

1. Gradle Configuration

app/build.gradle.kts (Module: app)

Add the viewBinding feature inside the android { ... } block:

```
// ... inside the android { ... } block
android {
    // ... other configurations

    buildFeatures {
        viewBinding = true
    }

    // ... other configurations
}
// ...
```

2. Kotlin Data Model

app/src/main/java/com/

yourpackage/flashcardquizapp/ FlashcardDeck.kt

(Replace com.yourpackage.flashcardquizapp
with your actual package name.)

```
package com.sosapackage.flashcardquizapp
```

```
data class Flashcard(  
    val question: String,  
    val option1: String,  
    val option2: String,  
    val correctAnswer: String  
)
```

```
object FlashcardDeck {  
    val deck: List<Flashcard> = listOf(  
        Flashcard("The capital of France is Paris.",  
        "True", "False", "True"),  
        Flashcard("Water boils at 100°F.", "True",  
        "False", "False"),  
        Flashcard("Mount Everest is the highest  
mountain on Earth.", "True", "False", "True"),
```

```
    Flashcard("A byte is composed of 16 bits.",  
    "True", "False", "False"),  
    Flashcard("Jupiter is the largest planet in  
our solar system.", "True", "False", "True")  
)  
}
```

3. User Interface Layouts (XML)

**app/src/main/res/layout/
activity_welcome.xml**

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://  
schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/  
tools"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:orientation="vertical"  
        android:gravity="center"  
        android:padding="32dp"  
        tools:context=".WelcomeActivity">
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Welcome to the Flashcard
```

Study App! 

```
    android:textSize="24sp"  
    android:textStyle="bold"  
    android:layout_marginBottom="16dp"/>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Test your knowledge on a  
variety of subjects. Answer all the questions to  
get your score and personalized feedback."
```

```
    android:textSize="16sp"  
    android:gravity="center"  
    android:layout_marginBottom="48dp"/>
```

```
<Button  
    android:id="@+id/btnStart"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

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

```
</LinearLayout>
```

```
app/src/main/res/layout/activity_quiz.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/
tools">
```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center_horizontal"
        android:padding="32dp"
        tools:context=".QuizActivity">
```

```
<TextView
    android:id="@+id/tvQuestionNumber"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:layout_marginBottom="16dp"
```

```
tools:text="Question 1 / 5"/>
```

```
<TextView  
    android:id="@+id/tvQuestion"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="22sp"  
    android:textStyle="italic"  
    android:gravity="center"  
    android:minHeight="100dp"  
    android:layout_marginBottom="32dp"  
    tools:text="Is the sky blue?"/>
```

```
<Button  
    android:id="@+id/btnOption1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="16dp"  
    tools:text="True"/>
```

```
<Button  
    android:id="@+id/btnOption2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"
```

```
    android:layout_marginBottom="32dp"  
    tools:text="False"/>
```

```
<TextView  
    android:id="@+id/tvFeedback"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textSize="20sp"  
    android:textStyle="bold"  
    android:textColor="@color/  
design_default_color_primary"  
    android:layout_marginBottom="32dp"  
    android:visibility="invisible"  
    tools:text="Correct!"  
    tools:visibility="visible"/>
```

```
<Button  
    android:id="@+id/btnNext"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Next Question"  
    android:enabled="false"/>
```

```
</LinearLayout>
```

app/src/main/res/layout/activity_score.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="32dp"
    tools:context=".ScoreActivity">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Quiz Complete! 🎉"
    android:textSize="28sp"
    android:textStyle="bold"
    android:layout_marginBottom="32dp"/>
```

```
<TextView
    android:id="@+id/tvFinalScore"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="48sp"
    android:textStyle="bold"
    android:textColor="@color/
design_default_color_primary"
    android:layout_marginBottom="16dp"
    tools:text="5/5"/>
```

```
<TextView
    android:id="@+id/tvPersonalFeedback"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:gravity="center"
    android:layout_marginBottom="48dp"
    tools:text="Great job! You are a master of
knowledge."/>
```

```
<Button
    android:id="@+id/btnReview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Review Answers"
```

```
        android:layout_marginBottom="16dp"/>

<Button
    android:id="@+id/btnRestart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Over"/>

</LinearLayout>
```

4. Kotlin Activity Logic

app/src/main/java/com/
yourpackage/flashcardquizapp/
WelcomeActivity.kt

package
com.yourpackage.flashcardquizapp

import android.content.Intent
import android.os.Bundle

```
import  
    androidx.appcompat.app.AppCompatActivity  
  
import  
    com.yourpackage.flashcardquizapp.d  
atabinding.ActivityWelcomeBinding
```

```
class WelcomeActivity :  
    AppCompatActivity() {
```

```
    private lateinit var binding:  
        ActivityWelcomeBinding
```

```
    override fun  
        onCreate(savedInstanceState:  
            Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
binding =  
ActivityWelcomeBinding.inflate(layoutInflater)  
setContentView(binding.root)  
  
binding.btnExit.setOnClickListener {  
    val intent = Intent(this,  
QuizActivity::class.java)  
    startActivity(intent)  
}  
}  
}
```

app/src/main/java/com/
yourpackage/flashcardquizapp/
QuizActivity.kt

```
package  
com.yourpackage.flashcardquizapp  
  
import android.content.Intent  
import android.graphics.Color  
import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import  
    androidx.appcompat.app.AppCompatActivity  
import  
    com.yourpackage.flashcardquizapp.da  
tabinding.ActivityQuizBinding  
  
class QuizActivity :  
    AppCompatActivity() {  
  
    private lateinit var binding:
```

```
ActivityQuizBinding
```

```
    private val deck =
```

```
FlashcardDeck.deck
```

```
    private var currentQuestionIndex = 0
```

```
    private var correctAnswers = 0
```

```
    private var isAnswered = false
```

```
    override fun
```

```
onCreate(savedInstanceState:
```

```
Bundle?) {
```

```
super.onCreate(savedInstanceState)
```

```
    binding =
```

```
ActivityQuizBinding.inflate(layoutInflater)
```

```
    setContentView(binding.root)
```

```
    loadQuestion()
```

```
binding.btnExit.setOnClickListener  
{ checkAnswer(it as Button) }
```

```
binding.btnOption2.setOnClickListener  
{ checkAnswer(it as Button) }
```

```
binding.btnNext.setOnClickListener {  
    currentQuestionIndex++  
    if (currentQuestionIndex <  
        deck.size) {  
        loadQuestion()  
    } else {  
        navigateToScoreScreen()  
    }  
}
```

```
private fun loadQuestion() {  
    val currentFlashcard =  
        deck[currentQuestionIndex]  
  
    isAnswered = false  
    binding.tvFeedback.visibility =  
        View.INVISIBLE  
    binding.btnNext.isEnabled = false
```

resetButtonState(binding.btnOption1)

resetButtonState(binding.btnOption2)

```
    binding.tvQuestionNumber.text =  
        getString(R.string.question_counter,  
        currentQuestionIndex + 1, deck.size)  
    binding.tvQuestion.text =  
        currentFlashcard.question
```

```
    binding.btnOption1.text =  
    currentFlashcard.option1  
    binding.btnOption2.text =  
    currentFlashcard.option2  
}
```

```
private fun  
checkAnswer(selectedButton: Button)  
{  
    if (isAnswered) return  
  
    isAnswered = true  
    binding.btnNext.isEnabled = true  
    val currentFlashcard =  
deck[currentQuestionIndex]  
    val selectedAnswer =  
selectedButton.text.toString()  
  
    binding.btnOption1.isEnabled =
```

```
false
```

```
    binding.btnOption2.isEnabled =  
false
```

```
if (selectedAnswer ==  
currentFlashcard.correctAnswer) {  
    correctAnswers++  
    binding.tvFeedback.text =  
"Correct! 
```

```
selectedButton.setBackgroundColor(C  
olor.GREEN)  
} else {  
    binding.tvFeedback.text = "Try  
Again! 
```

```
selectedButton.setBackgroundColor(C  
olor.RED)
```

```
    val correctButton = if  
(currentFlashcard.option1 ==  
currentFlashcard.correctAnswer)  
binding.btnOption1 else  
binding.btnOption2
```

```
correctButton.setBackgroundColor(Col  
or.parseColor("#ADD8E6")) // Light  
Blue for correct answer
```

```
}
```

```
    binding.tvFeedback.visibility =  
View.VISIBLE
```

```
}
```

```
private fun resetButtonState(button:  
Button) {  
    button.isEnabled = true
```

```
button.setBackgroundColor(Color.GRAY) // Use a default color
}

private fun
navigateToScoreScreen() {
    val intent = Intent(this,
ScoreActivity::class.java).apply {
        putExtra("SCORE",
correctAnswers)
        putExtra("TOTAL_QUESTIONS",
deck.size)
    }
    startActivity(intent)
    finish()
}
}
```

app/src/main/java/com/

yourpackage/flashcardquizapp/
ScoreActivity.kt

package
com.yourpackage.flashcardquizapp

import android.content.Intent
import android.os.Bundle
import
androidx.appcompat.app.AppCompatActivity
import
com.yourpackage.flashcardquizapp.da
tabinding.ActivityScoreBinding

class ScoreActivity :
AppCompatActivity() {

private lateinit var binding:

ActivityScoreBinding

```
override fun  
onCreate(savedInstanceState:  
Bundle?) {  
  
super.onCreate(savedInstanceState)  
binding =  
ActivityScoreBinding.inflate(layoutInfla  
ter)  
setContentView(binding.root)  
  
val score =  
intent.getIntExtra("SCORE", 0)  
val totalQuestions =  
intent.getIntExtra("TOTAL_QUESTIONS  
", FlashcardDeck.deck.size)  
  
binding.tvFinalScore.text =
```

"\$score/\$totalQuestions"

```
    val percentage =  
        (score.toDouble() /  
         totalQuestions.toDouble()) * 100  
  
    val feedback = when {  
        percentage == 100.0 ->  
        "Amazing! You're a true flashcard  
        master! ⭐"  
        percentage >= 75.0 -> "Great  
        job! Keep practising to reach 100%.  
        👍"  
        percentage >= 50.0 -> "Good  
        effort! A little more practice will help  
        you excel. 📚"  
        else -> "Keep practising! Review  
        the materials and try again. You'll get  
        there! 💪"  
    }
```

```
binding.tvPersonalFeedback.text  
= feedback
```

```
binding.btnReview.setOnClickListener  
{  
    // Simplified action for review  
button: restart the quiz  
    startActivity(Intent(this,  
WelcomeActivity::class.java))  
    finish()  
}
```

```
binding.btnRestart.setOnClickListener  
{  
    startActivity(Intent(this,  
WelcomeActivity::class.java))  
    finish()
```

```
    }  
}  
}  
}
```

5. Resources and Manifest

app/src/main/res/values/
strings.xml

<resources>

 <string

 name="app_name">Flashcard
 QuizApp</string>

 <string

 name="question_counter">Qu

estion %1\$d / %2\$d</string>
</resources>

app/src/main/

AndroidManifest.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.FlashcardQuizApp"
        tools:targetApi="31"

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

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

<activity
```

```
    android:name=".WelcomeActivity"
```

```
    android:exported="true">
```

```
        <intent-filter>
```

```
            <action
```

```
                android:name="android.intent.action.MAIN" />
```

```
            <category
```

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

```
        </intent-filter>
```

```
    </activity>
```

```
    </application>
```

```
</manifest>
```

