Proposal:

Learning your second programing language via Project-based learning using Kotlin

Assumption:

1 language has been acquired: Example Python.

Hands down, Python is considered the #1 language for beginners. And why not? Higher level language, which taught generally in the traditional approach is still a good means to get the first step into programming.

Breakdown:  
What is project-based learning?

Why is it better that Traditional?

Why Kotlin? A simple background of Kotlin

Java is ranked number #2 for beginners, and kotlin is the new and upcoming language

How would kotlin benefit the student?

* Android development is moving to Kotlin, though Java is still used as well.
* Student needs a means to show-case their skill. Creating an actual object is always the best way to build confidence.
* A one-time $25 cost is what it takes to upload an app, as long as google policy criteria are met. Versus $99 per year on apple.

Tutorial Steps:

1. Create a new project with empty activity.
   1. File -> New Project -> Empty Activity -> [Name your application], Language=Kotlin -> Finish
2. Design your database by Creating a new Kotlin file named “MyDBHelper.kt”
   1. From your Android file view go to the folder where your MainActivity.kt is, right click on the folder
      1. New -> Kotlin File/Class -> Name= “MyDBHelper”
      2. Use the below as a template to Create Tables, and Insert a few default data to test with.

(We will worry about adding and searching later)

import android.content.Context  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
  
  
class MyDBHelper (context: Context): SQLiteOpenHelper(context, "USERDB", null, 1 ) {  
 override fun onCreate(db: SQLiteDatabase?) {  
 //CREATE A TABLE WHERE primary key will be product barcode  
 db?.execSQL("CREATE TABLE IF NOT EXISTS PRODUCT(P\_BARCODE TEXT PRIMARY KEY , DESCRIPTION TEXT)")  
 db?.execSQL("CREATE TABLE IF NOT EXISTS STORE(STORE\_ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT)")  
 db?.execSQL("CREATE TABLE IF NOT EXISTS ORDERS(ORDER\_ID INTEGER PRIMARY KEY AUTOINCREMENT, TIMESTAMP DATETIME DEFAULT CURRENT\_TIMESTAMP, STORE\_ID INTEGER NOT NULL REFERENCES STORE(STORE\_ID), ORDER\_STATUS TEXT)")  
 db?.execSQL("CREATE TABLE IF NOT EXISTS ORDER\_ITEM(ORDER\_ID INTEGER NOT NULL REFERENCES ORDERS(ORDER\_ID), P\_BARCODE TEXT NOT NULL REFERENCES PRODUCT(P\_BARCODE), PRIMARY KEY(ORDER\_ID, P\_BARCODE) )")  
  
  
 //just some data added as default input  
 db?.execSQL("INSERT INTO PRODUCT(P\_BARCODE, DESCRIPTION) VALUES('09661975680','Kirkland 500ml Purified Water')")  
 db?.execSQL("INSERT INTO PRODUCT(P\_BARCODE, DESCRIPTION) VALUES('381371151035','Aveeno Daily Moisturizing Lotion 20fl oz')")  
 db?.execSQL("INSERT INTO PRODUCT(P\_BARCODE, DESCRIPTION) VALUES('667550822959','PocketBac Anti-Bacterial Hand gel 1fl oz')")  
  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
  
 }  
}

1. Design the Home Page (MainActivity.kt & activity\_main.xml)
   1. XML is the layout design (Frontend design)
      1. Create Buttons as needed for: *(Drag and drop buttons using Design view, or choose Split view and use the code down below to be a rough blueprint)*
         1. Cart
         2. Look up
         3. Exit
      2. You can create TextView as needed to display info about the App

<?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"  
 tools:context=".MainActivity">  
  
  
 <TextView  
 android:id="@+id/txtViewTitle"  
 android:layout\_width="0dp" //Width 0dp when constrained to the sides of the Screen (parent)  
 android:layout\_height="wrap\_content" //Height of text block matches the content. (multi-line = more height)  
 android:gravity="center" //Center is to have the writing be oriented in the middle of the screen  
 android:text="Main Activity" //Sort of a header under the Title of your app  
 android:textSize="25sp"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btnCart"  
 android:layout\_width="match\_parent" //match\_parent is the simpler way to stretch the size of button to

//screen Width  
 android:layout\_height="wrap\_content"   
 android:text="Shopping Cart" //What should the button say  
 app:layout\_constraintBottom\_toTopOf="@+id/btnLookup"  
 tools:layout\_editor\_absoluteX="0dp" />  
  
 <Button  
 android:id="@+id/btnLookup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Look up Receipt"   
 app:layout\_constraintBottom\_toTopOf="@+id/btnCloseApp"  
 tools:layout\_editor\_absoluteX="0dp" />  
  
 <Button  
 android:id="@+id/btnCloseApp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Close APP"   
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <TextView  
 android:id="@+id/txtViewResult"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Home Page" //Another Text space  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/btnCart"  
 app:layout\_constraintTop\_toBottomOf="@+id/txtViewTitle"  
 tools:layout\_editor\_absoluteX="175dp" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

* 1. .kt is the backend that connects it all
     1. Setup OnClickListeners for the buttons and map it to their separate Activities
     2. The buttons for Look Up will be different, as we wish to get back results (receipt information from our search)

An example code is below to setup your buttons

import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import kotlinx.android.synthetic.main.activity\_main.\*  
import kotlin.system.exitProcess  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
  
  
 //go to activity Cart. Here you will make virtual purchase and store it in database  
 btnCart.setOnClickListener **{** val intent = Intent(this, Cart::class.*java*)  
 startActivity(intent)  
 **}** //go to activity LookUp. Get back search result  
 btnLookup.setOnClickListener **{  
  
 }** //Close out of app by connecting it to a button  
 btnCloseApp.setOnClickListener **{** this@MainActivity.finish()  
 *exitProcess*(0)  
 **}** }  
}

1. Create new Activities for:
   1. Cart
   2. Look up

Right click on folder where on .kt files are -> New -> Activity -> Gallery… -> Empty Activity -> Name= Whatever seems sensible -> Finish