**3506 PROJECT**

**ONLINE QUIZ APPLICATION**

Team members

Anusri Saini 249502500

Sujith Vardhan 249502510

Rishi Koushik Gundenti 249502450

Bindu Maryada 249459110

**PHASE 1**

*TASK 2: CREATE PRODUCT VISION*

USING PRO PAD

FOR: EDUCATORS, TRAINERS, STUDENTS

THE: ONLINE QUIZ APPLICATION IS A WEB-BASED ASSESSMENT PLATFORM

THAT: ENABLES THE CREATION, DELIVERY AND ANALYSIS OF INTERACTIVE QUIZZES TO IMPROVE LEARNING AND EVALUATION OUTCOMES.

UNLIKE: TRADITIONAL PAPER -BASED TESTS OR BASIC QUIZ TOOLS

OUR PRODUCT: OFFERS REAL-TIME FEEDBACK, ADAPTIVE DIFFICULTY, DETAILED PERFORMANCE ANALYTICS AND A SECURE, USER FRIENDLY EXPERIENCE ACROSS DEVICES.

*TASK 3: DEVELOPMENT MODEL*

THE PROJECT FOLLOWS THE AGILE DEVELOPMENT MODEL WITH AN ITERATIV AND INCREMENTAL APPROACH. FEATURES LIKE LOGIN, REGISTRATION, QUIZ, FEEDBACK ARE DEVELOPED IN SEPARATE MODULES. EACH MODULE IS TESTED AND REFINED INDEPENDENTLY, ALLOWING CONTINUOUS IMPROVEMENT. THE USE OF FIREBASE AUTHENTICATION AND REAL-TIME FEEDBACK SUPPORTS AGILE PRACTICES. CODE IS ORGANISED IN A WAY THAT FITS SCRUM ROLES LIKE PRODUCT OWNER AND DEVELOPER. DEVELOPMENT CAN BE STURCTURED INTO SHORT SPRINTS, EACH DELIVERING WORKING FEATURES. THE PROJECT SUPPORTS CHANGES AND ENHANCEMENTS EASILY – A KEY AGILE ADVANTAGE. AGILE ENABLES QUICK INCORPORATION OF USER OR INSTURCTOR FEEDBACK DURING DEVELOPMENT. UNLIKE WATERFALL, TESTING AND CODING HAPPEN IN PARALLED, NOT SEQUENTIALLY. THEREFORE, AGILE WITH SCRUM IS THE MOST LOGICAL AND EFFECTIVE MODEL FOR PROJECT.

**PHASE 2:**

*TASK 1: SOFTWARE ARCHITECTURE*

1. Domain Model

The domain model represents the main entities in the AMMU Quiz App and their relationships.

Entities:- User: Represents both students and admins with attributes like user\_id, name, email, and

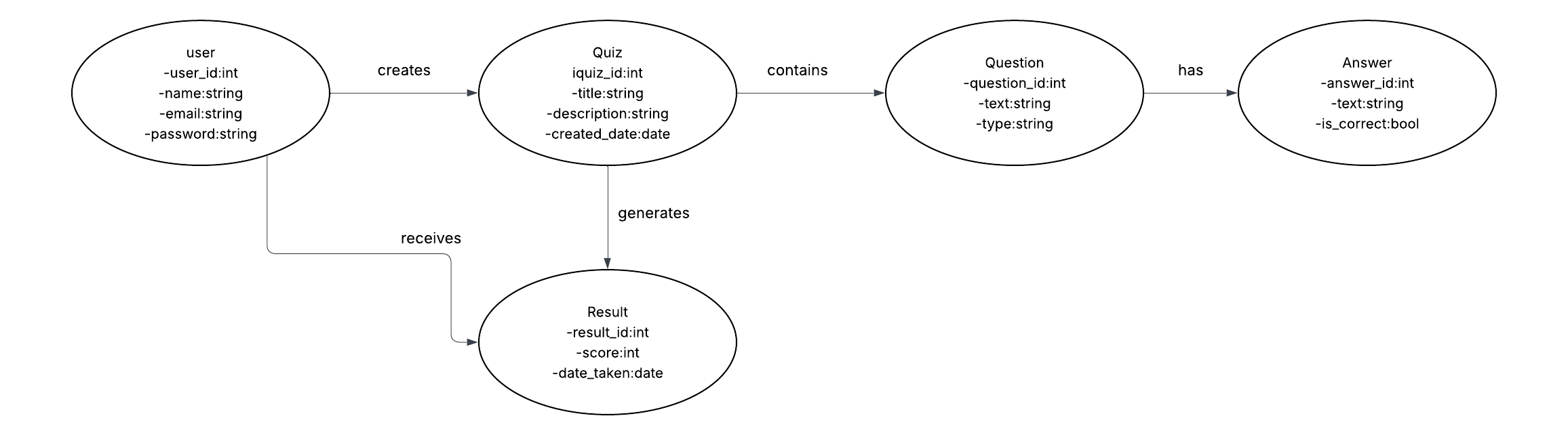
password.- Quiz: Represents quizzes created in the system with attributes like quiz\_id, title, description, and

created\_date.- Question: Represents individual quiz questions with attributes like question\_id, text, and type.- Answer: Represents possible answers linked to questions with attributes like answer\_id, text, and

is\_correct.- Result: Represents user scores with attributes like result\_id, score, and date\_taken.

Relationships:-

* A User creates Quizzes
* A Quiz contains Questions
* A Question has multiple Answers
* A Quiz generates a Result for a User.



2. Sequence Diagram: User Registration

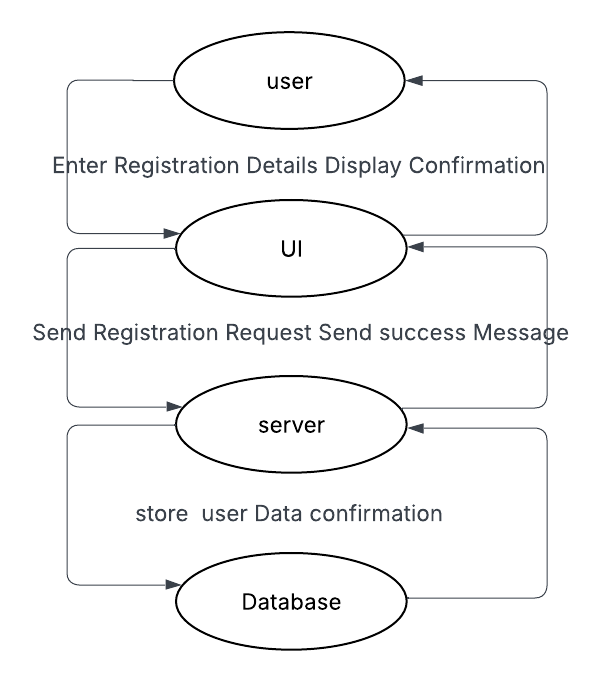
This sequence diagram outlines the workflow of user registration:-

-The User enters registration details (name, email, password)

-The UI sends the data to the Server.

- The Server stores the user data in the Database.-

-A success message is sent back to the User



3. Sequence Diagram: Quiz Creation

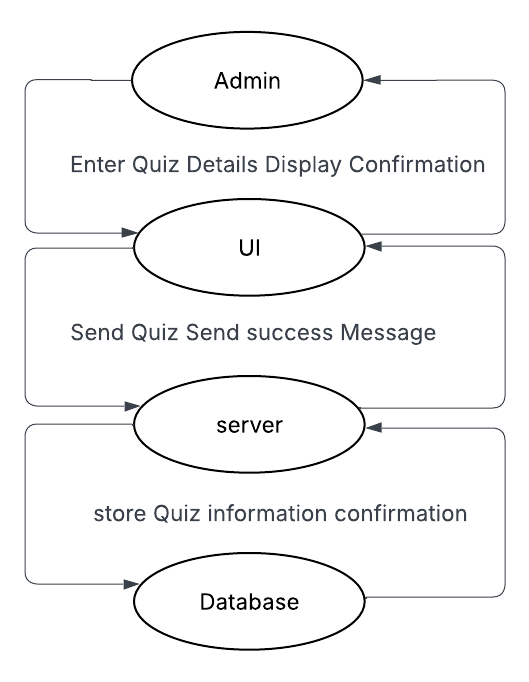
This diagram illustrates the process of quiz creation:-

-The Admin enters quiz details in the UI.

- The UI sends the details to the Server.

- The Server saves the quiz information in the Database.

- The Admin receives a confirmation message.



4. Sequence Diagram: Quiz Participation

This diagram describes the workflow of quiz participation:

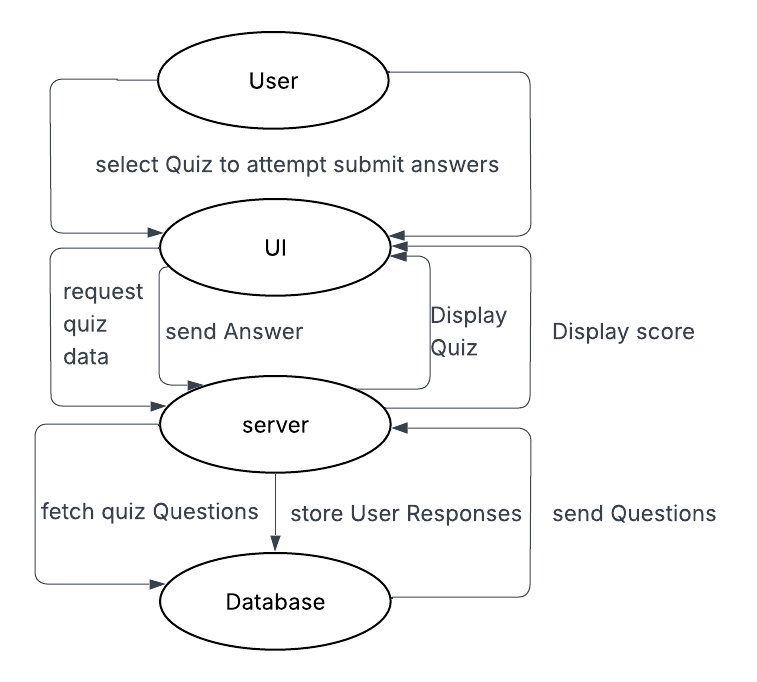
-The User selects a quiz to attempt.

- The UI requests quiz data from the Server.

- The Server fetches questions from the Database and sends them to the UI.

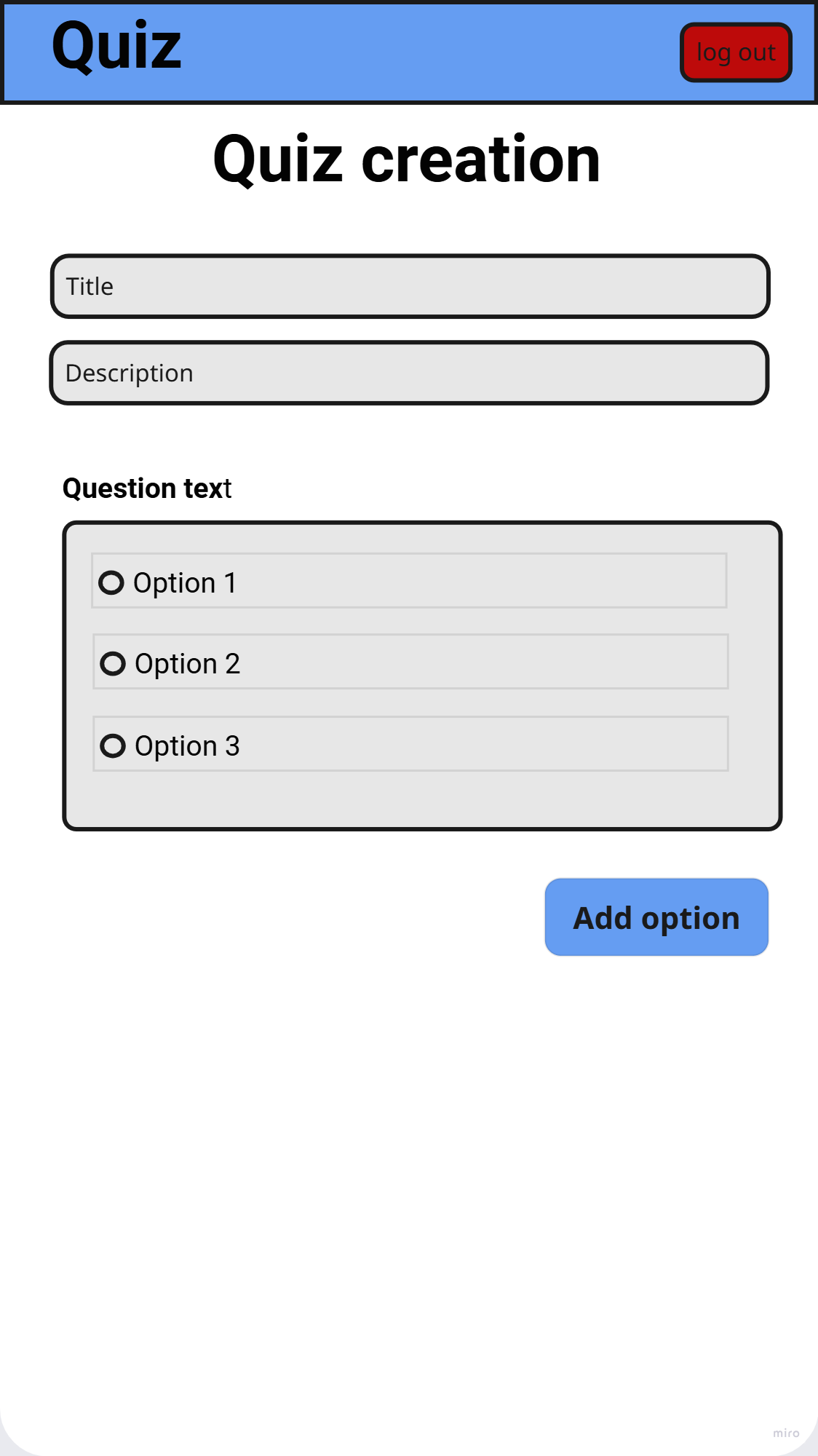
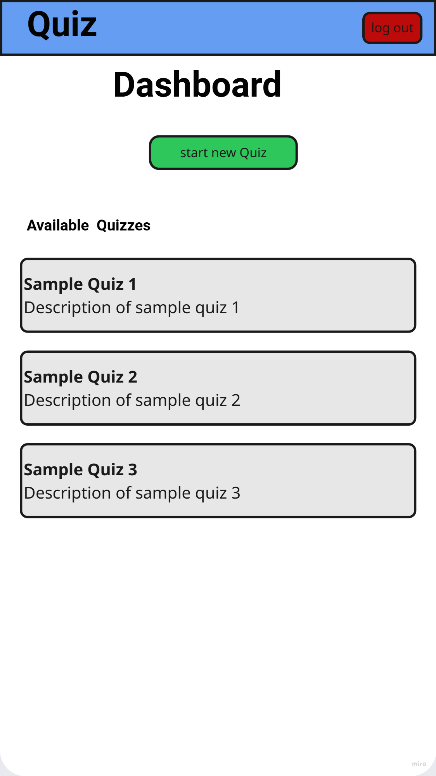
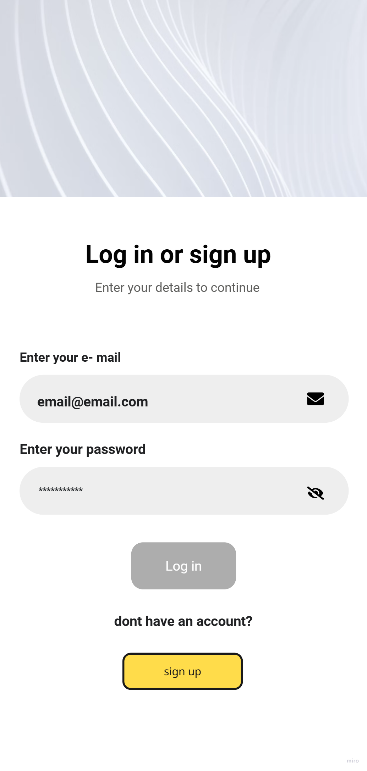
- The User submits answers.

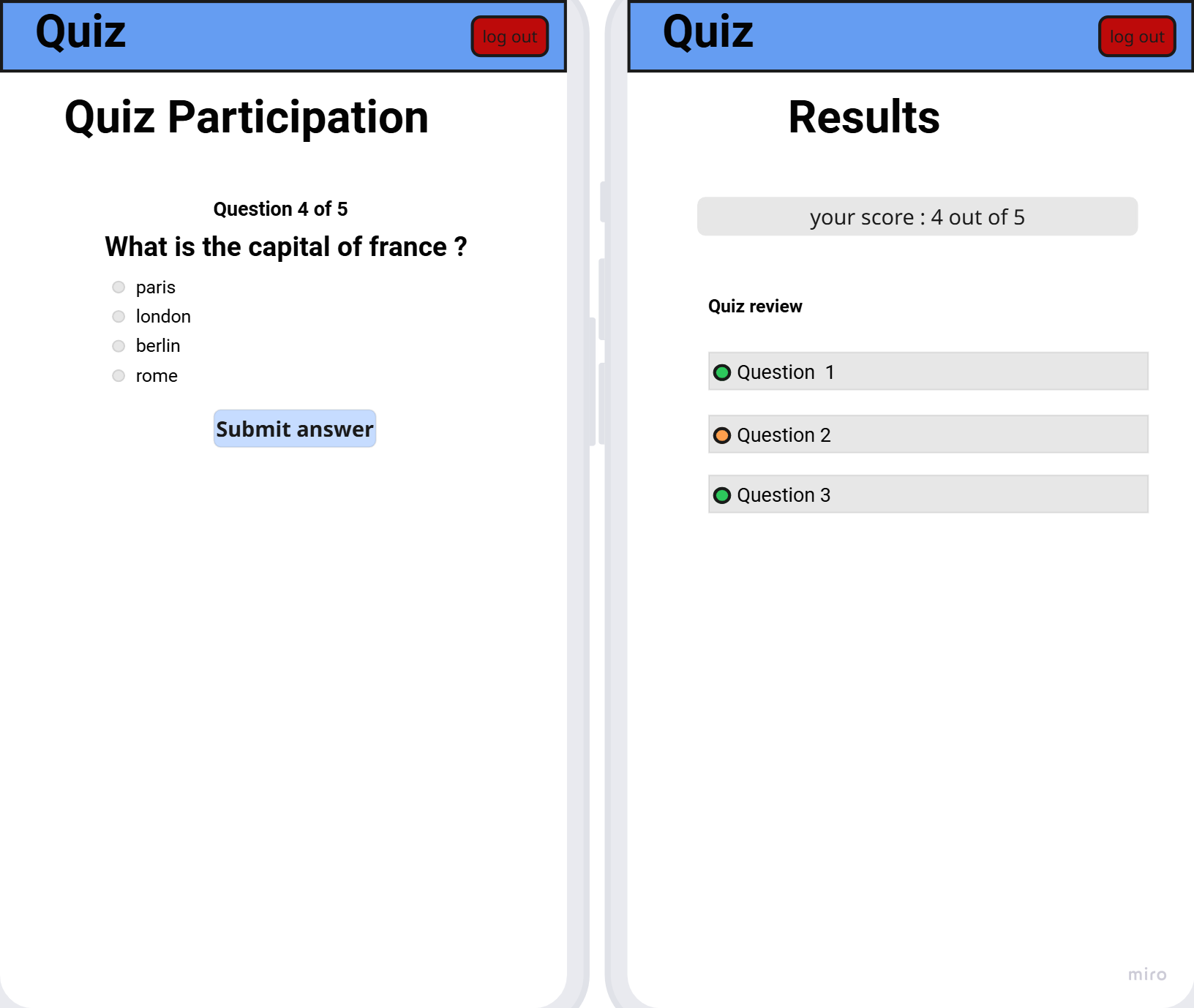
- The Server stores responses, calculates the score, and sends it to the UI for display.



**PHASE 3**

*TASK 1: WIREFRAMING*





*TASK 2: DESIGN DOCUMENTATION*

The design document for the online quiz application details functional and non-functional requirements and provides colourful wireframes for all key user interfaces. It aims to serve as a comprehensive blueprint for developers and stakeholders.

*FUNCTIONAL REQUIREMENTS*

FR-01: User Authentication

Users can register and log in securely using their email and password.

FR-02: Quiz Creation (Admin)

Admins can create quizzes with multiple questions and options.

FR-03: Quiz Participation (User)

Users can select and attempt quizzes from a dashboard.

FR-04: Automated Grading

The system grades submitted quizzes and calculates scores.

FR-05: Results Display

Users can view their results and review answers after submission.

*NON-FUNCTIONAL REQUIREMENTS*

NFR-01: Performance

The system should support up to 100 concurrent users without lag.

NFR-02: Security

All passwords are encrypted. Admin and user roles are properly separated.

NFR-03: Usability

The UI is intuitive and user-friendly for students and educators.

NFR-04: Scalability

New quiz types and question formats can be added easily.

NFR-05: Availability

The system will be available 24/7 with 99.9% uptime.

**PHASE 4**

*TASK 2: SOFTWARE DEVELOPMENT*

**FRONT END CODE**

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

<LinearLayout 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:background="@drawable/registerback"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".register">

<TextView

android:id="@+id/loginscrn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="80dp"

android:text="Registration"

android:textSize="25dp"

android:textStyle="bold"

android:layout\_gravity="center"/>

<TextView

android:id="@+id/fstTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:layout\_marginTop="20dp"

android:text="Full Name"/>

<EditText

android:id="@+id/txtName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:ems="10"/>

<TextView

android:id="@+id/secTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Email"

android:layout\_marginLeft="100dp" />

<EditText

android:id="@+id/txtEmail"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:ems="10" />

<TextView

android:id="@+id/thirdTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Password"

android:layout\_marginLeft="100dp" />

<EditText

android:id="@+id/txtPwd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:inputType="textPassword"

android:ems="10" />

<ProgressBar

android:id="@+id/progessBar"

android:visibility="gone"

android:layout\_gravity="center\_horizontal"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<Button

android:id="@+id/btnRegister"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:layout\_gravity="center\_horizontal"

android:text="Register" />

<TextView android:id="@+id/lnkLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:text="Already Registered? Login here"

android:gravity="center"

android:textSize="20dp"

android:textColor="#3F51B5"

android:onClick="test"/>

</LinearLayout>

**LOGIN**

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

<LinearLayout 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:background="@drawable/back"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".login">

<TextView

android:id="@+id/loginscrn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="80dp"

android:layout\_marginBottom="30dp"

android:text="Login"

android:textSize="35dp"

android:textStyle="bold"

android:layout\_gravity="center"/>

<TextView

android:id="@+id/fstTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:layout\_marginTop="20dp"

android:text="Email"/>

<EditText

android:id="@+id/txtEmail"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:ems="10"/>

<TextView

android:id="@+id/secTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Password"

android:layout\_marginLeft="100dp" />

<EditText

android:id="@+id/txtPwd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="100dp"

android:inputType="textPassword"

android:ems="10" />

<ProgressBar

android:id="@+id/progressBar"

android:visibility="gone"

android:layout\_gravity="center\_horizontal"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<Button

android:id="@+id/btnLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="20dp"

android:text="Login" />

<TextView android:id="@+id/lnkRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:text="don't have account? Register here"

android:gravity="center"

android:textSize="20dp"

android:textColor="#3F51B5"/>

</LinearLayout>

**BACKEND**

**Login page**

package easy.tuto.myquizapplication;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.Window;

import android.view.WindowManager;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class login extends AppCompatActivity {

EditText emailEdit,passwordEdit;

Button login;

TextView register;

FirebaseAuth mAuth;

ProgressBar progressBar;

@Override

public void onStart() {

super.onStart();

// Check if user is signed in (non-null) and update UI accordingly.

FirebaseUser currentUser = mAuth.getCurrentUser();

if(currentUser != null){

Intent intent = new Intent(login.this, Categories.class);

startActivity(intent);

finish();

}

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEATURE\_NO\_TITLE); //will hide the title

getSupportActionBar().hide(); // hide the title bar

this.getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

WindowManager.LayoutParams.FLAG\_FULLSCREEN); //enable full screen

setContentView(R.layout.activity\_login);

mAuth = FirebaseAuth.getInstance();

emailEdit = findViewById(R.id.txtEmail);

passwordEdit = findViewById(R.id.txtPwd);

progressBar = findViewById(R.id.progressBar);

login = findViewById(R.id.btnLogin);

register = findViewById(R.id.lnkRegister);

login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

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

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

if( email.length() == 0 && password.length() == 0 ){

Toast.makeText(login.this,

"please

password.",Toast.LENGTH\_SHORT).show();

}

else {

progressBar.setVisibility(view.VISIBLE);

enter

mAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

progressBar.setVisibility(view.GONE);

if (task.isSuccessful()) {

// Sign in success, update UI with the signed-in user's information

email

and

Toast.makeText(login.this, "Authentication sucessful.", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(login.this, Categories.class);

startActivity(intent);

finish();

} else {

// If sign in fails, display a message to the user.

Toast.makeText(login.this, "Authentication failed.", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

});

register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(login.this, register.class);

startActivity(intent);

}

});

}

}

**Mainactivity.javafile**

package easy.tuto.myquizapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;

import android.graphics.Color;

import android.os.Bundle;

import android.os.CountDownTimer;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import java.text.DecimalFormat;

import java.text.NumberFormat;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

TextView totalQuestionsTextView,countdown;

TextView questionTextView;

Button ansA, ansB, ansC, ansD;

Button submitBtn;

int score=0;

int totalQuestion = QuestionAnswer.question.length;

int currentQuestionIndex = 0;

String selectedAnswer = "";

int time=60000\*totalQuestion; // time's maximum value

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

countdown = findViewById(R.id.countdown);

//

count down timer

new CountDownTimer(time, 1000) {

public void onTick(long millisUntilFinished) {

// Used for formatting digit to be in 2 digits only

NumberFormat f = new DecimalFormat("00");

long min = (millisUntilFinished / 60000) % 60;

long sec = (millisUntilFinished / 1000) % 60;

countdown.setText(f.format(min) + ":" + f.format(sec));

}

// When the task is over it will print 00:00:00 there

public void onFinish() {

countdown.setText("00:00");

}

}.start();

totalQuestionsTextView = findViewById(R.id.total\_question);

questionTextView = findViewById(R.id.question);

ansA = findViewById(R.id.ans\_A);

ansB = findViewById(R.id.ans\_B);

ansC = findViewById(R.id.ans\_C);

ansD = findViewById(R.id.ans\_D);

submitBtn = findViewById(R.id.submit\_btn);

ansA.setOnClickListener(this);

ansB.setOnClickListener(this);

ansC.setOnClickListener(this);

ansD.setOnClickListener(this);

submitBtn.setOnClickListener(this);

totalQuestionsTextView.setText("Total questions : "+totalQuestion);

loadNewQuestion();

}

@Override

public void onClick(View view) {

ansA.setBackgroundColor(Color.WHITE);

ansB.setBackgroundColor(Color.WHITE);

ansC.setBackgroundColor(Color.WHITE);

ansD.setBackgroundColor(Color.WHITE);

Button clickedButton = (Button) view;

if(clickedButton.getId()==R.id.submit\_btn){

if(selectedAnswer.equals(QuestionAnswer.correctAnswers[currentQuestionIndex])){

score++;

}

currentQuestionIndex++;

loadNewQuestion();

}else{

//choices button clicked

selectedAnswer = clickedButton.getText().toString();

clickedButton.setBackgroundColor(Color.MAGENTA);

}

}

void loadNewQuestion(){

if(currentQuestionIndex == totalQuestion ){

finishQuiz();

return;

}

questionTextView.setText(QuestionAnswer.question[currentQuestionIndex]);

ansA.setText(QuestionAnswer.choices[currentQuestionIndex][0]);

ansB.setText(QuestionAnswer.choices[currentQuestionIndex][1]);

ansC.setText(QuestionAnswer.choices[currentQuestionIndex][2]);

ansD.setText(QuestionAnswer.choices[currentQuestionIndex][3]);

}

void finishQuiz(){

String passStatus = "";

if(score > totalQuestion\*0.60){

passStatus = "Passed";

}else{

passStatus = "Failed";

}

new AlertDialog.Builder(this)

.setTitle(passStatus)

.setMessage("Score is "+ score+" out of "+ totalQuestion)

.setPositiveButton("Restart",(dialogInterface, i) -> restartQuiz() )

.setCancelable(false)

.show();

}

void restartQuiz(){

score = 0;

currentQuestionIndex =0;

loadNewQuestion();

}

}

**Register.java**

package easy.tuto.myquizapplication;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.view.View;

import android.view.Window;

import android.view.WindowManager;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class register extends AppCompatActivity {

EditText name,emailEdit,passwordEdit;

FirebaseAuth mAuth;

ProgressBar progressBar;

@Override

public void onStart() {

super.onStart();

// Check if user is signed in (non-null) and update UI accordingly.

FirebaseUser currentUser = mAuth.getCurrentUser();

if(currentUser != null){

Intent intent = new Intent(register.this, Categories.class);

startActivity(intent);

finish();

}

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

getSupportActionBar().hide();

// hide

//will the hide title the

this.getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

WindowManager.LayoutParams.FLAG\_FULLSCREEN); //enable full screen

setContentView(R.layout.activity\_register);

mAuth = FirebaseAuth.getInstance();

name = findViewById(R.id.txtName);

emailEdit = findViewById(R.id.txtEmail);

passwordEdit = findViewById(R.id.txtPwd);

progressBar = findViewById(R.id.progessBar);

Button register = findViewById(R.id.btnRegister);

TextView login = findViewById(R.id.lnkLogin);

register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

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

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

if( email.length() == 0) {

Toast.makeText(register.this,"fill the name", Toast.LENGTH\_SHORT).show();

return;

title

bar

}

if (password.length()==0) {

Toast.makeText(register.this, "fill all the email", Toast.LENGTH\_SHORT).show();

return;

}

progressBar.setVisibility(view.VISIBLE);

mAuth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

progressBar.setVisibility(view.GONE);

if (task.isSuccessful()) {

// Sign in success, update UI with the signed-in user's information

Toast.makeText(register.this, "Authentication sucessful.", Toast.LENGTH\_SHORT).show();

} else {

// If sign in fails, display a message to the user.

Toast.makeText(register.this, "Authentication failed.", Toast.LENGTH\_SHORT).show();

}

}

});

}

});

login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(register.this, login.class);

startActivity(intent);

}

});

}

}

**Catagerioes.java**

package easy.tuto.myquizapplication;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;

public class Categories extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_categories);

Button verbal = findViewById(R.id.verbal);

Button numerical = findViewById(R.id.numberical);

Button reasoning = findViewById(R.id.reasoning);

verbal.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(Categories.this,MainActivity.class);

startActivity(intent);

}

});

numerical.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(Categories.this,Numerical.class);

startActivity(intent);

}

});

reasoning.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(Categories.this,Reasoning.class);

startActivity(intent);

}

});

}

@Override

public void onBackPressed(){

new AlertDialog.Builder(this)

.setTitle("Really Exit")

.setMessage("Are you sure you want to exit?")

.setNegativeButton(android.R.string.no, null)

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

public void onClick(DialogInterface arg0, int arg1) {

Categories.super.onBackPressed();

}

}).create().show();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.menu,menu);

return true;

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()){

case R.id.feedback:

Intent i = new Intent(Categories.this,feedback.class);

startActivity(i);

return true;

case R.id.logout:

FirebaseAuth.getInstance().signOut();

Intent it = new Intent(Categories.this,login.class);

startActivity(it);

return true;

}

return super.onOptionsItemSelected(item);

}

}

**Feedback.java**

package easy.tuto.myquizapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class feedback extends AppCompatActivity {

EditText name,email,problem;

Button submit;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_feedback);

name = findViewById(R.id.name);

email = findViewById(R.id.email);

problem = findViewById(R.id.problem);

submit = findViewById(R.id.submit);

submit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(name.getText().toString().isEmpty()

problem.getText().toString().isEmpty()){

||

email.getText().toString().isEmpty()

||

Toast.makeText(feedback.this,"please fill all the details",Toast.LENGTH\_SHORT).show();

}else {

Intent i = new Intent(Intent.ACTION\_SEND);

i.setType("message/html");

i.putExtra(Intent.EXTRA\_EMAIL, new String[]{"ushasri.boyina@gmail.com"});

i.putExtra(Intent.EXTRA\_SUBJECT, "Feedback From Quiz App");

i.putExtra(Intent.EXTRA\_TEXT, "Name: " + name.getText() + "\n Message:" +

problem.getText());

try {

startActivity(Intent.createChooser(i, "please select Email"));

} catch (android.content.ActivityNotFoundException ex) {

Toast.makeText(feedback.this,

"There

Toast.LENGTH\_SHORT).show();

}

}

}

});

}

}

**QUESTIONANSWER.JAVA**

package easy.tuto.myquizapplication;

public class QuestionAnswer {

public static String question[] ={

are

or

Email

clients",

"Choose one of the following options that means the opposite of the given word;

Copious:",

"Which one is not the programming language?",

"Where you are watching this video?"

};

public static String choices[][] = {

{"Reverse","Scarce","Abundant","Short"},

{"Java","Kotlin","Notepad","Python"},

{"Facebook","Whatsapp","Instagram","Youtube"}

};

public static String correctAnswers[] = {

"Scarce",

"Notepad",

"Youtube"

};

***PHASE 5***

*TASK 1: PROJECT TESTING*

*1.FUNCTIONAL TESTING*

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Output** | **Actual Output** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC\_F01 | User Registration | Open app → Go to Register → Enter valid Name, Email, Password → Tap Register | User is registered and redirected to Category screen | ✅ As Expected | Pass |
| TC\_F02 | Registration with blank fields | Leave fields blank → Tap Register | Error messages shown, no registration | ✅ As Expected | Pass |
| TC\_F03 | User Login (Valid) | Enter valid registered email/password → Tap Login | Redirect to Category screen | ✅ As Expected | Pass |
| TC\_F04 | User Login (Invalid) | Enter invalid credentials → Tap Login | Show “Authentication failed” | ✅ As Expected | Pass |
| TC\_F05 | Category Navigation | After login → Tap on Verbal/Numerical/Reasoning buttons | Navigate to respective quiz activity | ✅ As Expected | Pass |
| TC\_F06 | Quiz Attempt & Submit | Load quiz → Select options → Tap Submit | Answer is recorded, next question shown | ✅ As Expected | Pass |
| TC\_F07 | Complete Quiz | Finish last question → Tap Submit | Show Score + Pass/Fail message | ✅ As Expected | Pass |
| TC\_F08 | Restart Quiz | On result alert → Tap “Restart” | Quiz restarts from Q1 | ✅ As Expected | Pass |
| TC\_F09 | Feedback Submission | Fill name, email, message → Tap Submit | Email app opens with prefilled content | ✅ As Expected | Pass |
| TC\_F10 | Back Navigation Prompt | Tap back on Category screen | Confirmation dialog shown | ✅ As Expected | Pass |
| TC\_F11 | Logout from Menu | On Category screen → Tap Logout in menu | Return to Login screen | ✅ As Expected | Pass |

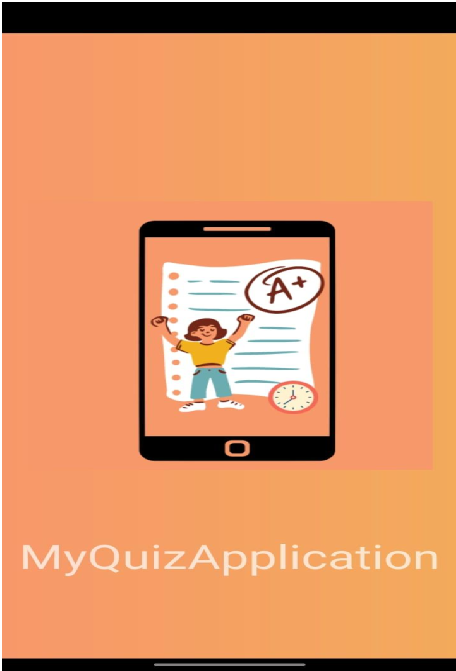
*2. NON-FUNCTIONAL TESTING*

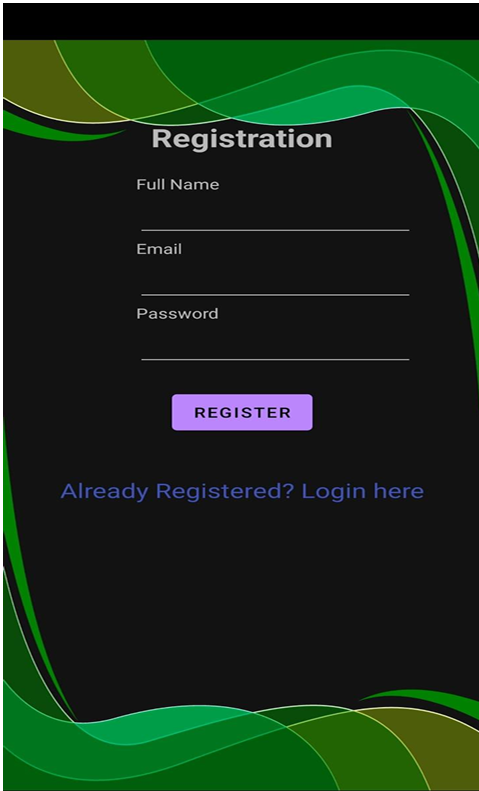
| **Test Type** | **Scenario** | **Expected Behavior** | **Result** |
| --- | --- | --- | --- |
| Performance Testing | App should respond to input and transition between screens within 2 seconds | Transitions are smooth and under 1.5 seconds | ✅ Pass |
| Usability Testing | UI should be readable, accessible, and logically structured | Text size is readable, navigation is clear | ✅ Pass |
| Compatibility | Run app on different Android versions (e.g., 8, 10, 11) | App launches and functions properly | ✅ Pass |
| Security | Attempt to access quiz screen without login | Access denied, redirected to login | ✅ Pass |
| Error Handling | Leave fields empty during registration/login | Relevant error messages shown | ✅ Pass |
| Input Validation | Input invalid email format during registration | Show appropriate input error | ✅ Pass |
| Stability | Use app for long session (20+ min) | App doesn’t crash or freeze | ✅ Pass |
| Load Handling | Multiple attempts of quiz restart and navigation | App maintains state and memory efficiently | ✅ Pass |

*TASK 2: TEST REPORT*

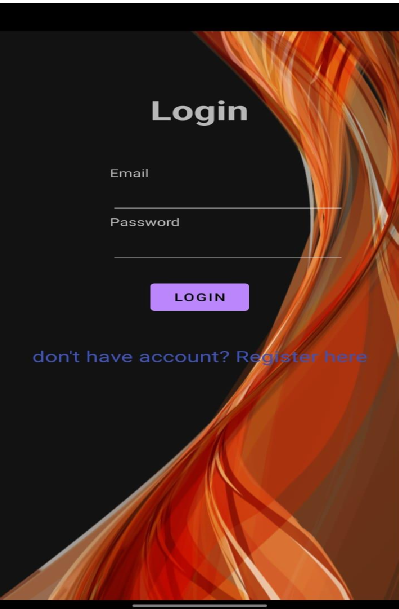
User Registration

Registration screen will appear after Home page when you click on application, after registration user will be able to choose & take quiz which he / she want.

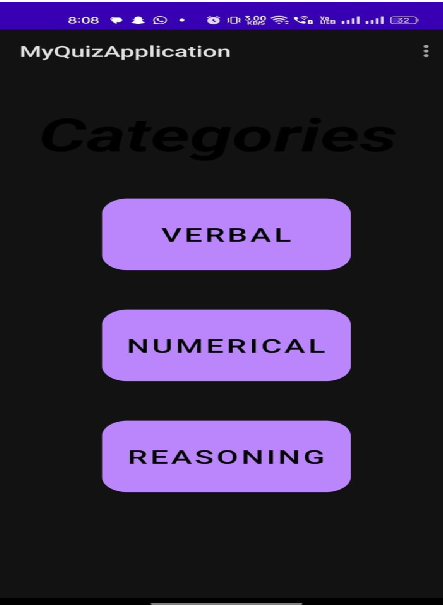




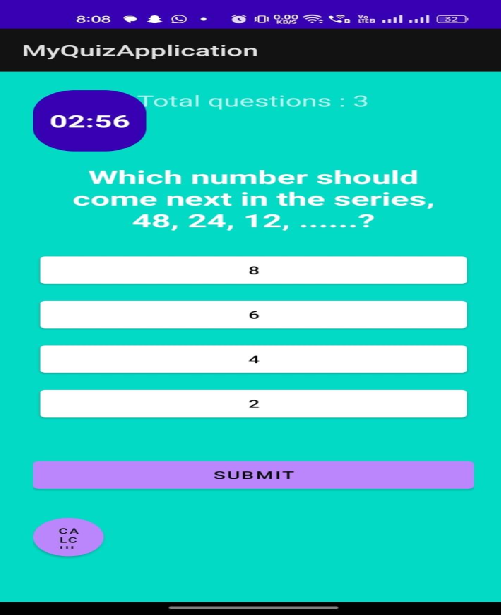
Login

Login screen will appear after successful user registration. Enter username and password to the desire field. Hint text indicates where you want to input username and password .Tap the login button after type username and password.

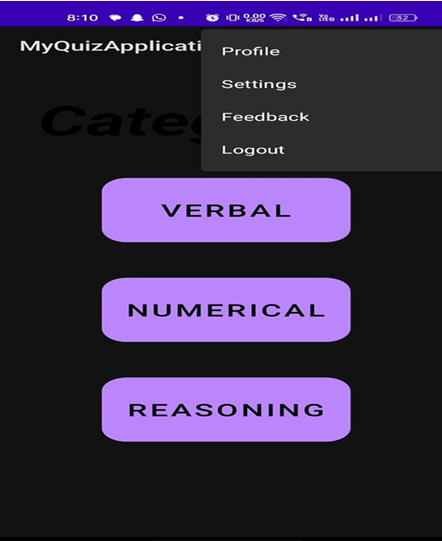
Categories: these are divided into the verbal , numerical , reasoning.



My quiz applications questions: here there are the question and for that questions there is a timer if that time completed the question will be changed the is a calculator for the questions.



In this page, we can go throw profile, settings, feedback and logout option. Profile displays user information, settings help user for adjustments and feedback help user to check his/her ability. Finally logout option help to remove user account. This page mainly focus on giving information and displaying user details.



If any user faces any problem using this application, he can say his problem using this page. This is nothing but a feedback form. For submitting this, user have to give his/her name, email id and his/her experience.

