't update.

StatefulWidget — A widget that allows the state to change dynamically.

Widgets implement Google's material design, and Cupertino Widgets will give the native iOS experience. It doesn't mean that you have to maintain two different codebases.

Material Additionally, Flutter provides platform-specific widgets. For example, material Design can be used across all platforms.

3.2 GLOBAL FLUTTER USE CASE

Today Flutter is used globally across industries, platforms, and Operating Systems, mainly due to its cross-platform support. Teams have identified Flutter to be time and cost-saving as they can use a single codebase to develop across platforms.

Toyota powering their infotainment systems with Flutter shows the ability of Flutter inside embedded systems.

4. DART PROGRAMMING :

One of the most interesting features of Flutter is the language it uses: Dart. Like other systems that use reactive views, Flutter refreshes the view tree for every new frame. To

accomplish that, it creates many objects that may live for no more than one frame. Dart uses generational garbage collection that has proven to be very efficient for this type of system.

Moreover, Dart has a “tree shaking” compiler that only includes the code you need in your app. Even if you need just a widget or two, you can use its large library of widgets freely. Dart is object-oriented and compiles into native code or JavaScript, making it useful to develop faster apps. A core feature of Flutter is the usage of widgets to structure the UI (user interface)

Finally, Dart comes with a repository of software packages for extending the capabilities of apps. For example, it offers a few packages that help to access Firebase so that developers can build serverless apps. Another package allows accessing a Redux data store or makes it easier to access platform services and hardware like the camera.

5.Quiz Application

Quiz Contest is an android application that has general questions related to current affairs and computers. It has multiple choice questions with a time limit and it also calculates scores of each correct answer. It is good for students of every age group. It

helps in increasing general knowledge about the world ,Sports and computers etc. Don't need to register, simply give any username and password it will save automatically and you can login again with the same user name and password without having to worry about the past score.

The application helps the user to increase his/her knowledge. Since Smartphone mobiles are being widely used by general population and students, the Quiz Contest application can provide on the Student's mobile. This application provides facility to play quiz, aptitude and general knowledge provides good platform where students can improve their knowledge and skill at the same time

5.1 ENVIRONMENT FRIENDLY

Going from hard-copy tests/ quizzes to offering the same capabilities online reduces consumption of goods such as paper – especially important when the online classroom is large and growing!

5.2 SELF ASSESSMENT TOOL

Testing and quizzing online will usually provide the user with results instantly. This is good for students because it allows them to know what they did wrong immediately, what they need to focus on, and how to improve should they have to retake the test.

5.3 INSTANT GRADING

Grading and giving feedback is probably the most time-consuming task for the instructor. It's where the instructor has the ability to comment on the strengths and weaknesses of a learner and enable learning to actually take place

6.CONCLUSION

This quiz application saves time since the user does not need to wait for results. All students/users can get extra knowledge and skills. Administrator has a privilege to put as

many questions in any category. User can give the register, log in and give the test with his/her specific id and can see the results as well

The project has been successfully completed by having established the users with the help of Android Studio tool. It consists of design plots, layouts plots, java codes, oncreate plots and computing plots. At the same time there is some scope for improvement in the future. It can be possible to make it more users friendly by adding more variety of functions to it

SOURCE CODE:

```
Import 'package:flutter/material.dart';  
import 'package:get/get_navigation/src/root/get_material_app.dart';
```

```
import 'package:quiz_app/screens/welcome/welcome_screen.dart';

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

```
class MyApp extends StatelessWidget {
  // This widget is the root of your application.

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Quiz App',
      debugShowCheckedModeBanner: false,
      theme: ThemeData.dark(),
      home: WelcomeScreen(),
    );
  }
}
```

main.dart

```
// This is a basic Flutter widget test.
//
// To perform an interaction with a widget in your test, use the WidgetTester
// utility that Flutter provides. For example, you can send tap and scroll
// gestures. You can also use WidgetTester to find child widgets in the widget
// tree, read text, and verify that the values of widget properties are correct.

import 'package:flutter/material.dart';
import 'package:flutter_test/flutter_test.dart';
import 'package:quiz_app/main.dart';

void main() {
  testWidgets('Counter increments smoke test', (WidgetTester tester) async {
    // Build our app and trigger a frame.
    await tester.pumpWidget(MyApp());

    // Verify that our counter starts at 0.
```

```
expect(find.text('0'), findsOneWidget);
expect(find.text('1'), findsNothing);
// Tap the '+' icon and trigger a frame.
await tester.tap(find.byIcon(Icons.add));
await tester.pump();
// Verify that our counter has incremented.
expect(find.text('0'), findsNothing);
expect(find.text('1'), findsOneWidget);
});
```

widgettest.dart

```
import 'package:flutter/material.dart';
const kSecondaryColor = Color(0xFF8B94BC);
const kGreenColor = Color(0xFF6AC259);
const kRedColor = Color(0xFFE92E30);
const kGrayColor = Color(0xFFC1C1C1);
const kBlackColor = Color(0xFF101010);
const kPrimaryGradient = LinearGradient(
  colors: [Color(0xFF46A0AE), Color(0xFF00FFCB)],
  begin: Alignment.centerLeft,
  end: Alignment.centerRight,
);
```

```
const double kDefaultPadding = 20.0;
```

constant.dart

quizscreen.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:quiz_app/constants.dart';
import 'package:quiz_app/controllers/question_controller.dart';
import 'package:flutter_svg/svg.dart';
class ScoreScreen extends StatelessWidget {
  @override
```

```

Widget build(BuildContext context) {
  QuestionController _qnController = Get.put(QuestionController());
  return Scaffold(
    body: Stack(
      fit: StackFit.expand,
      children: [
        SvgPicture.asset("assets/icons/bg.svg", fit: BoxFit.fill),
        Column(
          children: [
            Spacer(flex: 3),
            Text(
              "Score",
              style: Theme.of(context)
                .textTheme
                .headline3

```

score.dart

```

import 'package:flutter/material.dart';
import 'package:flutter_svg/svg.dart';
import 'package:get/get.dart';
import 'package:quiz_app/constants.dart';
import 'package:quiz_app/screens/quiz/quiz_screen.dart';
class WelcomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          SvgPicture.asset("assets/icons/bg.svg", fit: BoxFit.fill),
          SafeArea(
            child: Padding(

```

```
padding: const EdgeInsets.symmetric(horizontal: kDefaultPadding),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Spacer(flex: 2), //2/6
    Text(
      "Let's Play Quiz,",
      style: Theme.of(context).textTheme.headline4.copyWith(
        color: Colors.white, fontWeight: FontWeight.bold),
    ),
    Text("Enter your informations below"),
    Spacer(), // 1/6
    TextField(
      decoration: InputDecoration(
        filled: true,
        fillColor: Color(0xFF1C2341),
        hintText: "Full Name",
        border: OutlineInputBorder(
          borderRadius: BorderRadius.all(Radius.circular(12)),
        ),
      ),
    ),
    Spacer(), // 1/6
    InkWell(
      onTap: () => Get.to(QuizScreen()),
      child: Container(
        width: double.infinity,
        alignment: Alignment.center,
        padding: EdgeInsets.all(kDefaultPadding * 0.75), // 15
        decoration: BoxDecoration(
          gradient: kPrimaryGradient,
```

```

        borderRadius: BorderRadius.all(Radius.circular(12)),
      ),
      child: Text(
        "Lets Start Quiz",
        style: Theme.of(context)
          .textTheme
          .button
          .copyWith(color: Colors.black),
      ),
    ),
    Spacer(flex: 2), // it will take 2/6 spaces
  ],
),
),
),
],
),
);
}
}

```

welcomescreen.dart

```

class Question {
  final int id, answer;
  final String question;
  final List<String> options;
  Question({this.id, this.question, this.answer, this.options});
}

const List sample_data = [
  {
    "id": 1,

```

```

"question":
  "Flutter is an open-source UI software development kit created by __",
"options": ['Apple', 'Google', 'Facebook', 'Microsoft'],
"answer_index": 1,
},
{
  "id": 2,
  "question": "When google release Flutter.",
  "options": ['Jun 2017', 'Jun 2017', 'May 2017', 'May 2018'],
  "answer_index": 2,
},
{
  "id": 3,
  "question": "A memory location that holds a single letter or number.",
  "options": ['Double', 'Int', 'Char', 'Word'],
  "answer_index": 2,
},
{
  "id": 4,
  "question": "What command [9:45 AM, 4/28/2022] Dany Clg: do you use to output data
to the screen?",
  "options": ['Cin', 'Count>>', 'Cout', 'Output>>'],
  "answer_index": 2,
},
];

```

[9:45 AM, 4/28/2022] Dany Clg: questions.dart

[9:46 AM, 4/28/2022] Dany Clg: ethan da questions

[9:46 AM, 4/28/2022] Dany Clg: import 'package:flutter/widgets.dart';

import 'package:get/get.dart';

import 'package:get/state_manager.dart';

import 'package:quiz_app/models/Questions.dart';

```
import 'package:quiz_app/screens/score/score_screen.dart';

// We use get package for our state management
class QuestionController extends GetxController
  with SingleGetTickerProviderMixin {
  // Lets animated our progress bar
  AnimationController _animationController;
  Animation _animation;
  // so that we can access our animation outside
  Animation get animation => this._animation;
  PageController _pageController;
  PageController get pageController => this._pageController;
  List<Question> _questions = sample_data
    .map(
      (question) => Question(
        id: question['id'],
        question: question['question'],
        options: question['options'],
        answer: question['answer_index']),
    )
    .toList();
  List<Question> get questions => this._questions;
  bool _isAnswered = false;
  bool get isAnswered => this._isAnswered;
  int _correctAns;
  int get correctAns => this._correctAns;
  int _selectedAns;
  int get selectedAns => this._selectedAns;
  // for more about obs please check documentation
  RxInt _questionNumber = 1.obs;
  RxInt get questionNumber => this._questionNumber;
  int _numOfCorrectAns = 0;
```

```

int get numOfCorrectAns => this._numOfCorrectAns;
// called immediately after the widget is allocated memory
@override
void onInit() {
    // Our animation duration is 60 s
    // so our plan is to fill the progress bar within 60s
    _animationController =
        AnimationController(duration: Duration(seconds: 60), vsync: this);
    _animation = Tween<double>(begin: 0, end: 1).animate(_animationController)
        ..addListener(() {
            // update like setState
            update();
        });
    // start our animation
    // Once 60s is completed go to the next qn
    _animationController.forward().whenComplete(nextQuestion);
    _pageController = PageController();
    super.onInit();
}
// // called just before the Controller is deleted from memory
@override
void onClose() {
    super.onClose();
    _animationController.dispose();
    _pageController.dispose();
}
void checkAns(Question question, int selectedIndex) {
    // because once user press any option then it will run
    _isAnswered = true;
    _correctAns = question.answer;
    _selectedAns = selectedIndex;
}

```

```

if (_correctAns == _selectedAns) _numOfCorrectAns++;
// It will stop the counter
_animationController.stop();
update();
// Once user select an ans after 3s it will go to the next qn
Future.delayed(Duration(seconds: 3), () {
    nextQuestion();
});
}

void nextQuestion() {
    if (_questionNumber.value != _questions.length) {
        _isAnswered = false;
        _pageController.nextPage(
            duration: Duration(milliseconds: 250), curve: Curves.ease);
        // Reset the counter
        _animationController.reset();
        // Then start it again
        // Once timer is finish go to the next qn
        _animationController.forward().whenComplete(nextQuestion);
    } else {
        // Get package provide us simple way to naviigate another page
        Get.to(ScoreScreen());
    }
}

void updateTheQnNum(int index) {
    _questionNumber.value = index + 1;
}

```

}