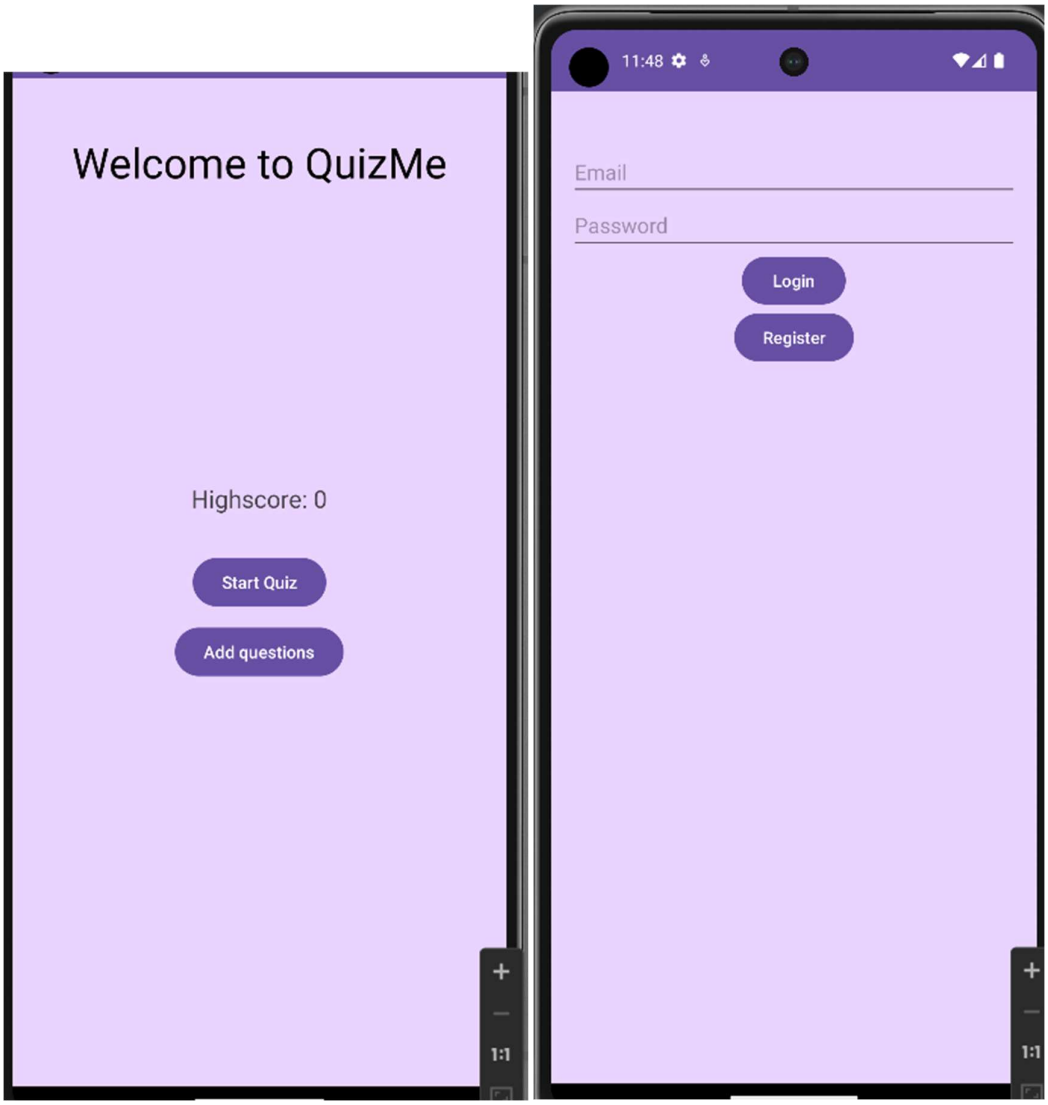
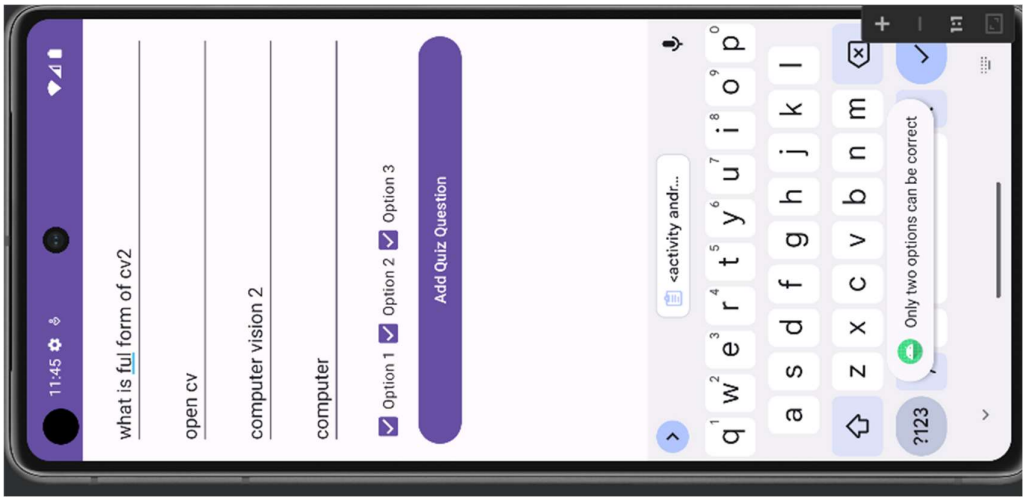
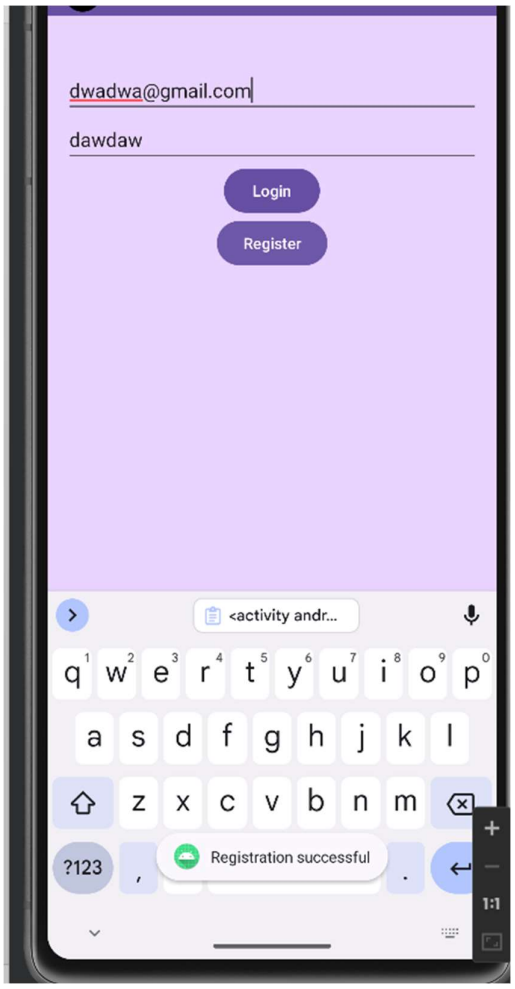
MAD Assignment 2

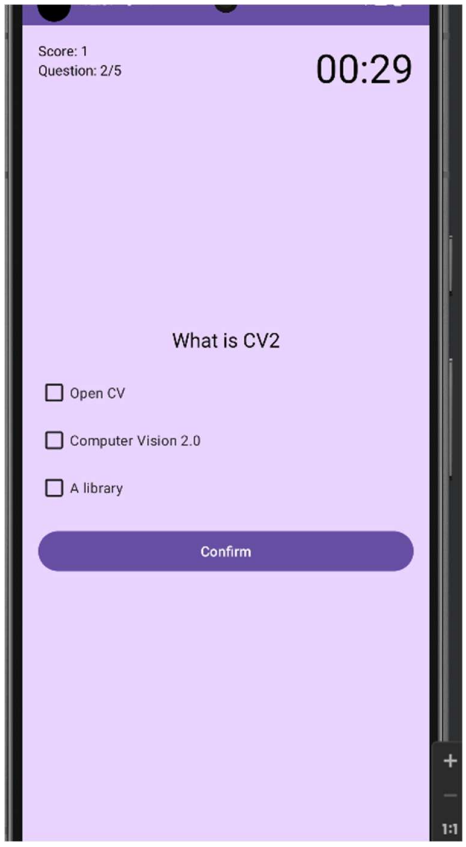
Bhavya Chopra A001 70022100100

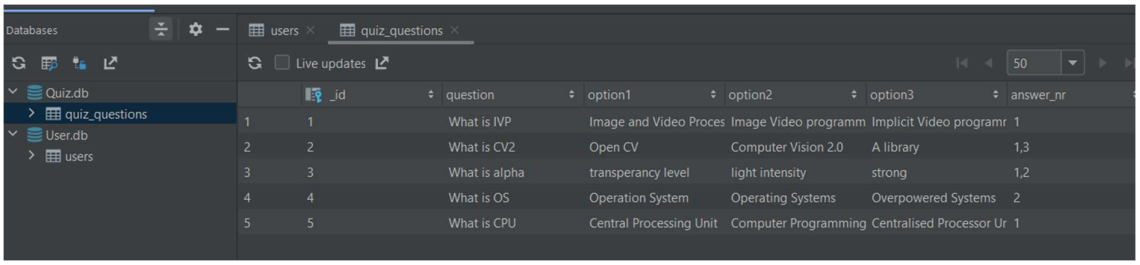
Outputs –

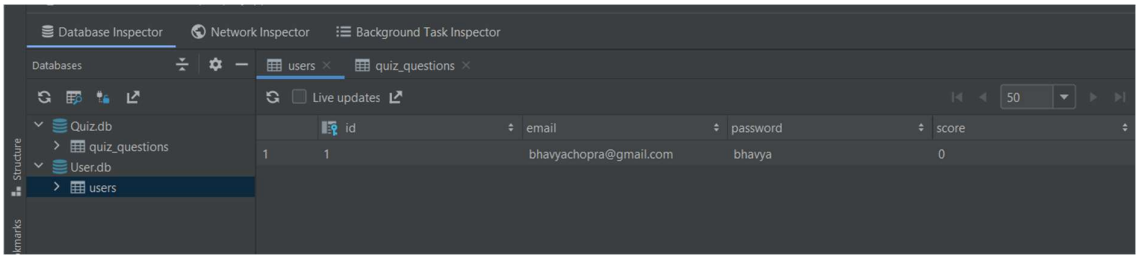



11:45
what is ful form of cv2
open cv
computer vision 2
computer
Option 1
Option 2
Option 3
Add Quiz Question
Only two options can be correct
?123

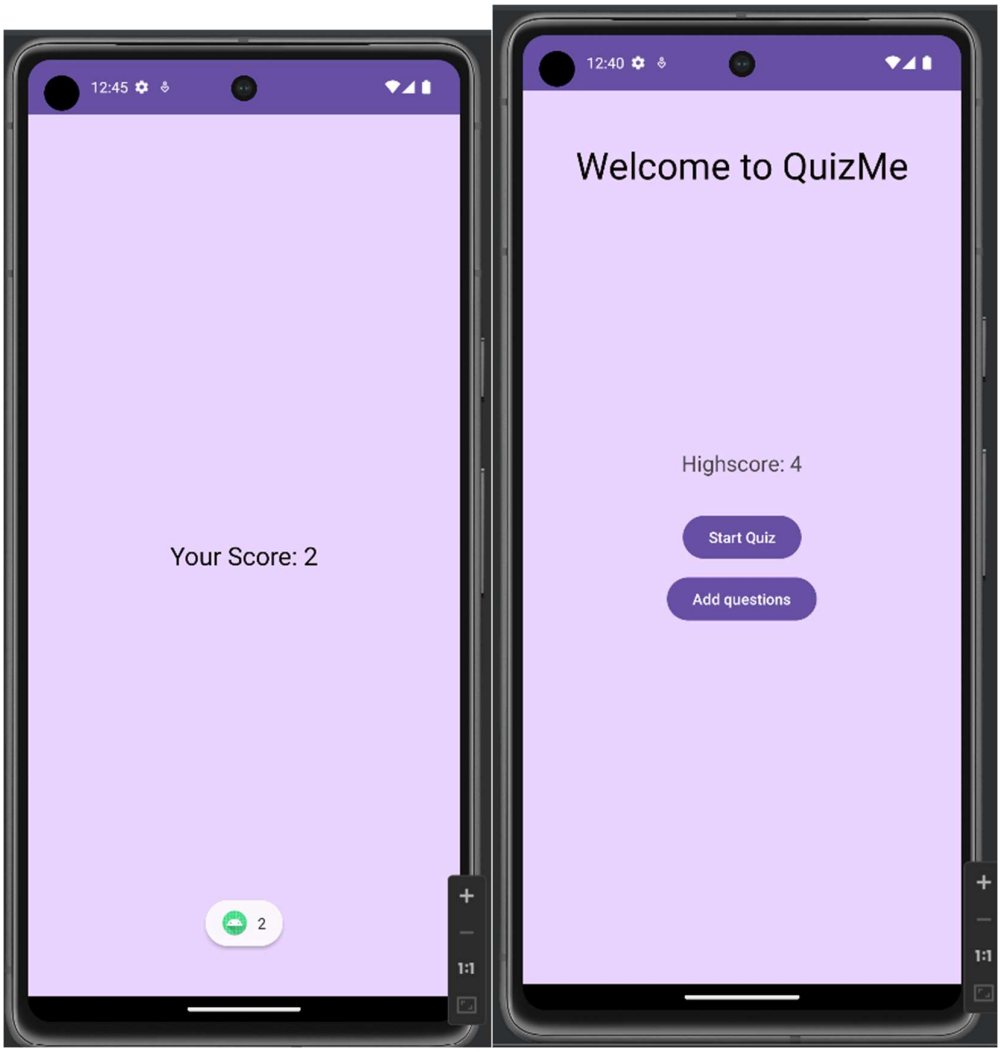

dwadwa@gmail.com
dawdaw
Login
Register
Registration successful
?123

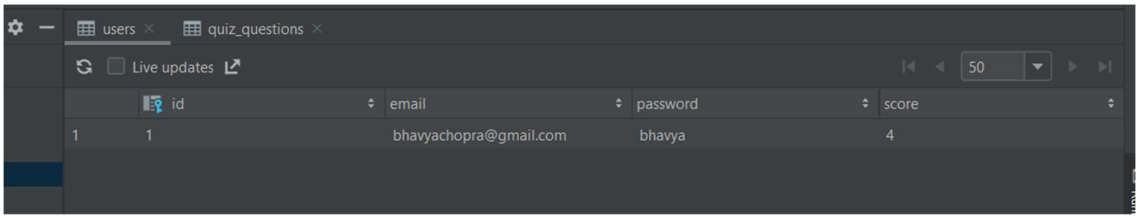

Score: 1
Question: 2/5
00:29
What is CV2
Open CV
Computer Vision 2.0
A library
Confirm



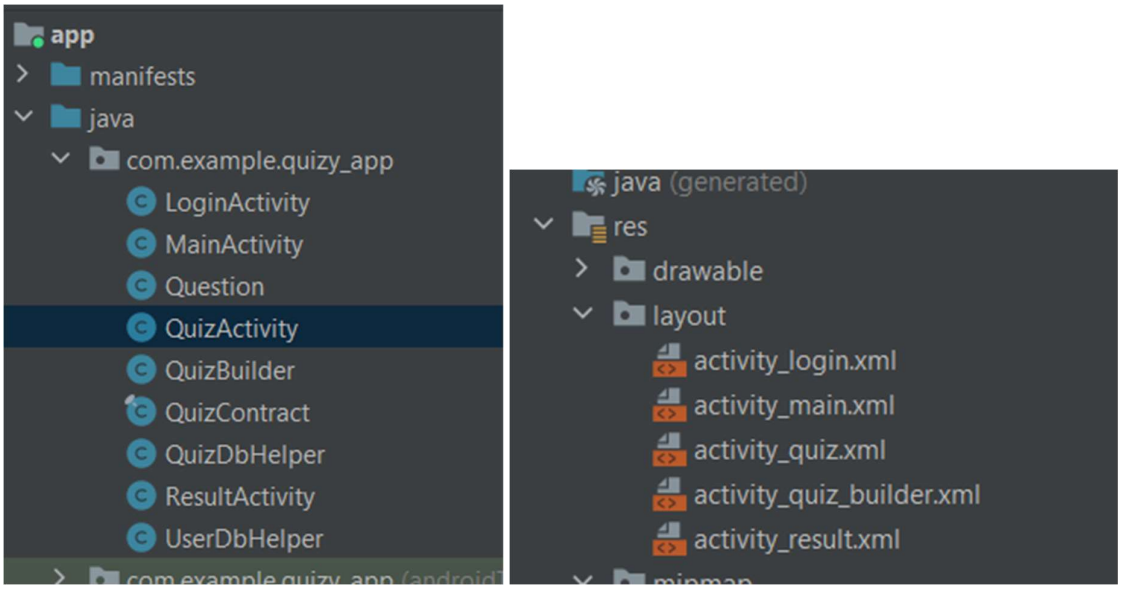


After Playing –




users
quiz_questions
Live updates
50
id
email
password
score
1
1
bhavyachopra@gmail.com
bhavya
4

Project Structure –



Android Manifest.xml –

```
<?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/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Quizy_App"
        tools:targetApi="31">
        <activity android:name=".ResultActivity"
            android:exported="false"/>
        <activity
            android:name=".LoginActivity"
            android:exported="false" />
        <activity
            android:name=".QuizBuilder"
            android:exported="false" />
        <activity
            android:name=".QuizActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```
        </activity>
    </application>

</manifest>
```

LoginActivity.java –

```
package com.example.quizy_app;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.quizy_app.UserDbHelper;

public class LoginActivity extends AppCompatActivity {

    private EditText editTextEmail;
    private EditText editTextPassword;
    private UserDbHelper userDbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        // Initialize UserDbHelper and UI elements
        userDbHelper = new UserDbHelper(this);

        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
    }

    public void login(View view) {
        String email = editTextEmail.getText().toString();
        String password = editTextPassword.getText().toString();

        // Check if the email and password exist in the user database
        if (userDbHelper.checkCredentials(email, password)) {
            // Successful login
            Toast.makeText(this, "Login successful",
Toast.LENGTH_SHORT).show();
            // Start the QuizActivity and pass the user's email as an extra
in the intent
            Intent intent_start = new Intent(this, QuizActivity.class);
            intent_start.putExtra("user_email", email);
            startActivity(intent_start);
            finish();
        } else {
            // Incorrect email or password
            Toast.makeText(this, "Login failed. Incorrect email or
password.", Toast.LENGTH_SHORT).show();
        }
    }

    public void register(View view) {
        String email = editTextEmail.getText().toString();
        String password = editTextPassword.getText().toString();
```

```
        // Check if the email already exists in the user database
        if (userDbHelper.checkEmailExists(email)) {
            Toast.makeText(this, "Email already registered. Please log
in.", Toast.LENGTH_SHORT).show();
        } else {
            // Email is not registered, so insert the new user into the
user database
            if (userDbHelper.insertUser(email, password)) {
                Toast.makeText(this, "Registration successful",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Registration failed. Please try
again.", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

MainActivity.java –

```
package com.example.quizy_app;

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

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView highScoreTextView;
    private UserDbHelper userDbHelper;
    private QuizDbHelper quizDbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        highScoreTextView = findViewById(R.id.text_view_highscore);
        userDbHelper = new UserDbHelper(this);
        quizDbHelper = new QuizDbHelper(this);

        updateHighScoreText();
    }
    public void startQuiz(View view) {
        // Navigate to the LoginActivity to start the quiz
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
    }

    public void goToQuizBuilder(View view) {
        // Navigate to the QuizBuilder activity to add questions
        Intent intent = new Intent(MainActivity.this, QuizBuilder.class);
        startActivity(intent);
    }
```

```
    private void updateHighScoreText() {
        // Fetch the highest score from the user database
        int highestScore = userDbHelper.getHighestScore();
        highScoreTextView.setText("Highscore: " + highestScore);
    }

}
```

Question.java –

```
package com.example.quizy_app;

import java.util.ArrayList;
import java.util.List;

public class Question {

    private String question;
    private String option1;
    private String option2;
    private String option3;
    private List<Integer> answerNrs; // Store multiple correct answers

    public Question() {
        answerNrs = new ArrayList<>();
    }

    public Question(String question, String option1, String option2, String
option3, List<Integer> answerNrs) {
        this.question = question;
        this.option1 = option1;
        this.option2 = option2;
        this.option3 = option3;
        this.answerNrs = answerNrs;
    }

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public String getOption1() {
        return option1;
    }

    public void setOption1(String option1) {
        this.option1 = option1;
    }

    public String getOption2() {
        return option2;
    }

    public void setOption2(String option2) {
        this.option2 = option2;
    }
```

```
    public String getOption3() {
        return option3;
    }

    public void setOption3(String option3) {
        this.option3 = option3;
    }

    public List<Integer> getAnswerNrs() {
        return answerNrs;
    }

    public void setAnswerNrs(List<Integer> answerNrs) {
        this.answerNrs = answerNrs;
    }
}
```

QuizActivity.java –

```
package com.example.quizy_app;

import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.List;
public class QuizActivity extends AppCompatActivity {

    private TextView scoreTextView;
    private TextView questionCountTextView;
    private TextView countdownTextView;
    private TextView questionTextView;
    private LinearLayout checkboxGroup;
    private CheckBox checkboxOption1;
    private CheckBox checkboxOption2;
    private CheckBox checkboxOption3;
    private Button confirmNextButton;

    private List<Question> questionList;
    private int questionCounter;
    private int totalQuestions;
    private Question currentQuestion;
    private int score;

    private CountDownTimer countDownTimer;
    private long timeLeftInMillis;
    private String userEmail;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz);

        Intent intent = getIntent();
        if (intent != null) {
            userEmail = intent.getStringExtra("user_email");
        }

        scoreTextView = findViewById(R.id.text_view_score);
        questionCountTextView =
findViewById(R.id.text_view_question_count);
        countdownTextView = findViewById(R.id.text_view_countdown);
        questionTextView = findViewById(R.id.text_view_question);
        checkboxGroup = findViewById(R.id.checkbox_group);
        checkboxOption1 = findViewById(R.id.checkbox_option1);
        checkboxOption2 = findViewById(R.id.checkbox_option2);
        checkboxOption3 = findViewById(R.id.checkbox_option3);
        confirmNextButton = findViewById(R.id.button_confirm_next);

        QuizDbHelper dbHelper = new QuizDbHelper(this);
        questionList = dbHelper.getAllQuestions();
        totalQuestions = questionList.size();

        showNextQuestion();
    }

    private void showNextQuestion() {
        if (questionCounter < totalQuestions) {
            currentQuestion = questionList.get(questionCounter);

            questionTextView.setText(currentQuestion.getQuestion());
            checkboxOption1.setText(currentQuestion.getOption1());
            checkboxOption2.setText(currentQuestion.getOption2());
            checkboxOption3.setText(currentQuestion.getOption3());

            timeLeftInMillis = 30000;
            startCountDown();

            questionCounter++;
            updateQuestionCount();
        } else {
            UserDbHelper dbHelper = new UserDbHelper(this);
            dbHelper.updateUserScore(userEmail, score);

            finishQuiz();
        }
    }

    private void updateQuestionCount() {
        questionCountTextView.setText("Question: " + questionCounter + "/"
+ totalQuestions);
    }

    private void startCountDown() {
        countDownTimer = new CountDownTimer(timeLeftInMillis, 1000) {
            @Override
            public void onTick(long millisUntilFinished) {
                timeLeftInMillis = millisUntilFinished;
                updateCountdownText();
            }
```

```
            @Override
            public void onFinish() {
                timeLeftInMillis = 0;
                updateCountdownText();
                handleTimeUp();
            }
        }.start();
    }

    private void updateCountdownText() {
        int minutes = (int) (timeLeftInMillis / 1000) / 60;
        int seconds = (int) (timeLeftInMillis / 1000) % 60;
        String timeLeftFormatted = String.format("%02d:%02d", minutes,
seconds);
        countdownTextView.setText(timeLeftFormatted);
    }

    public void confirmNext(View view) {
        int selectedCount = 0;

        if (checkboxOption1.isChecked()) selectedCount++;
        if (checkboxOption2.isChecked()) selectedCount++;
        if (checkboxOption3.isChecked()) selectedCount++;

        if (selectedCount == 0) {
            Toast.makeText(this, "Please select an answer.",
Toast.LENGTH_SHORT).show();
        } else {
            List<Integer> selectedAnswers = new ArrayList<>();
            if (checkboxOption1.isChecked()) {
                selectedAnswers.add(1);
            }
            if (checkboxOption2.isChecked()) {
                selectedAnswers.add(2);
            }
            if (checkboxOption3.isChecked()) {
                selectedAnswers.add(3);
            }

            List<Integer> correctAnswers = currentQuestion.getAnswerNrs();

            if (selectedAnswers.containsAll(correctAnswers) &&
correctAnswers.containsAll(selectedAnswers)) {
                score++;
                scoreTextView.setText("Score: " + score);
            }

            checkboxOption1.setChecked(false);
            checkboxOption2.setChecked(false);
            checkboxOption3.setChecked(false);
            countDownTimer.cancel();
            showNextQuestion();
        }
    }

    private void finishQuiz() {
        Intent intent = new Intent(this, ResultActivity.class);
        intent.putExtra("USER_SCORE", score);
        intent.putExtra("user_email", userEmail);
        startActivity(intent);
```

```
            finish();
        }

        private void handleTimeUp() {
            checkboxOption1.setChecked(false);
            checkboxOption2.setChecked(false);
            checkboxOption3.setChecked(false);
            showNextQuestion();
        }

        @Override
        public void onBackPressed() {
            super.onBackPressed();
        }
}
```

QuizBuilder.java –

```
package com.example.quizy_app;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

import com.example.quizy_app.QuizDbHelper;
import com.example.quizy_app.R;

import java.util.ArrayList;
import java.util.List;

public class QuizBuilder extends AppCompatActivity {
    private EditText editTextQuestion;
    private EditText editTextOption1;
    private EditText editTextOption2;
    private EditText editTextOption3;
    private CheckBox checkBoxOption1;
    private CheckBox checkBoxOption2;
    private CheckBox checkBoxOption3;
    private QuizDbHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_quiz_builder);

        dbHelper = new QuizDbHelper(this);
        editTextQuestion = findViewById(R.id.editTextQuestion);
        editTextOption1 = findViewById(R.id.editTextOption1);
        editTextOption2 = findViewById(R.id.editTextOption2);
        editTextOption3 = findViewById(R.id.editTextOption3);
        checkBoxOption1 = findViewById(R.id.checkBoxOption1);
        checkBoxOption2 = findViewById(R.id.checkBoxOption2);
        checkBoxOption3 = findViewById(R.id.checkBoxOption3);
```

```
        Button buttonAddQuiz = findViewById(R.id.buttonAddQuiz);
        buttonAddQuiz.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addQuiz();
            }
        });
    }

    private void addQuiz() {
        String question = editTextQuestion.getText().toString();
        String option1 = editTextOption1.getText().toString();
        String option2 = editTextOption2.getText().toString();
        String option3 = editTextOption3.getText().toString();

        boolean isOption1Correct = checkBoxOption1.isChecked();
        boolean isOption2Correct = checkBoxOption2.isChecked();
        boolean isOption3Correct = checkBoxOption3.isChecked();

        if (question.isEmpty() || option1.isEmpty() || option2.isEmpty() ||
option3.isEmpty()) {
            Toast.makeText(this, "Please fill in all fields",
Toast.LENGTH_SHORT).show();
        } else if (!isOption1Correct && !isOption2Correct &&
!isOption3Correct) {
            Toast.makeText(this, "Select at least one correct option",
Toast.LENGTH_SHORT).show();
        } else if ((isOption1Correct ? 1 : 0) + (isOption2Correct ? 1 : 0)
+ (isOption3Correct ? 1 : 0) > 2) {
            Toast.makeText(this, "Only two options can be correct",
Toast.LENGTH_SHORT).show();
        } else {
            List<Integer> correctOptions = new ArrayList<>();
            if (isOption1Correct) correctOptions.add(1);
            if (isOption2Correct) correctOptions.add(2);
            if (isOption3Correct) correctOptions.add(3);

            long result =
dbHelper.insertQuestion(dbHelper.getWritableDatabase(),question, option1,
option2, option3, dbHelper.convertListToString(correctOptions));
            if (result != -1) {
                Toast.makeText(this, "Question added to the database",
Toast.LENGTH_SHORT).show();
                clearFields();
            } else {
                Toast.makeText(this, "Error adding question to the
database", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void clearFields() {
        editTextQuestion.setText("");
        editTextOption1.setText("");
        editTextOption2.setText("");
        editTextOption3.setText("");
        checkBoxOption1.setChecked(false);
        checkBoxOption2.setChecked(false);
        checkBoxOption3.setChecked(false);
```

```
    }
}
```

QuizContract.java –

```
package com.example.quizy_app;

import android.provider.BaseColumns;

public final class QuizContract {

    private QuizContract() {
    }

    public static class QuestionsTable implements BaseColumns {
        public static final String TABLE_NAME = "quiz_questions";
        public static final String COLUMN_QUESTION = "question";
        public static final String COLUMN_OPTION1 = "option1";
        public static final String COLUMN_OPTION2 = "option2";
        public static final String COLUMN_OPTION3 = "option3";
        public static final String COLUMN_ANSWER_NR = "answer_nr";
    }
}
```

QuizDbHelper.java

```
package com.example.quizy_app;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class QuizDbHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "Quiz.db";
    private static final int DATABASE_VERSION = 1;

    public QuizDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String SQL_CREATE_QUESTIONS_TABLE = "CREATE TABLE " +
QuizContract.QuestionsTable.TABLE_NAME + " ("
                + QuizContract.QuestionsTable._ID + " INTEGER PRIMARY KEY
AUTOINCREMENT, "
                + QuizContract.QuestionsTable.COLUMN_QUESTION + " TEXT, "
                + QuizContract.QuestionsTable.COLUMN_OPTION1 + " TEXT, "
                + QuizContract.QuestionsTable.COLUMN_OPTION2 + " TEXT, "
                + QuizContract.QuestionsTable.COLUMN_OPTION3 + " TEXT, "
                + QuizContract.QuestionsTable.COLUMN_ANSWER_NR + " TEXT)";
// Use TEXT for multiple answers
```

```
        db.execSQL(SQL_CREATE_QUESTIONS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " +
QuizContract.QuestionsTable.TABLE_NAME);
        onCreate(db);
    }

    public long insertQuestion(SQLiteDatabase db, String question, String
option1, String option2, String option3, String answerNr) {
        ContentValues values = new ContentValues();
        values.put(QuizContract.QuestionsTable.COLUMN_QUESTION, question);
        values.put(QuizContract.QuestionsTable.COLUMN_OPTION1, option1);
        values.put(QuizContract.QuestionsTable.COLUMN_OPTION2, option2);
        values.put(QuizContract.QuestionsTable.COLUMN_OPTION3, option3);
        values.put(QuizContract.QuestionsTable.COLUMN_ANSWER_NR, answerNr);

        long result = db.insert(QuizContract.QuestionsTable.TABLE_NAME,
null, values);
        return result;
    }

    public List<Question> getAllQuestions() {
        List<Question> questionList = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " +
QuizContract.QuestionsTable.TABLE_NAME, null);

        if (cursor.moveToFirst()) {
            do {
                @SuppressLint("Range") String questionText =
cursor.getString(cursor.getColumnIndex(QuizContract.QuestionsTable.COLUMN_Q
UESTION));
                @SuppressLint("Range") String option1 =
cursor.getString(cursor.getColumnIndex(QuizContract.QuestionsTable.COLUMN_O
PTION1));
                @SuppressLint("Range") String option2 =
cursor.getString(cursor.getColumnIndex(QuizContract.QuestionsTable.COLUMN_O
PTION2));
                @SuppressLint("Range") String option3 =
cursor.getString(cursor.getColumnIndex(QuizContract.QuestionsTable.COLUMN_O
PTION3));
                @SuppressLint("Range") String answerNr =
cursor.getString(cursor.getColumnIndex(QuizContract.QuestionsTable.COLUMN_A
NSWER_NR));

                List<Integer> correctAnswers =
convertStringToList(answerNr);

                Question question = new Question(questionText, option1,
option2, option3, correctAnswers);
                questionList.add(question);
            } while (cursor.moveToNext());
        }
        cursor.close();
        return questionList;
    }
```

```
    String convertListToString(List<Integer> list) {
        StringBuilder sb = new StringBuilder();
        for (Integer item : list) {
            sb.append(item);
            sb.append(",");
        }
        sb.setLength(sb.length() - 1); // Remove the trailing comma
        return sb.toString();
    }

    private List<Integer> convertStringToList(String str) {
        List<Integer> list = new ArrayList<>();
        String[] items = str.split(",");
        for (String item : items) {
            list.add(Integer.parseInt(item));
        }
        return list;
    }
}
```

ResultActivity.java –

```
package com.example.quizy_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

public class ResultActivity extends AppCompatActivity {

    private String userEmail;
    private int userScore;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);

        Intent intent = getIntent();
        if (intent != null) {
            userEmail = intent.getStringExtra("user email");
            userScore = intent.getIntExtra("USER_SCORE", 0);
            Toast.makeText(this, String.valueOf(userScore),
Toast.LENGTH_SHORT).show();
        }

        // Display the user's score in a TextView
        TextView textViewScore = findViewById(R.id.text_view_score);
        textViewScore.setText("Your Score: " + userScore);

        // Check and update the highest score if necessary
        updateUserHighScore(userEmail, userScore);
    }

    private void updateUserHighScore(String email, int score) {
        UserDbHelper dbHelper = new UserDbHelper(this);
        int highestScore = dbHelper.getHighestScoreFor(userEmail);
```

```
        if (score > highestScore) {
            dbHelper.updateUserScore(email, score);
            Toast.makeText(this, "New highest score!",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

UserDbHelper.java –

```
package com.example.quizy_app;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class UserDbHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "User.db";
    private static final int DATABASE_VERSION = 1;

    // Define the user table and its columns
    private static final String TABLE_USERS = "users";
    private static final String COLUMN_ID = "id";
    private static final String COLUMN_EMAIL = "email";
    private static final String COLUMN_PASSWORD = "password";
    private static final String COLUMN_SCORE = "score";

    public UserDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Create the user table
        String CREATE_USERS_TABLE = "CREATE TABLE " + TABLE_USERS + " ("
                + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + COLUMN_EMAIL + " TEXT, "
                + COLUMN_PASSWORD + " TEXT, "
                + COLUMN_SCORE + " INTEGER DEFAULT 0)";
        db.execSQL(CREATE_USERS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        // Drop the user table if it exists and recreate it
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
        onCreate(db);
    }

    // Add methods for user database operations
    public boolean insertUser(String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_EMAIL, email);
```

```
        values.put(COLUMN_PASSWORD, password);
        values.put(COLUMN_SCORE, 0); // Initialize score to 0

        long newRowId = db.insert(TABLE_USERS, null, values);

        return newRowId != -1;
    }

    public boolean checkCredentials(String email, String password) {
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_ID};
        String selection = COLUMN_EMAIL + " = ? AND " + COLUMN_PASSWORD + "
= ?";
        String[] selectionArgs = {email, password};

        Cursor cursor = db.query(TABLE_USERS, columns, selection,
selectionArgs, null, null, null);

        boolean loginSuccessful = cursor.getCount() > 0;

        cursor.close();

        return loginSuccessful;
    }

    public boolean checkEmailExists(String email) {
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_ID};
        String selection = COLUMN_EMAIL + " = ?";
        String[] selectionArgs = {email};

        Cursor cursor = db.query(TABLE_USERS, columns, selection,
selectionArgs, null, null, null);

        boolean emailExists = cursor.getCount() > 0;

        cursor.close();

        return emailExists;
    }

    public int getHighestScoreFor(String userEmail) {
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_SCORE};
        String selection = COLUMN_EMAIL + " = ?";
        String[] selectionArgs = {userEmail};
        String orderBy = COLUMN_SCORE + " DESC";

        Cursor cursor = db.query(TABLE_USERS, columns, selection,
selectionArgs, null, null, orderBy);

        int highestScore = 0;

        if (cursor.moveToFirst()) {
            highestScore =
cursor.getInt(cursor.getColumnIndex(COLUMN_SCORE));
        }

        cursor.close();

        return highestScore;
```

```
    }

    @SuppressLint("Range")
    public int getHighestScore() {
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_SCORE};
        String orderBy = COLUMN_SCORE + " DESC";
        Cursor cursor = db.query(TABLE_USERS, columns, null, null, null,
null, orderBy);

        int highestScore = 0;

        if (cursor.moveToFirst()) {
            highestScore =
cursor.getInt(cursor.getColumnIndex(COLUMN_SCORE));
        }

        cursor.close();

        return highestScore;
    }

    public void updateUserScore(String userEmail, int newScore) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(COLUMN_SCORE, newScore);

        String selection = COLUMN_EMAIL + " = ?";
        String[] selectionArgs = { userEmail };

        int updatedRows = db.update(TABLE_USERS, values, selection,
selectionArgs);

    }

}
```

Activity_login.xml –

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="@color/colorBackground"
    android:padding="16dp"
    tools:context="com.example.quizy_app.LoginActivity">>

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:hint="Email"
        android:layout_marginTop="30dp" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextEmail"
        android:layout_centerHorizontal="true"
        android:hint="Password" />

    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextPassword"
        android:layout_centerHorizontal="true"
        android:text="Login"
        android:onClick="login" />

    <Button
        android:id="@+id/buttonRegister"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonLogin"
        android:layout_centerHorizontal="true"
        android:text="Register"
        android:onClick="register" />
</RelativeLayout>
```

Activity_main.xml –

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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/colorBackground"
    android:padding="16dp"
    tools:context="com.example.quizy_app.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"
        android:text="Welcome to QuizMe"
        android:textColor="@android:color/black"
        android:textSize="35sp" />

    <TextView
        android:id="@+id/text_view_highscore"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/button_start_quiz"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="32dp"
        android:text="Highscore: 0"
        android:textSize="20sp" />

    <Button
        android:id="@+id/button_start_quiz"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
```

```
        android:text="Start Quiz"
        android:onClick="startQuiz" />

    <Button
        android:id="@+id/button_add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/button_start_quiz"
        android:layout_marginTop="10dp"
        android:text="Add questions"
        android:layout_centerHorizontal="true"
        android:onClick="goToQuizBuilder" />

</RelativeLayout>
```

Activity_quiz.xml –

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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/colorBackground"
    android:padding="16dp"
    tools:context="com.example.quizy_app.QuizActivity">

    <TextView
        android:id="@+id/text_view_score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Score: 0"
        android:textColor="@android:color/black" />

    <TextView
        android:id="@+id/text_view_question_count"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/text_view_score"
        android:text="Question: 1/x"
        android:textColor="@android:color/black" />

    <TextView
        android:id="@+id/text_view_countdown"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:text="00:30"
        android:textColor="@android:color/black"
        android:textSize="40sp" />

    <TextView
        android:id="@+id/text_view_question"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/checkbox_group"
        android:layout_marginBottom="16dp"
        android:text="Here will be the question text\nHere will be the
question text\nHere will be the question text"
        android:textAlignment="center"
```

```
        android:textColor="@android:color/black"
        android:textSize="20sp" />

    <LinearLayout
        android:id="@+id/checkbox_group"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:orientation="vertical">

        <CheckBox
            android:id="@+id/checkbox_option1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 1" />

        <CheckBox
            android:id="@+id/checkbox_option2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 2" />

        <CheckBox
            android:id="@+id/checkbox_option3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 3" />

    </LinearLayout>

    <Button
        android:id="@+id/button_confirm_next"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/checkbox_group"
        android:layout_marginTop="16dp"
        android:onClick="confirmNext"
        android:text="Confirm" />

</RelativeLayout>
```

Activity_quiz_builder.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".QuizBuilder">

    <EditText
        android:id="@+id/editTextQuestion"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the question"
        android:inputType="text"
        android:layout_marginBottom="16dp"/>
```

```
    <EditText
        android:id="@+id/editTextOption1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Option 1"
        android:inputType="text"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextOption2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Option 2"
        android:inputType="text"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextOption3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Option 3"
        android:inputType="text"
        android:layout_marginBottom="16dp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <CheckBox
            android:id="@+id/checkBoxOption1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 1"/>

        <CheckBox
            android:id="@+id/checkBoxOption2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 2"/>

        <CheckBox
            android:id="@+id/checkBoxOption3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 3"/>
    </LinearLayout>

    <Button
        android:id="@+id/buttonAddQuiz"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add Quiz Question"/>

</LinearLayout>
```

Activity_result.xml –

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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/colorBackground"
    android:padding="16dp"
    tools:context="com.example.quizy_app.ResultActivity">

    <TextView
        android:id="@+id/text_view_score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Your Score: 0"
        android:textSize="24sp"
        android:textColor="@android:color/black" />

</RelativeLayout>
```

Github –

https://github.com/bhavyachopra99/MAD_Quizapp