

I B. Tech - II Semester Idea Creation-2

“QUIZ BUZZ”

Department of CSE-Cyber Security

By

B. Bhavana	-	2311CS040020
K. Pardhu	-	2311CS040089
A. Vyshnavi	-	2311CS040014
G. Sadhvik	-	2311CS040062
E. Vivek	-	2311CS040051

Under the Esteemed Guidance of

Mrs. M.V. Sucharitha

Assistant Professor - CSE



Malla Reddy University

Maisammaguda, Kompally, Hyderabad- 500100, Telangana State.

(Telangana State Private Universities Act No. 13 of 2020 & G. O.Ms.No.14,
Higher Education (UE) Department)



MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No. 13 of 2020 &
G.O.Ms.No. 14, Higher Education (UE) Department)

Maisammaguda, Kompally,
Hyderabad - 500100,
Telangana State.

Department of CSE-Cyber Security

CERTIFICATE

This is to certify that the Idea Creation report entitled **“QUIZ BUZZ”** By **B. Bhavana (2311CS040025), K. Pardhu (2311CS040089), A. Vyshnavi (2311CS040014), G. Sadhvik (2311CS040062), E. Vivek (2311CS040050)**, CSE-Cyber Security, Malla Reddy University, Hyderabad was submitted in partial fulfillment of the requirements for the completion of the course during the academic year 2023-2024, is a bonafide work carried out under our guidance and supervision.

Internal Guide

Mrs. M.V. Sucharitha

**Idea Creation
Coordinator**

Mr. T. A. Joseph Judson

HOD

Dr.I. Sreenath

External Examiner

Acknowledgment

We have been truly blessed to have a wonderful internal guide **Ms. M.V. Sucharitha, Assistant Professor, Department of CSE-CS, Malla Reddy University**, for guiding us to explore the ramification of our work and we express our sincere gratitude towards him for leading me through the completion of Project.

We would like to express our gratitude to **Mr. T. A. Joseph Judson, Assistant Professor, Department of CSE, Idea Creation Coordinator, Malla Reddy University**, for providing seamless support and right suggestions are given in the development of the application.

We would like to say our special thanks to **Dr.I. Sreenath, Associate Professor and Incharge HOD, I B. Tech, Malla Reddy University**, for providing seamless support and right suggestions are given in the development of the application.

We would like to convey our heartfelt gratitude to **Mrs. Lakshmi. T. K, Incharge HOD & Assistant Professor, Department of CSE, I B. Tech, Malla Reddy University**, for providing seamless support and right suggestions in the development of the application.

We would like to express our heartfelt thanks to **Dr. V. Dhanunjana Chari, Dean I B. Tech, SOE & SOS, Malla Reddy University**, for providing us with the conducive environment for carrying out academic schedules and project with ease.

We are grateful to **Vice Chancellor, Chancellor and The Management Malla Reddy University**, for providing excellent infrastructure and their visionary thoughts to prepare ourselves industry ready by focusing on new technologies.

Finally, we would like to thank our family members and friends for their moral support and encouragement to achieve goals.

Bhavana	- 2311CS040020
Pardhu	- 2311CS040089
Vyshnavi	- 2311CS040014
Vivek	- 2311CS040050
Sadhvik	- 2311CS040062

ABSTRACT

The QUIZ BUZZ generates new questions and the levels of critical thinking problems is functioned although u can set the difficulty and you can choose interested yet desired topic so that one can focus on the particular subject. We additionally have an option of choosing all subject questions in a single quiz and grades will be stored so that improvement can be regulated on regular basis.

The quiz buzz also incorporates diverse question formats, including multiple-choice, true/false, and open-ended questions, fostering a comprehensive understanding of the subject matter. Additionally, it offers instant feedback and detailed explanations, promoting active learning and addressing misconceptions.

This application is not only a valuable tool for educators seeking to personalize assessments but also serves as a self-paced learning platform for individuals. By harnessing the power of adaptability and artificial intelligence, the quiz buzz revolutionizes traditional quiz formats, making learning more efficient, enjoyable, and tailored to the unique needs of each user.

This project is basically a collection of number of different types of quizzes like technical, games, sports, etc. A user can access/play all of the quiz and can attempt any of the one. There will be limited number of questions and for each correct answer user will get a credit score. User can see answers as well as can ask a query related to it.

INDEX

CHAPTER NO.	TITLE	PAGE NO.
1	Introduction	1
1.1	Summary of application	2
1.2	Background of application	2-3
2	Literature Survey and Existing System	4
2.1	Existing System	4
2.2	Proposed System and Advantages	4-5
3	Software and Hardware Requirements	6
3.1	Software Requirements	6-8
3.2	Hardware Requirements	8
4	Design-algorithm or flow chart	9-11
4.1	Screen Shots of App	12-16
5	Writing Your Code	17-53
5.1	Handling Errors and Test Cases	53-54
6	Conclusion	55
7	Future Scope	56
	References	57

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
Fig.1	Dataflow diagram of quiz buzz	9
Fig.2	Architecture diagram of quiz buzz	10
Fig.3	Screen shot for login page	12
Fig.4	Screen shot for register page	13
Fig.5	Screen shot for home page	14
Fig.6	Screen shot for subject page	15

Chapter -1

INTRODUCTION

In a world where knowledge is power and learning is continuous, Quiz Buzz emerges as a beacon of interactive education and entertainment. Developed with a passion for fostering curiosity and engaging minds, Quiz Buzz is not just another quiz application; it's a dynamic platform designed to revolutionize the way we learn and challenge ourselves.

In today's fast-paced digital landscape, traditional methods of learning often struggle to captivate our attention. Textbooks and lectures can feel dull and disconnected from our daily lives. Recognizing this gap, Quiz Buzz bridges the divide between education and entertainment by offering a captivating quiz experience that is both enriching and enjoyable. With Quiz Buzz, learning becomes an adventure. Whether you're a trivia enthusiast seeking to test your knowledge, a student looking for a fun way to study, or simply someone who loves a good challenge, Quiz Buzz has something for everyone. From history buffs to science aficionados, from pop culture junkies to sports fanatics, there's a quiz for every interest and passion.

What sets Quiz Buzz apart is its commitment to innovation and user engagement. We understand that learning is most effective when it's interactive and personalized. That's why Quiz Buzz allows users to create custom quizzes tailored to their interests and skill levels. Whether you want to focus on a specific topic, challenge yourself with difficult questions, or compete with friends in multiplayer mode, the power is in your hands.

But Quiz Buzz is more than just a quiz platform; it's a community of learners united by a shared love for knowledge and discovery. With features like leaderboards, achievements, and real-time multiplayer competitions, Quiz Buzz brings people together in a spirit of friendly competition and camaraderie.

In this introduction, we invite you to embark on a journey of learning and fun with Quiz Buzz. Join us as we explore the endless realms of trivia, unlock new achievements, and challenge ourselves to reach new heights of knowledge.

1.1 Summary of Application

Quiz Buzz is a cutting-edge quiz application that combines education with entertainment to create a dynamic and engaging learning experience. With its user-friendly interface and diverse features, Quiz Buzz offers something for everyone, whether you're a trivia enthusiast, a student, or simply someone who enjoys a good challenge.

The app allows users to create custom quizzes tailored to their interests and skill levels, covering a wide range of topics from history and science to pop culture and sports. Multiplayer mode enables users to compete against friends or random opponents in real-time quizzes, adding a social and competitive element to the experience.

Quiz Buzz incorporates various question formats, including multiple-choice, true/false, fill-in-the-blank, and picture-based questions, keeping users engaged and challenged. Leaderboards and achievement badges encourage continuous improvement and provide a sense of accomplishment as users track their progress and compare their scores with others. In addition to quizzes, Quiz Buzz provides access to informative articles, videos, and explanations related to each topic, enriching the learning experience with interactive resources. With its sleek interface and intuitive design, Quiz Buzz makes learning fun, accessible, and interactive for users of all ages and backgrounds.

1.2 Background of App

The background of Quiz Buzz app originates from a recognition of the evolving landscape of education and entertainment in the digital age. The developers noticed a growing demand for interactive and engaging learning platforms that cater to diverse interests and learning styles. They observed that traditional methods of learning often struggle to capture and maintain the attention of modern audiences, particularly in an era dominated by smartphones and on-the-go lifestyles.

Driven by a passion for knowledge and a desire to make learning fun and accessible, the creators set out to develop a quiz application that would bridge the gap between education and entertainment. They envisioned an app that would not only challenge users intellectually

but also provide a platform for social interaction and friendly competition.

Through research and development, the team behind Quiz Buzz identified key features and functionalities that would set their app apart from existing quiz platforms. They focused on creating a user-friendly interface, incorporating diverse question formats, and integrating multiplayer capabilities to enhance the social aspect of the app.

Furthermore, the developers recognized the importance of customization and personalization in the learning experience. They understood that users have different interests, preferences, and levels of expertise, and sought to empower them to create custom quizzes tailored to their individual needs.

With these principles in mind, Quiz Buzz was born—a dynamic quiz application designed to engage, entertain, and educate users of all ages and backgrounds. Its development represents a commitment to innovation, user engagement, and the belief that learning should be a rewarding and enjoyable experience.

Chapter -2

Literature Survey and Existing Systems

The documentary "Quiz Show" delves into the captivating yet scandalous history of quiz shows in the 1950s United States, unveiling the darker truths behind the seemingly innocent world of televised trivia. Through interviews, archival footage, and meticulous research, the documentary sheds light on the manipulation, deception, and betrayal that tainted the integrity of these shows, ultimately leading to widespread public outrage and regulatory reforms.

Meanwhile, online platforms like Mental Floss, BBC, and BuzzFeed offer a treasure trove of articles exploring various facets of quiz shows. From iconic moments in quiz show history to intriguing trivia facts and exclusive behind-the-scenes insights into popular programs, these articles provide a fascinating glimpse into the world of televised trivia, captivating audiences with stories of triumph, controversy, and intrigue.

Podcasts such as "Good Job, Brain!" and NPR's "Ask Me Another" offer immersive audio experiences for trivia enthusiasts. Through lively discussions, entertaining trivia games, and insightful interviews with contestants and industry insiders, these podcasts celebrate the art of quizzing while unravelling the mysteries and myths surrounding quiz shows. Whether listeners seek to test their knowledge, learn about the history of trivia, or simply enjoy a good story, these podcasts deliver an engaging and educational journey into the realm of quizzes.

2.1 Existing System

The existing quiz generator app allows users to create quizzes manually by inputting questions, answers, and other relevant details. However, it lacks automation and customization features, making it time-consuming for users to create quizzes efficiently.

2.2 Proposed System

The proposed quiz generator app will feature an intuitive user interface where users can input topics and desired difficulty levels, and the app will automatically generate quizzes based on predefined question databases. Additionally, it will offer customization options such as adding images, randomizing question order, and generating answer keys. This system aims to streamline the quiz creation process, saving users time and effort while providing engaging and varied quizzes.

Advantages

A Quiz Buzz app presents numerous advantages, making it a popular choice among users seeking entertainment, education, and engagement. Firstly, the app's interactive nature ensures high levels of engagement, as users actively participate in quizzes, keeping them entertained and invested. Secondly, the educational potential of quiz apps cannot be understated, offering users the opportunity to learn new facts, reinforce existing knowledge, or even delve into specialized subjects. Additionally, the competitive elements often incorporated into quiz apps, such as leaderboards and multiplayer modes, foster a sense of challenge and motivation among users. Moreover, the social interaction aspect allows users to compete against friends or other players online, enhancing the overall experience and building a sense of community. Furthermore, features like customization, instant feedback, and progress tracking provide users with a personalized and rewarding experience, catering to their individual preferences and goals. With the accessibility of smartphones and the cost-effectiveness of many quiz apps, users can easily access and enjoy a wide range of content anytime, anywhere. In essence, the versatility and appeal of quiz buzz apps lie in their ability to entertain, educate, and engage users in a dynamic and interactive manner.

Chapter -3

Software and Hardware Requirements

3.1 Software Requirements

Front End: Java script, XML will be used to develop the user interface of the application.

Back End: Java is used as backend for storing and retrieving data.

Markup Language: XML (Extensible markup language) is used as markup language.

Frame Work: Gradle is used as framework.

3.1.1 Java

Java is a widely-used, high-level programming language known for its platform independence and versatility. Developed by Sun Microsystems (now owned by Oracle Corporation) in the mid-1990s, Java has since become one of the most popular programming languages in the world. One of its key features is its "write once, run anywhere" capability, meaning that Java code can be compiled into bytecode that runs on any device with a Java Virtual Machine (JVM), making it highly portable. Java is also known for its strong emphasis on object-oriented programming principles, such as encapsulation, inheritance, and polymorphism. It offers a rich standard library that provides ready-to-use classes and methods for common programming tasks, making development faster and more efficient. Java is commonly used for a wide range of applications, including web development, mobile app development (particularly for Android), desktop applications, enterprise software, and more. Its robustness, security features, and large community of developers contribute to its continued popularity and widespread adoption in the software industry.

3.1.2 Sql Lite

SQLite is a popular and widely-used relational database management system (RDBMS) known for its lightweight nature and ease of integration into applications. Unlike traditional database systems that require a separate server process, SQLite operates in a serverless manner, with the entire database contained within a single file. This makes it incredibly portable and easy to deploy, requiring minimal setup and configuration. Despite its small footprint, SQLite offers powerful features such as ACID compliance, support for SQL queries, transactions, indexes, views, triggers, and more. It is highly versatile and can be used in various applications, including mobile app development, desktop software, embedded systems, and more. Its compatibility with multiple platforms, including Windows, macOS, Linux, Android, and iOS, further contributes to its widespread adoption. With its combination of simplicity, reliability, and performance, SQLite continues to be a popular choice for developers seeking a lightweight and efficient database solution.

3.1.3 Gradle

Gradle isn't exactly a framework in the traditional sense; rather, it's a powerful build automation tool and dependency management system that plays a crucial role in the development workflow of many projects. Developed with a focus on flexibility, efficiency, and extensibility, Gradle has become a cornerstone in the Java ecosystem and beyond.

At its core, Gradle utilizes a declarative approach to build automation, allowing developers to define their project's desired state without getting bogged down in procedural details. Build scripts, typically written in Groovy or Kotlin, provide a clear and concise representation of the project structure, dependencies, tasks, and configurations. This convention-over-configuration philosophy streamlines the build process, making it easy to set up and maintain projects of any size or complexity.

One of Gradle's standout features is its robust dependency management capabilities. By integrating with repositories like Maven Central and Center, Gradle simplifies the process of declaring and resolving project dependencies. Developers can specify dependencies in their build scripts, and Gradle automatically downloads the required libraries, ensuring consistency and reliability across environments.

Gradle's plugin ecosystem further extends its functionality, offering a wide range of plugins for tasks such as code quality analysis, testing, code generation, and deployment. These

plugins, which can be easily added to the build script, enable developers to tailor Gradle to their specific project requirements, enhancing productivity and enabling best practices.

Another key advantage of Gradle is its support for incremental builds and multi-project setups. Incremental builds optimize build times by only rebuilding the parts of the project that have changed since the last build, while multi-project builds facilitate the management of complex project structures with multiple modules or subprojects, promoting code reuse and maintainability.

In summary, Gradle serves as a versatile and efficient tool for automating the build process and managing dependencies in modern software development. Whether used in Java projects, Android app development, or other domains, Gradle empowers developers to streamline their workflows, improve productivity, and deliver high-quality software with confidence.

3.2 Hardware Requirements

Processor: Snapdragon 600 or Mediatek Dimensity Series

RAM: 2GB to 4GB

Storage: 50MB to several GBs

Operating System (OS): iOS and Android

Chapter – 4

DATA FLOW DIAGRAM

4.1 System Architecture

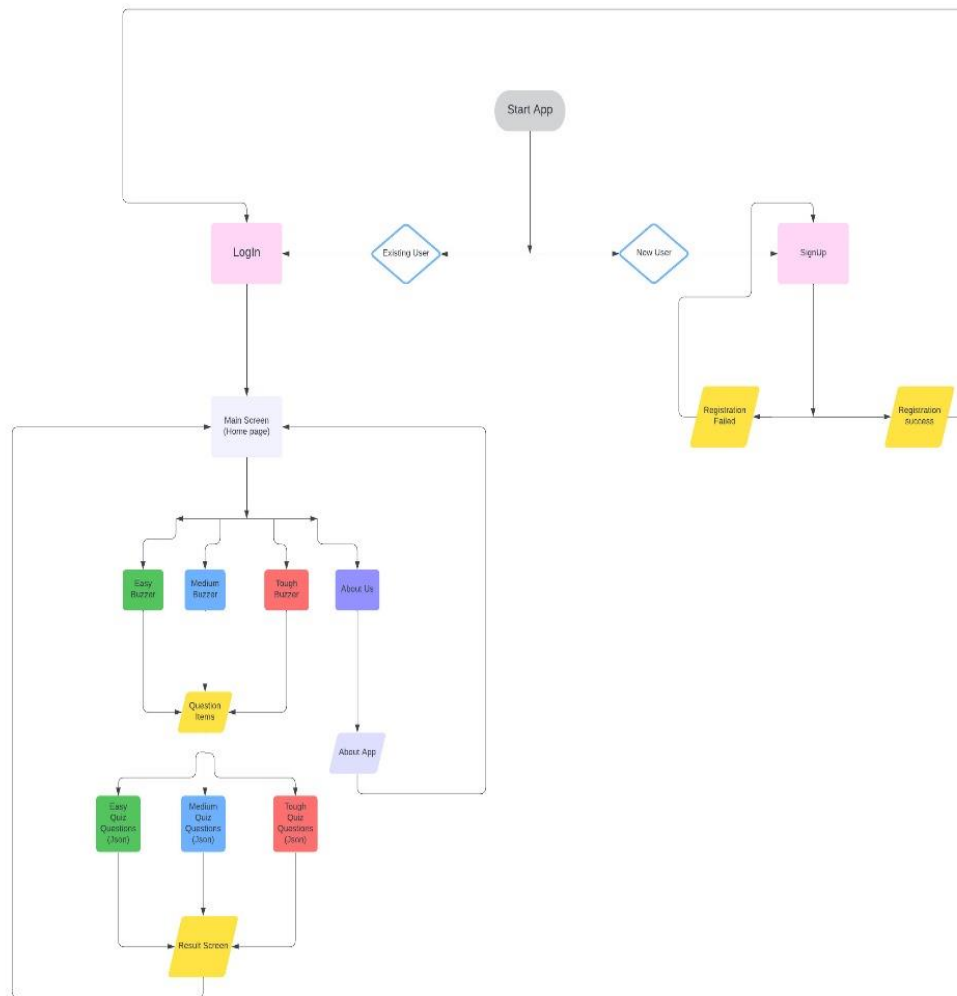


Fig. 1 System Architecture of QUIZ BUZZ

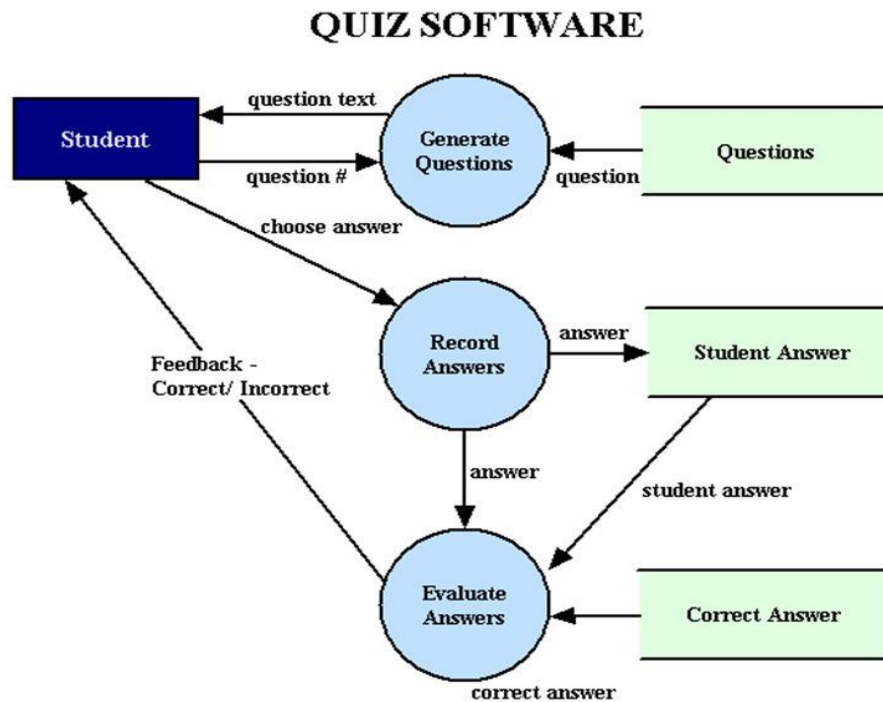


Fig. 2 Database diagram of QUIZ BUZZ

4.2 Algorithm

Here's an algorithm for the QUIZ BUZZ Application:

Algorithm For Quiz Buzz:

Step.1: Display the login/registration screen to the user.

1.1. If the user is new, provide an option to register.

1.2. If the user is already registered, allow them to log in with their credentials.

Step.2: After successful login or registration, present the user with the main menu:

Easy Buzzer (contains easy quiz)

Medium Buzzer (contains medium quiz)

Tough Buzzer (contains hard quiz)

About Us (contains information about the app)

Step.3: For each quiz option, perform the following steps:

- 3.1. Display the selected quiz category to the user.
- 3.2. Present the user with a series of questions, one at a time.
- 3.3. For each question, show the user the available answer options (e.g., A, B, C, D)
- 3.4. Allow the user to select an answer.
- 3.5. After the user selects an answer, reveal whether the answer is correct, incorrect.
- 3.6. Keep track of the user's score.
- 3.7. After all questions have been answered, display the user's score.

Step.4: For the "About Us" option, display information about the app, such as its purpose, features, and development team.

Step.5: Allow the user to exit the app at any time.

Chapter -4

SCREENSHOTS OF APPLICATION

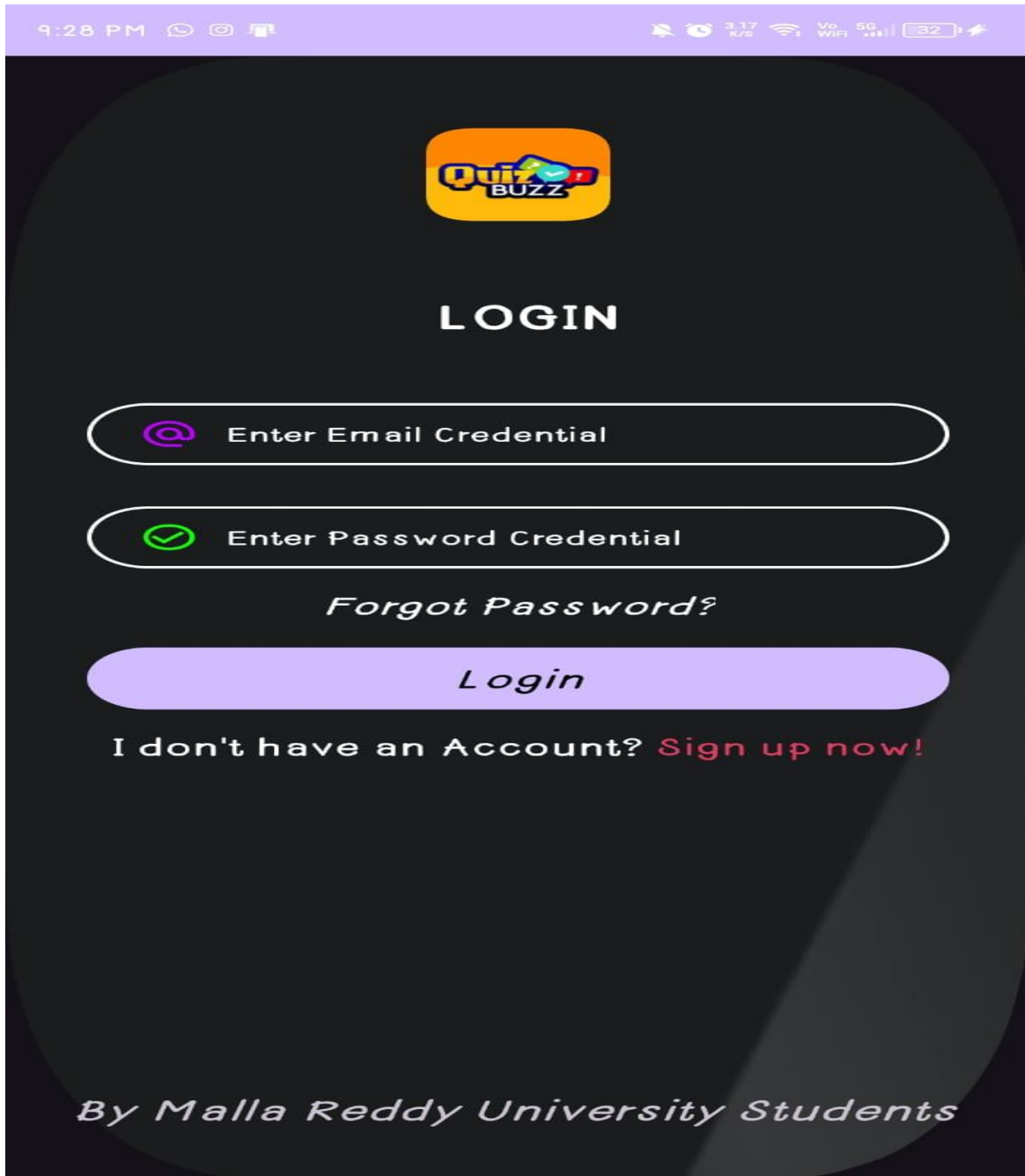





Fig. 3 Login Page


9:28 PM 270 B/S Vo WiFi 5G 32




REGISTRATION

 Enter Unique Username

 Enter Authorised Email Address

 Enter Unique Password

 Re-Enter Your Unique Password

Already Have An Account?

REGISTER

By Malla Reddy University Students

Fig. 4 Register page



Fig. 5 Home page

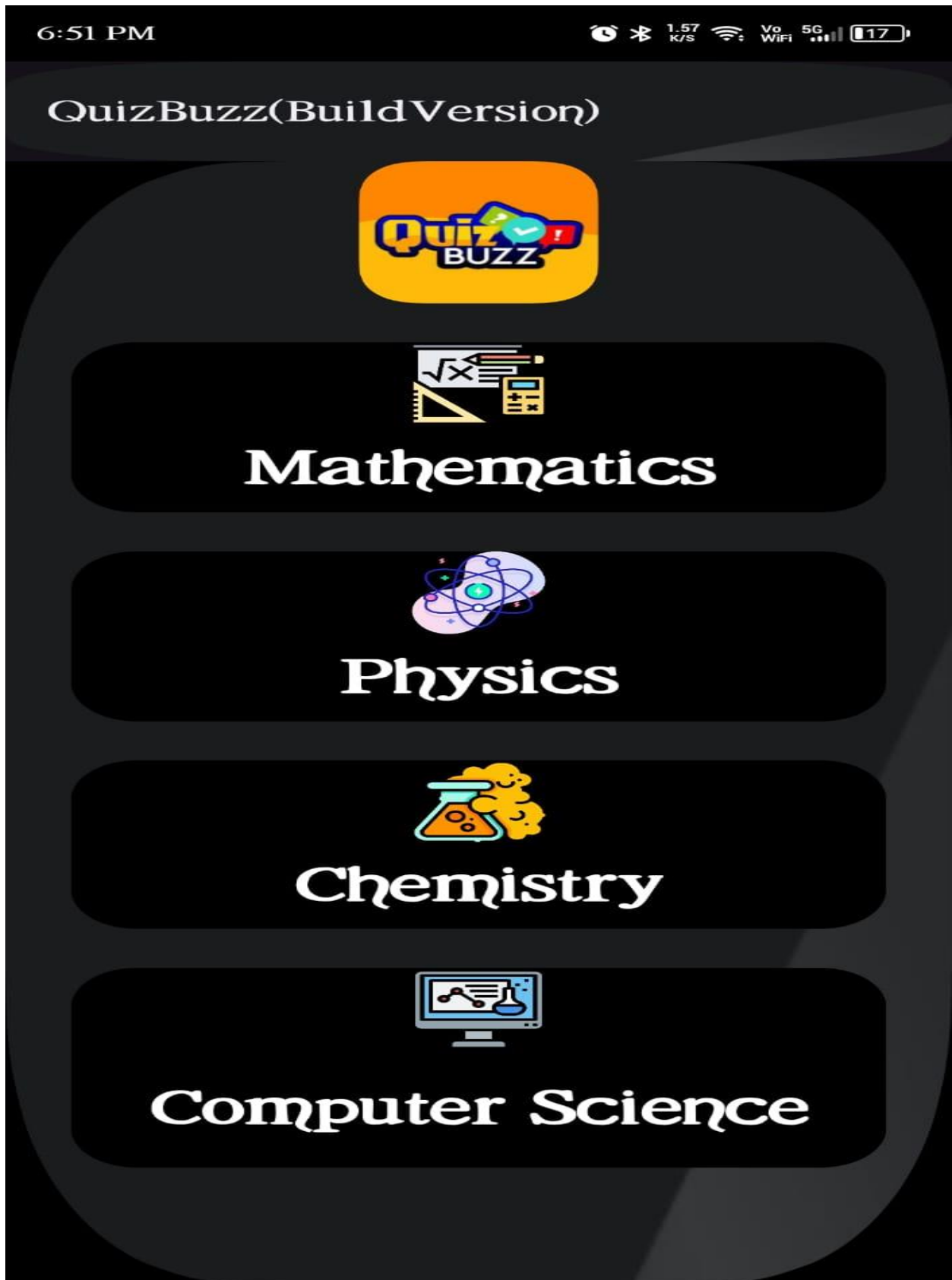


Fig. 6 Subject page

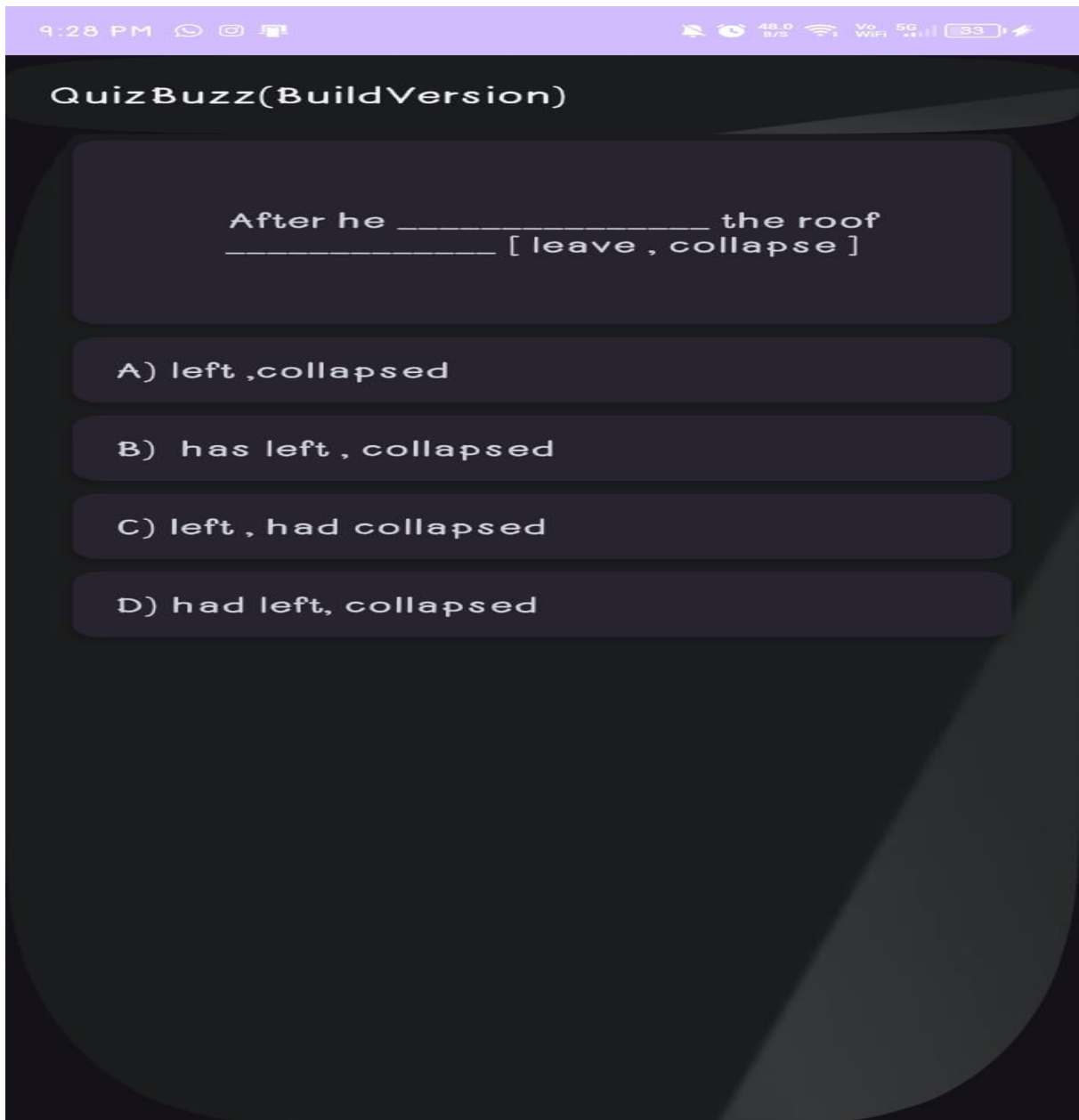


Fig. 7 App page

Chapter – 5

APPLICATION CODE

Frontend code

ACTIVITY LOGIN.java

```
package com.androidapplicationdevelopment.quizbuzzbuildversion;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class LoginActivity extends AppCompatActivity {

    DBHelper DB;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

        TextView btn=findViewById(R.id.textViewSignUp);

        Button btnlogin = (Button) findViewById(R.id.loginbtn);

        DB = new DBHelper(this);

        btn.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                startActivity(new Intent(LoginActivity.this,RegistrationActivity.class));

            }

        });

        btnlogin.setOnClickListener(new View.OnClickListener() {
```

```

@Override

public void onClick(View v) {

    startActivity(new Intent(LoginActivity.this,MainActivity.class));

}

});

btnlogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

    Intent intent = new Intent(getApplicationContext(), MainActivity.class);

    startActivity(intent);

}

});

}

}

```

ACTIVITY LOGIN.XML

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/background3"
    tools:context=".LoginActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```



```

    android:layout_marginTop="56dp"
    android:layout_marginBottom="42dp"
    android:src="@mipmap/applogo"
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="0dp" />

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="320dp"
    android:layout_height="75dp"
    android:layout_marginTop="42dp"
    android:drawablePadding="10dp"
    android:gravity="center"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="20dp"
    android:text="Login"
    android:textAllCaps="true"
    android:textColor="@color/white"
    android:textSize="25dp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    tools:ignore="MissingConstraints" />

```

```

<EditText
    android:id="@+id/inputUsername"
    android:layout_width="0dp"

```

```

android:layout_height="48dp"
android:layout_marginStart="32dp"
android:layout_marginTop="24dp"
android:layout_marginEnd="32dp"
android:background="@drawable/input_bg"
android:drawableLeft="@drawable/ic_email"
    android:drawablePadding="10dp"
    android:drawableTint="#BB00FF"
    android:ems="10"
    android:hint="Enter Email Credential"
    android:inputType="text"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="15dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView"
tools:ignore="TextSizeCheck" />

```

<EditText

```

    android:id="@+id/inputPassword"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:background="@drawable/input_bg"
    android:drawableLeft="@drawable/ic_confirmpasswordcheck"
    android:drawablePadding="10dp"
    android:drawableTint="#11FF00"

```

```

android:ems="10"
android:hint="Enter Password Credential"
android:inputType="text"
android:minHeight="48dp"
android:paddingLeft="20dp"
android:paddingTop="10dp"
android:paddingRight="10dp"
android:paddingBottom="10dp"
android:textColor="@color/white"
android:textColorHint="@color/white"
android:textSize="15dp"
app:layout_constraintEnd_toEndOf="@+id/inputUsername"
app:layout_constraintStart_toStartOf="@+id/inputUsername"
app:layout_constraintTop_toBottomOf="@+id/inputUsername"
tools:ignore="TextSizeCheck" />

```

<TextView

```

android:id="@+id/forgotpassword"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="32dp"
android:layout_marginTop="16dp"
android:layout_marginEnd="32dp"
android:gravity="center"
android:text="Forgot Password?"
android:textColor="@color/white"
android:textSize="19dp"
android:textStyle="italic"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/inputPassword"
app:layout_editor_absoluteX="97dp"
tools:ignore="MissingConstraints" />

```

<Button

```
    android:id="@+id/loginbtn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="32dp"
    android:background="@drawable/btn_bg"
    android:text="Login"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/forgotpassword"
    tools:ignore="MissingConstraints" />
```

<LinearLayout

```
    android:id="@+id/linearLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:gravity="center"
    android:orientation="horizontal"
    android:textAlignment="center"
    app:layout_constraintEnd_toEndOf="@+id/loginbtn"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/loginbtn"
    app:layout_constraintTop_toBottomOf="@+id/loginbtn">
```

<TextView

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center"
android:text="I don't have an Account?"
android:textColor="@color/white"
android:textSize="19dp" />
```

```
<TextView
```

```
    android:id="@+id/textViewSignUp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" Sign up now!"
    android:textColor="@color/red"
    android:textSize="18dp" />
```

```
</LinearLayout>
```

```
<TextView
```

```
    android:id="@+id/byviewtext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="64dp"
    android:text="By Malla Reddy University Students"
    android:textSize="22dp"
    android:textStyle="italic"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.463"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/linearLayout"
app:layout_constraintVertical_bias="0.824" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

REGISTRATION ACTIVITY.java

```
package com.androidapplicationdevelopment.quizbuzzbuildversion;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class RegistrationActivity extends AppCompatActivity {

    DBHelper DB;

    TextView btn;

    private EditText inputUsername, inputPassword, inputEmail, inputConformPassword;

    Button btnRegister;

    @SuppressWarnings("MissingInflatedId")

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_registration);

        btn = findViewById(R.id.alreadyhaveAccount);

        inputUsername = findViewById(R.id.inputUsername);

        inputEmail = findViewById(R.id.inputEmail);

        inputPassword = findViewById(R.id.inputPassword);
```

```

inputConformPassword = findViewById(R.id.inputConformPassword);

DB = new DBhelper(this);

btnRegister = findViewById(R.id.btnRegister);

btnRegister.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        checkCredentials();

        }

    });

btn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        startActivity(new Intent(RegistrationActivity.this, LoginActivity.class));

        }

    });

}

private void checkCredentials() {

    String username = inputUsername.getText().toString();

    String email = inputEmail.getText().toString();

    String password = inputPassword.getText().toString();

    String conformpassword = inputConformPassword.getText().toString();

    if (username.equals("") || password.equals("") || email.equals("") || conformpassword.equals(""))

        Toast.makeText(RegistrationActivity.this, "Please Enter All Fields",

            Toast.LENGTH_SHORT).show();

    else {

```

```

if (password.equals(conformpassword)) {

Boolean checkuser = DB.checkusername(username);

if (checkuser == false) {

Boolean insert = DB.insertData(username, password);

if (insert == true) {

Toast.makeText(RegistrationActivity.this, "Registration Success",
Toast.LENGTH_SHORT).show();

Intent intent = new Intent(getApplicationContext(), LoginActivity.class);

startActivity(intent);

} else {

        Toast.makeText(RegistrationActivity.this, "Registration Failure (Something Went
        Wrong)",

        Toast.LENGTH_SHORT).show();

    }

} else {

Toast.makeText(RegistrationActivity.this, "User Already Exists! Please Login Back",
Toast.LENGTH_SHORT).show();

    }

} else {

        Toast.makeText(RegistrationActivity.this, "Incorrect Password (MissMatching Error)",
        Toast.LENGTH_SHORT).show();

    }

}

}

```


Activity_registration.xml

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/background3"
    tools:context=".RegistrationActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:src="@mipmap/applogo"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="320dp"
        android:layout_height="75dp"
        android:layout_marginTop="32dp"
```

```
android:layout_marginBottom="16dp"
android:drawablePadding="10dp"
android:gravity="center"
android:paddingLeft="20dp"
android:paddingTop="10dp"
android:paddingRight="10dp"
android:paddingBottom="20dp"
android:text="Registration"
android:textAllCaps="true"
android:textColor="@color/white"
android:textSize="25dp"
android:textStyle="italic"
app:layout_constraintBottom_toTopOf="@+id/inputUsername"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView"
tools:ignore="MissingConstraints" />
```

<EditText

```
android:id="@+id/inputUsername"
android:layout_width="0dp"
android:layout_height="48dp"
android:layout_marginStart="32dp"
android:layout_marginTop="16dp"
android:layout_marginEnd="32dp"
android:background="@drawable/input_bg"
```

```
android:drawableLeft="@drawable/ic_person"

android:drawablePadding="10dp"

android:drawableTint="#A600FF"

android:ems="10"

android:textColorHint="@color/white"

android:hint="Enter Unique Username"

android:inputType="text"

android:paddingLeft="20dp"

android:paddingTop="10dp"

android:paddingRight="10dp"

android:paddingBottom="10dp"

android:textColor="@color/white"

android:textSize="15dp"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/textView"

tools:ignore="TextSizeCheck" />
```

<EditText

```
android:id="@+id/inputEmail"

android:layout_width="0dp"

android:layout_height="wrap_content"

android:layout_marginTop="16dp"

android:background="@drawable/input_bg"

android:drawableLeft="@drawable/ic_email"

android:drawablePadding="10dp"
```

```
android:drawableTint="#00FFE1"

    android:ems="10"

    android:textColorHint="@color/white"

    android:hint="Enter Authorised Email Address"

    android:inputType="text"

    android:minHeight="48dp"

    android:paddingLeft="20dp"

    android:paddingTop="10dp"

    android:paddingRight="10dp"

    android:paddingBottom="10dp"

    android:textColor="@color/white"

    android:textSize="15dp"

    app:layout_constraintEnd_toEndOf="@+id/inputUsername"

    app:layout_constraintStart_toStartOf="@+id/inputUsername"

    app:layout_constraintTop_toBottomOf="@+id/inputUsername"

    tools:ignore="TextSizeCheck" />
```

<EditText

```
    android:id="@+id/inputPassword"

    android:layout_width="0dp"

    android:layout_height="wrap_content"

    android:layout_marginTop="16dp"

    android:background="@drawable/input_bg"

    android:drawableLeft="@drawable/ic_passwordcheck"

    android:drawablePadding="10dp"

    android:drawableTint="#FF0000"
```

```

android:ems="10"

    android:textColorHint="@color/white"
    android:hint="Enter Unique Password"
    android:inputType="text"
    android:minHeight="48dp"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    android:textColor="@color/white"
    android:textSize="15dp"
    app:layout_constraintEnd_toEndOf="@+id/inputEmail"
    app:layout_constraintStart_toStartOf="@+id/inputEmail"
    app:layout_constraintTop_toBottomOf="@+id/inputEmail"
    tools:ignore="TextSizeCheck" />

```

<EditText

```

    android:id="@+id/inputConformPassword"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@drawable/input_bg"
    android:drawableLeft="@drawable/ic_confirmpasswordcheck"
    android:drawablePadding="10dp"
    android:drawableTint="#11FF00"
    android:ems="10"

```

```

android:hint="Re-Enter Your Unique Password"

android:inputType="text"

android:minHeight="48dp"

android:paddingLeft="20dp"

android:paddingTop="10dp"

android:paddingRight="10dp"

android:paddingBottom="10dp"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:textSize="15dp"

app:layout_constraintEnd_toEndOf="@+id/inputPassword"

app:layout_constraintStart_toStartOf="@+id/inputPassword"

app:layout_constraintTop_toBottomOf="@+id/inputPassword"

tools:ignore="TextSizeCheck" />

```

<TextView

```

android:id="@+id/alreadyhaveAccount"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginStart="97dp"

android:layout_marginTop="20dp"

android:layout_marginEnd="97dp"

android:layout_marginBottom="23dp"

android:text="Already Have An Account?"

android:textColor="@color/green"

android:textSize="19dp"

```

```
android:textStyle="italic"

app:layout_constraintBottom_toTopOf="@+id/btnRegister"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/inputConformPassword"

tools:ignore="MissingConstraints" />
```

<Button

```
android:id="@+id/btnRegister"

android:layout_width="0dp"

android:layout_height="wrap_content"

android:layout_marginStart="32dp"

android:layout_marginTop="23dp"

android:layout_marginEnd="32dp"

android:layout_marginBottom="167dp"

android:background="@drawable/btn_bg"

android:gravity="center"

android:paddingLeft="20dp"

android:paddingTop="10dp"

android:paddingRight="10dp"

android:paddingBottom="10dp"

android:text="REGISTER"

android:textColor="@color/black"

android:textSize="15dp"

android:textStyle="italic"

app:layout_constraintBottom_toBottomOf="parent"
```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/alreadyhaveAccount"
tools:ignore="TextSizeCheck" />

```

```

<TextView
    android:id="@+id/bytext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="By Malla Reddy University Students"
    android:textColor="@color/white"
    android:textSize="22dp"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnRegister" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MAIN ACTIVITY.java

```

package com.androidapplicationdevelopment.quizbuzzbuildversion;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.DialogInterface;

```



```

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import com.google.android.material.card.MaterialCardView;

import com.google.android.material.dialog.MaterialAlertDialogBuilder;

public class MainActivity extends AppCompatActivity {

    public MaterialCardView easycard, mediumcard, toughcard, aboutcard;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        easycard = (MaterialCardView) findViewById(R.id.easyCard);

        mediumcard = (MaterialCardView) findViewById(R.id.mediumCard);

        toughcard = (MaterialCardView) findViewById(R.id.toughCard);

        aboutcard = (MaterialCardView) findViewById(R.id.aboutCard);

        easycard.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                startActivity(new Intent(MainActivity.this, EasyBuzzer.class));

            }

        });

    }

    @Override

    public void onBackPressed () {

        MaterialAlertDialogBuilder materialAlertDialogBuilder = new

```

```

MaterialAlertDialogBuilder(MainActivity.this);

materialAlertDialogBuilder.setTitle(R.string.app_name);

materialAlertDialogBuilder.setMessage("Are you sure want to exit the app?");

materialAlertDialogBuilder.setNegativeButton(android.R.string.no, new
DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

    dialogInterface.dismiss();

    }

});

materialAlertDialogBuilder.setPositiveButton(android.R.string.yes, new
DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

    finish();

    }

});

materialAlertDialogBuilder.show();

super.onBackPressed();

}

@SuppressWarnings("CustomSplashScreen")

public static class SplashActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_splash);

Thread mSplashThread;

```

```

mSplashThread = new Thread() {

    @Override

    public void run() {

        try {

            synchronized (this) {

                wait(2000);

            }

        } catch (InterruptedException ignored) {

        } finally {

            startActivity(new Intent(getApplicationContext(),
            com.androidapplicationdevelopment.quizbuzzbuildversion.MainActivity.class));

            finish();

        }

    }

};

mSplashThread.start();

}

}

}

```

ACTIVITY MAIN.XML

```

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

```

```

android:layout_width="match_parent"

android:layout_height="match_parent"

android:background="@color/black"

tools:context=".MainActivity">

<androidx.core.widget.NestedScrollView

android:id="@+id/nestedScrollView"

android:layout_width="match_parent"

android:layout_height="0dp"

android:background="@mipmap/background3"

app:layout_behavior="@string/appbar_scrolling_view_behavior"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="1.0"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/appbarLayout">

```

```

<LinearLayout

android:layout_width="match_parent"

android:layout_height="match_parent"

android:orientation="vertical">

<ImageView

android:layout_width="match_parent"

android:layout_height="91dp"

android:layout_gravity="center"

android:src="@mipmap/applogo_round" />

<com.google.android.material.card.MaterialCardView

```

```

        android:id="@+id/easyCard"

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="25dp"
        android:layout_marginEnd="25dp"
        app:cardBackgroundColor="@color/card_background"
        app:cardCornerRadius="25dp"
        app:strokeWidth="0dp">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="112dp"
    android:background="@color/green"
    android:gravity="center"
    android:orientation="vertical">

<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="EasyBuzzer"
    android:textSize="30dp"
    android:gravity="center"
    android:textStyle="bold"
    android:textColor="@color/white"/>
</LinearLayout>
</com.google.android.material.card.MaterialCardView>

```

```

<com.google.android.material.card.MaterialCardView
    android:id="@+id/mediumCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginEnd="25dp"
    app:cardBackgroundColor="@color/card_background"
    app:cardCornerRadius="25dp"
    app:strokeWidth="0dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="109dp"
        android:background="@color/primary_blue"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="MediumBuzzer"
            android:textSize="30dp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/white"/>
        </LinearLayout>
    </com.google.android.material.card.MaterialCardView>

```

```

<com.google.android.material.card.MaterialCardView
    android:id="@+id/toughCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginEnd="25dp"
    app:cardBackgroundColor="@color/card_background"
    app:cardCornerRadius="25dp"
    app:strokeWidth="0dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="108dp"
        android:background="@color/red"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="ToughBuzzer"
            android:textSize="30dp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/white"/>
        </LinearLayout>
    </com.google.android.material.card.MaterialCardView>

```

```

<com.google.android.material.card.MaterialCardView
    android:id="@+id/aboutCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginEnd="25dp"
    app:cardBackgroundColor="@color/card_background"
    app:cardCornerRadius="25dp"
    app:strokeWidth="0dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="128dp"
        android:background="@color/purple_500"
        android:gravity="center"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="About Us"
            android:textSize="30dp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/white"/>
        </LinearLayout>
    </com.google.android.material.card.MaterialCardView>

```



```

</LinearLayout>

</androidx.core.widget.NestedScrollView>

<com.google.android.material.appbar.AppBarLayout
    android:id="@+id/appbarLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fitsSystemWindows="true"
    app:statusBarForeground="@color/black"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="-3dp"
    tools:layout_editor_absoluteY="2dp">
    <com.google.android.material.appbar.MaterialToolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@mipmap/background3"
        android:elevation="0dp"
        app:title="@string/app_name"
        app:titleTextColor="@color/background" />
</com.google.android.material.appbar.AppBarLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

SPLASH ACTIVITY.java

```
package com.androidapplicationdevelopment.quizbuzzbuildversion;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;

public class SplashActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_splash);

        Thread mSplashThread;

        mSplashThread = new Thread() {

            @Override public void run() {

                try {

                    synchronized (this) {

                        wait(2000);

                    }

                } catch (InterruptedException ignored) {

                }

                finally {

                    startActivity(new Intent(getApplicationContext(), RegistrationActivity.class));

                    finish();

                }

            }

        };

    }

};
```

```
mSplashThread.start();  
  
}  
  
}
```

ACTIVITY_SPLASH.XML

```
<?xml version="1.0" encoding="utf-8"?>  
  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="@color/black"  
    tools:context="com.androidapplicationdevelopment.quizbuzzbuildversion.SplashActivity"  
    tools:ignore="MissingClass">  
  
    <androidx.constraintlayout.widget.ConstraintLayout  
        android:id="@+id/backgroundLayout"  
        android:layout_width="200dp"  
        android:layout_height="200dp"  
        app:layout_constraintBaseline_toBottomOf="parent"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
  
    <LinearLayout  
        android:layout_width="match_parent"
```

```

android:layout_height="match_parent"

android:background="@mipmap/background3"

android:gravity="center"

android:orientation="vertical"

app:layout_constraintStart_toEndOf="@+id/backgroundLayout"

tools:layout_editor_absoluteY="16dp">

<ImageView

android:layout_width="100dp"

android:layout_height="100dp"

android:src="@mipmap/applogo"

tools:src="@mipmap/applogo" />

<TextView

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:layout_marginTop="20sp"

android:gravity="center"

android:text="@string/app_name"

android:textColor="@color/white"

android:textSize="20sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

BACK-END CODE

DB HELPER (database)

```
package com.androidapplicationdevelopment.quizbuzzbuildversion;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DBhelper extends SQLiteOpenHelper {

    public static final String DBNAME = "Login.DB";

    public DBhelper(Context context) {

        super(context, "Login.DB", null, 1);

    }

    @Override

    public void onCreate(SQLiteDatabase MyDB) {

        MyDB.execSQL("create Table users(username TEXT primary key, password TEXT)");

    }

    @Override

    public void onUpgrade(SQLiteDatabase MyDB, int oldVersion, int newVersion) {

        MyDB.execSQL("drop Table if exists users");

    }

    public Boolean insertData(String username, String password) {

        SQLiteDatabase MyDB = this.getWritableDatabase();
```

```

ContentValues contentValues = new ContentValues();

contentValues.put("username", username);

contentValues.put("password", password);

long result = MyDB.insert("users", null, contentValues);

if (result == -1) return false;

else

return true;

}

public Boolean checkusername(String username) {

SQLiteDatabase MyDB = this.getWritableDatabase();

Cursor cursor = MyDB.rawQuery("Select * from users where username = ?", new
String[]{username});

if (cursor.getCount() > 0)

return true;

else

return false;

}

public Boolean checkusernamepassword(String username, String password) {

SQLiteDatabase MyDB = this.getWritableDatabase();

Cursor cursor = MyDB.rawQuery("Select * from users where username = ? and password = ?",
new String[]{username, password});

if (cursor.getCount() > 0)

return true;

else

return false;

}

}

```

QuestionItems.java

```
package com.androidapplicationdevelopment.quizbuzzbuildversion;

public class QuestionsItem {
    String questions,answer1,answer2, answer3, answer4, correct;

    public QuestionsItem(String questions, String answer1, String answer2, String answer3, String
    answer4, String correct) {
        this.questions = questions;
        this.answer1 = answer1;
        this.answer2 = answer2;
        this.answer3 = answer3;
        this.answer4 = answer4;
        this.correct = correct;
    }

    public String getQuestions() {
        return questions;
    }

    public void setQuestions(String questions) {
        this.questions = questions;
    }

    public String getAnswer1() {
        return answer1;
    }

    public void setAnswer1(String answer1) {
        this.answer1 = answer1;
    }
}
```

```

    }

    public String getAnswer2() {
        return answer2;
    }

    public void setAnswer2(String answer2) {
        this.answer2 = answer2;
    }

    public String getAnswer3() {
        return answer3;
    }

    public void setAnswer3(String answer3) {
        this.answer3 = answer3;
    }

    public String getAnswer4() {
        return answer4;
    }

    public void setAnswer4(String answer4) {
        this.answer4 = answer4;
    }

    public String getCorrect() {
        return correct;
    }

    public void setCorrect(String correct) {
        this.correct = correct;
    }
}

```


AndroidManifest.xml (Manifestation File)

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/applogo"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/applogo_round"
        android:supportsRtl="true"
        android:theme="@style/Base.Theme.QuizBuzzBuildVersion"
        tools:targetApi="31">

        <activity
            android:name=".ResultActivity"
            android:exported="false" />

        <activity
            android:name=".EasyBuzzer"
            android:exported="true">

            <intent-filter>

                <action
                    android:name="com.androidapplicationdevelopment.quizbuzzbuildversion.EASY_BUZZER"
                />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

</intent-filter>

</activity>

<activity
    android:name=".MediumBuzzer"
    android:exported="true">
    <intent-filter>
        <action
            android:name="com.androidapplicationdevelopment.quizbuzzbuildversion.MEDIUM_BUZZER" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    </activity>

    <activity
        android:name=".ToughBuzzer"
        android:exported="true">
        <intent-filter>
            <action
                android:name="com.androidapplicationdevelopment.quizbuzzbuildversion.TOUGH_BUZZER" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
        </activity>

        <activity
            android:name=".LoginActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

```

```

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

    android:name=".RegistrationActivity"

    android:exported="true" />

<activity

    android:name=".SplashActivity"

    android:exported="true" />

<activity

    android:name=".MainActivity"

    android:exported="true" />

</application>

</manifest>

```

5.1 Handling Errors and Test Cases

Error Handling:

1. Identify Potential Errors: Begin by identifying potential errors that could occur during the execution of your Quiz Buzz app. This might include network errors, data validation issues, unexpected user input, or system failures.
2. Implement Exception Handling: Use try-catch blocks to handle exceptions gracefully. For example, if there's an error fetching quiz data from a server, catch the exception and display an appropriate error message to the user.
3. Provide User Feedback: When an error occurs, communicate clearly with the user by displaying meaningful error messages. This helps users understand what went wrong and how they can resolve the issue.
4. Log Errors: Logging errors to a file or console can be helpful for debugging and

troubleshooting issues. Use logging frameworks like Log4j or SLF4J to log error messages along with relevant context information.

Test Cases :

1. **Unit Tests:** Write unit tests to validate the functionality of individual components of your Quiz Buzz app, such as quiz generation algorithms, scoring logic, and data validation methods. Use testing frameworks like JUnit or TestNG to automate the testing process.
2. **Integration Tests:** Test the interaction between different modules or layers of your application, such as the user interface, backend server, and database. Integration tests ensure that all components work together as expected.
3. **UI Tests:** Use automated UI testing frameworks like Selenium or Appium to test the user interface of your Quiz Buzz app. Verify that buttons, text fields, and other UI elements behave correctly and respond to user interactions as intended.
4. **Edge Cases:** Consider edge cases and boundary conditions when designing test cases. Test scenarios such as empty quiz data, invalid user inputs, and extreme values to ensure that your app handles these situations gracefully.
5. **Regression Tests:** As you make changes or add new features to your Quiz Buzz app, run regression tests to ensure that existing functionality remains unaffected. This helps prevent unintended side effects and regressions.
6. **Load Testing:** Perform load testing to assess the performance and scalability of your Quiz Buzz app under heavy usage. Identify bottlenecks and optimize your app to handle a large number.

Chapter – 6

CONCLUSION

The Quiz Buzz app represents a dynamic and engaging platform that seamlessly blends education with entertainment. Through its innovative features, intuitive interface, and diverse content, Quiz Buzz offers users a captivating experience that caters to a wide range of interests and learning styles.

With customizable quizzes, multiplayer competitions, and interactive learning resources, Quiz Buzz empowers users to challenge themselves, compete with friends, and explore new topics in a fun and engaging way. The app's emphasis on user interaction, social connectivity, and continuous improvement fosters a sense of community and camaraderie among its users. Furthermore, Quiz Buzz stands out for its commitment to quality, reliability, and user satisfaction. Thorough error handling, comprehensive test coverage, and ongoing updates ensure that the app delivers a seamless and enjoyable experience for users across various platforms and devices.

In a world where learning is a lifelong journey and curiosity knows no bounds, Quiz Buzz serves as a beacon of knowledge and discovery, inviting users to embark on an exciting adventure of trivia, learning, and fun. Whether you're a trivia enthusiast, a student, or simply someone who loves a good challenge, Quiz Buzz has something for everyone, making learning accessible, interactive, and endlessly rewarding.

Chapter – 7

FUTURE SCOPE

The future scope of Quiz Buzz holds promising opportunities for expansion and enhancement, aimed at further enriching the user experience and broadening its reach. One avenue for growth lies in the continuous expansion of content, with the addition of new quizzes, topics, and categories to cater to diverse interests and learning preferences. Personalization features represent another area ripe for development, offering users tailored quizzes aligned with their interests, proficiency levels, and learning goals through advanced algorithms and analytics. Introducing gamification elements such as challenges, leaderboards, and rewards can enhance user engagement and motivation, while fostering a sense of community through interactive forums and social media integration can deepen user interaction and collaboration. Moreover, strategic partnerships with educational institutions and organizations can position Quiz Buzz as a valuable educational tool, integrating curriculum-aligned content and reaching a broader audience of students and educators. Embracing accessibility features and expanding multiplatform support can further ensure inclusivity and accessibility for users of all backgrounds and abilities. By focusing on these avenues of growth and innovation, Quiz Buzz can continue to evolve as a leading quiz application, inspiring learning, curiosity, and engagement among its users well into the future.

REFERENCES

1. Academic Journals and Articles:

- Explore academic databases such as IEEE Xplore, ACM Digital Library, or Google Scholar for research papers on mobile app development, gamification, user engagement, and educational technology.

2. Books:

- "**Game Design Workshop: A Play centric Approach to Creating Innovative Games**" by Tracy Fullerton.
- "**Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps**" by Theresa Neil.

3. Online Resources:

- Blogs and websites related to app development, user experience design, and gamification.
- Documentation and tutorials from mobile app development platforms like Android Developers and iOS Dev Center.

4. Market Research:

- Analyze existing quiz applications available on app stores (e.g., Google Play Store, Apple App Store) to understand their features, user ratings, and reviews.

5. Interviews and Surveys:

- Conduct interviews with potential users or domain experts to gather insights and feedback on their preferences, needs, and pain points related to quiz applications.