****

**UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB**

**Assignment 4**

Name: Jayannthan P T

Dept: CSE ‘A’

Roll No.: 205001049

Develop a Product information application in Android that enables to perform CRUD

operations on data stored in SQLite Database.

In main activity display the following buttons: Create, Insert, Update, Delete, Retrieve, Retrieve All

1. On clicking Create Button, create a new database to store the Product details. (Use SQLite Database)
   1. Product ID (Make this field as primary key)
   2. Product Name
   3. Product Brand
   4. Description
   5. Product Price
2. On Clicking Insert, move to a new view which contains the following details: (Insert new Product to the database)
   1. Product ID (EditText-Validation checking- 4 digit Numbers)
   2. Product Name (Spinner)
   3. Product Brand (RadioButton)
   4. Description (EditText-Alphanumeric characters)
   5. Product Price (EditText-Validation checking Numbers)
   6. Submit (Button) – On press, Insert the data into database.
3. On clicking Update, move to a new view which contains above details and Update Product Price using Product ID.
4. On clicking Delete, Delete the whole row in the table by Product ID.
5. On clicking Retrieve, Retrieve the product using Product ID.

6. On clicking Retrieve All, retrieve the details of all the products in a particular brand.

**Ex. No:4**

**Date:**30/9/2023

**Title of the Program:** Android Application Development using Database

**Objective:**

The objective of the provided Android application is to manage a list of products. Users can perform various operations such as inserting new products, updating product prices, deleting products, searching for product details by ID, and viewing products grouped by brand.

**Algorithm:**

1. MainActivity:
   * Set up buttons for inserting, deleting, updating, searching, and viewing products.
   * On button click, navigate to the respective activities.
2. InsertActivity:
   * Provide input fields for product ID, name, brand, description, and price.
   * On the "INSERT" button click:
     + Validate user input.
     + Insert a new product into the database.
3. UpdateActivity:
   * Allow users to input a product ID and a new price.
   * On the "UPDATE" button click:
     + Validate user input.
     + Update the price of the specified product in the database.
4. DeleteActivity:
   * Allow users to input a product ID for deletion.
   * On the "DELETE" button click:
     + Validate user input.
     + Delete the specified product from the database.
5. SearchActivity:
   * Enable users to search for product details by entering a product ID.
   * On the "SEARCH" button click:
     + Retrieve and display product details if found.
6. ViewActivity:
   * Display a list of product brands in a ListView.
   * On selecting a brand:
     + Retrieve and display a list of products for that brand.
7. DatabaseHelper:
   * Manage the SQLite database for storing product information.
   * Define database schema and operations for creating, querying, updating, and deleting records.

**Features used:**

* Android activities for user interaction.
* SQLite database for data storage.
* Input validation for user data.
* Navigation between activities.
* ListView for displaying lists of items.
* Custom XML layouts for UI design.

**Source code:**

* MainActivity.java

package com.example.products;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button insertButton;

    Button deleteButton;

    Button updateButton;

    Button searchButton;

    Button viewButton;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.activity\_main);

        // Initialize buttons inside the onCreate method

        insertButton = **findViewById**(R.id.insert);

        deleteButton = **findViewById**(R.id.delete);

        updateButton = **findViewById**(R.id.update);

        searchButton = **findViewById**(R.id.search);

        viewButton = **findViewById**(R.id.view);

        insertButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View v) {

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

**startActivity**(intent);

            }

        });

        deleteButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View v) {

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

**startActivity**(intent);

            }

        });

        updateButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View v) {

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

**startActivity**(intent);

            }

        });

        searchButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View v) {

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

**startActivity**(intent);

            }

        });

        viewButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View v) {

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

**startActivity**(intent);

            }

        });

    }

}

* 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" tools:context=".MainActivity" tools:ignore="ExtraText">

    <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Product Database" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.06" android:textSize="35sp" />

    <Button android:id="@+id/insert" android:layout\_width="122dp" android:layout\_height="55dp" android:text="Insert" app:layout\_constraintTop\_toBottomOf="@+id/Heading" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp" android:textSize="20sp" />

    <Button android:id="@+id/delete" android:layout\_width="122dp" android:layout\_height="55dp" android:text="Delete" app:layout\_constraintTop\_toBottomOf="@+id/insert" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp" android:textSize="20sp" />

    <Button android:id="@+id/update" android:layout\_width="122dp" android:layout\_height="55dp" android:text="Update" app:layout\_constraintTop\_toBottomOf="@+id/delete" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp" android:textSize="20sp" />

    <Button android:id="@+id/search" android:layout\_width="122dp" android:layout\_height="55dp" android:text="Search" app:layout\_constraintTop\_toBottomOf="@+id/update" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp" android:textSize="20sp" />

    <Button android:id="@+id/view" android:layout\_width="122dp" android:layout\_height="55dp" android:text="View" app:layout\_constraintTop\_toBottomOf="@+id/search" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp" android:textSize="20sp" />

</androidx.constraintlayout.widget.ConstraintLayout>

* InsertActivity.java

package com.example.products;

import android.content.ContentValues;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class InsertActivity extends AppCompatActivity {

    private EditText productIdEditText, productNameEditText, productBrandEditText, productDescriptionEditText, productPriceEditText;

    private Button submitButton;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.insert);

        // Initialize UI elements

        productIdEditText = **findViewById**(R.id.id);

        productNameEditText = **findViewById**(R.id.name);

        productBrandEditText = **findViewById**(R.id.brand);

        productDescriptionEditText = **findViewById**(R.id.desc);

        productPriceEditText = **findViewById**(R.id.price);

        submitButton = **findViewById**(R.id.submit);

        // Set click listener for the Submit button

        submitButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View view) {

                // Get user input

                String productId = productIdEditText.**getText**().**toString**();

                String productName = productNameEditText.**getText**().**toString**();

                String productBrand = productBrandEditText.**getText**().**toString**();

                String productDescription = productDescriptionEditText.**getText**().**toString**();

                String productPriceStr = productPriceEditText.**getText**().**toString**();

                // Check if any of the fields are empty

                if (productId.**isEmpty**() || productName.**isEmpty**() || productBrand.**isEmpty**() || productDescription.**isEmpty**() || productPriceStr.**isEmpty**()) {

                    Toast.**makeText**(InsertActivity.this, "Please fill in all fields", Toast.LENGTH\_SHORT).**show**();

                    return;

                }

                // Convert product price to double

                double productPrice = Double.**parseDouble**(productPriceStr);

                // Insert data into the database

**insertProduct**(productId, productName, productBrand, productDescription, productPrice);

                // Show success message

                Toast.**makeText**(InsertActivity.this, "Product inserted successfully", Toast.LENGTH\_SHORT).**show**();

                // Clear input fields

                productIdEditText.**getText**().**clear**();

                productNameEditText.**getText**().**clear**();

                productBrandEditText.**getText**().**clear**();

                productDescriptionEditText.**getText**().**clear**();

                productPriceEditText.**getText**().**clear**();

            }

        });

    }

    private void **insertProduct**(String productId, String productName, String productBrand, String productDescription, double productPrice) {

        // Create or open the database for writing

        DatabaseHelper dbHelper = new **DatabaseHelper**(this);

        SQLiteDatabase db = dbHelper.**getWritableDatabase**();

        // Create a new map of values, where column names are the keys

        ContentValues values = new **ContentValues**();

        values.**put**(DatabaseContract.ProductEntry.COLUMN\_PRODUCT\_ID, productId);

        values.**put**(DatabaseContract.ProductEntry.COLUMN\_PRODUCT\_NAME, productName);

        values.**put**(DatabaseContract.ProductEntry.COLUMN\_PRODUCT\_BRAND, productBrand);

        values.**put**(DatabaseContract.ProductEntry.COLUMN\_PRODUCT\_DESCRIPTION, productDescription);

        values.**put**(DatabaseContract.ProductEntry.COLUMN\_PRODUCT\_PRICE, productPrice);

        // Insert the new row, returning the primary key value of the new row

        long newRowId = db.**insert**(DatabaseContract.ProductEntry.TABLE\_NAME, null, values);

        // Close the database

        db.**close**();

    }

}

* Insert.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:id="@+id/linearLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

    <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Insert" android:textSize="35sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.06" />

    <TableLayout android:layout\_width="0dp" android:layout\_height="0dp" app:layout\_constraintTop\_toBottomOf="@+id/Heading" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintBottom\_toBottomOf="parent" android:layout\_marginTop="16dp">

        <!-- First Row -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <!-- Second Row -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Name" android:gravity="center" />

            <EditText android:id="@+id/name" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <!-- Third Row (and so on) -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Brand" android:gravity="center" />

            <EditText android:id="@+id/brand" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Description" android:gravity="center" />

            <EditText android:id="@+id/desc" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Price" android:gravity="center" />

            <EditText android:id="@+id/price" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="numberDecimal" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <Button android:id="@+id/submit" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="INSERT" android:background="@color/black" android:textColor="@color/white"/>

        </TableRow>

    </TableLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

* UpdateActivity.java

package com.example.products;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class UpdateActivity extends AppCompatActivity {

    private EditText productIdEditText;

    private EditText newPriceEditText;

    private Button updateButton;

    private DatabaseHelper dbHelper;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.update);

        productIdEditText = **findViewById**(R.id.id);

        newPriceEditText = **findViewById**(R.id.price);

        updateButton = **findViewById**(R.id.submit);

        dbHelper = new **DatabaseHelper**(this);

        updateButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View view) {

                // Get the product ID and new price from the EditText fields

                String productIdText = productIdEditText.**getText**().**toString**().**trim**();

                String newPriceText = newPriceEditText.**getText**().**toString**().**trim**();

                if (productIdText.**isEmpty**() || newPriceText.**isEmpty**()) {

                    // Product ID or new price field is empty, show an error message

                    Toast.**makeText**(UpdateActivity.this, "Product ID and new price are required", Toast.LENGTH\_SHORT)

                            .**show**();

                } else {

                    try {

                        // Parse the product ID and new price as needed (integer and double)

                        int productId = Integer.**parseInt**(productIdText);

                        double newPrice = Double.**parseDouble**(newPriceText);

                        // Update the product price in the database

                        boolean isUpdated = dbHelper.**updateProductPrice**(productId, newPrice);

                        if (isUpdated) {

                            // Show a success message

                            Toast.**makeText**(UpdateActivity.this, "Price updated successfully", Toast.LENGTH\_SHORT)

                                    .**show**();

                        } else {

                            // Show an error message if the update failed

                            Toast.**makeText**(UpdateActivity.this, "Failed to update price. Product not found.",

                                    Toast.LENGTH\_SHORT).**show**();

                        }

                        // Clear the EditText fields

                        productIdEditText.**setText**("");

                        newPriceEditText.**setText**("");

                    } catch (NumberFormatException e) {

                        // Handle invalid input (non-integer or non-double)

                        Toast.**makeText**(UpdateActivity.this, "Invalid Product ID or new price", Toast.LENGTH\_SHORT)

                                .**show**();

                    }

                }

            }

        });

    }

}

* update.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:id="@+id/linearLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

    <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Update" android:textSize="35sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout\_width="0dp" android:layout\_height="wrap\_content" app:layout\_constraintTop\_toBottomOf="@+id/Heading" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp">

        <!-- Row for Product ID Search -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Price" android:gravity="center" />

            <EditText android:id="@+id/price" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="numberDecimal" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <Button android:id="@+id/submit" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="UPDATE PRICE" android:background="@color/black" android:textColor="@color/white" app:layout\_constraintTop\_toBottomOf="@id/price" />

        </TableRow>

    </TableLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

* DeleteActivity.java

package com.example.products;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class DeleteActivity extends AppCompatActivity {

    private EditText productIdEditText;

    private Button deleteButton;

    private DatabaseHelper dbHelper;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.delete);

        productIdEditText = **findViewById**(R.id.id);

        deleteButton = **findViewById**(R.id.submit);

        dbHelper = new **DatabaseHelper**(this);

        deleteButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View view) {

                // Get the product ID from the EditText

                String productIdText = productIdEditText.**getText**().**toString**().**trim**();

                if (productIdText.**isEmpty**()) {

                    // Product ID field is empty, show an error message

                    Toast.**makeText**(DeleteActivity.this, "Product ID is required", Toast.LENGTH\_SHORT).**show**();

                } else {

                    try {

                        // Parse the product ID as an integer

                        int productId = Integer.**parseInt**(productIdText);

                        // Attempt to delete the product by ID

                        boolean isDeleted = dbHelper.**deleteProductById**(productId);

                        if (isDeleted) {

                            // Product deleted successfully

                            Toast.**makeText**(DeleteActivity.this, "Product deleted successfully", Toast.LENGTH\_SHORT).**show**();

                            // Clear the product ID EditText

                            productIdEditText.**setText**("");

                        } else {

                            // Product not found, show an error message

                            Toast.**makeText**(DeleteActivity.this, "Product not found for deletion", Toast.LENGTH\_SHORT).**show**();

                        }

                    } catch (NumberFormatException e) {

                        // Handle invalid input (non-integer)

                        Toast.**makeText**(DeleteActivity.this, "Invalid Product ID", Toast.LENGTH\_SHORT).**show**();

                    }

                }

            }

        });

    }

}

* delete.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:id="@+id/linearLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

    <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Delete" android:textSize="35sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout\_width="0dp" android:layout\_height="wrap\_content" app:layout\_constraintTop\_toBottomOf="@+id/Heading" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp">

        <!-- Row for Product ID Search -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <Button android:id="@+id/submit" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="DELETE" android:background="@color/black" android:textColor="@color/white" app:layout\_constraintTop\_toBottomOf="@id/id" />

        </TableRow>

    </TableLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

* SearchActivity.java

package com.example.products;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TableLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class SearchActivity extends AppCompatActivity {

    private EditText productIdEditText;

    private TableLayout outputTable;

    private TextView nameTextView;

    private TextView brandTextView;

    private TextView descriptionTextView;

    private TextView priceTextView;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.search);

        // Initialize UI elements

        productIdEditText = **findViewById**(R.id.id);

        outputTable = **findViewById**(R.id.outputTable);

        nameTextView = **findViewById**(R.id.name);

        brandTextView = **findViewById**(R.id.brand);

        descriptionTextView = **findViewById**(R.id.desc);

        priceTextView = **findViewById**(R.id.price);

        Button searchButton = **findViewById**(R.id.submit);

        searchButton.**setOnClickListener**(new View.**OnClickListener**() {

            @Override

            public void **onClick**(View view) {

**searchProduct**();

            }

        });

    }

    private void **searchProduct**() {

        // Get the product ID entered by the user

        String productIdText = productIdEditText.**getText**().**toString**();

        if (productIdText.**isEmpty**()) {

            // Product ID field is empty, show an error message or handle it as needed

            Toast.**makeText**(this, "Please enter a Product ID", Toast.LENGTH\_SHORT).**show**();

            return;

        }

        int productId = Integer.**parseInt**(productIdText);

        // Use the DatabaseHelper to search for the product by ID

        DatabaseHelper databaseHelper = new **DatabaseHelper**(this);

        Product product = databaseHelper.**searchProductById**(productId);

        if (product != null) {

            // Display the product information in the outputTable

            nameTextView.**setText**(product.**getName**());

            brandTextView.**setText**(product.**getBrand**());

            descriptionTextView.**setText**(product.**getDescription**());

            priceTextView.**setText**(String.**valueOf**(product.**getPrice**()));

            // Make the outputTable visible

            outputTable.**setVisibility**(View.VISIBLE);

        } else {

            // Product not found, you can display an error message or handle it as needed

            Toast.**makeText**(this, "Product not found", Toast.LENGTH\_SHORT).**show**();

            // For now, let's hide the outputTable

            outputTable.**setVisibility**(View.GONE);

        }

    }

}

* search.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:id="@+id/linearLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

    <!-- Heading TextView -->

    <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Search" android:textSize="35sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout\_width="0dp" android:layout\_height="wrap\_content" app:layout\_constraintTop\_toBottomOf="@+id/Heading" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" android:layout\_marginTop="16dp">

        <!-- Row for Product ID Search -->

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <!-- Row for Search Button -->

        <TableRow android:layout\_height="wrap\_content">

            <Button android:id="@+id/submit" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Search" android:background="@color/black" android:textColor="@color/white" app:layout\_constraintTop\_toBottomOf="@id/id" />

        </TableRow>

    </TableLayout>

    <TableLayout android:id="@+id/outputTable" android:layout\_width="0dp" android:layout\_height="0dp" app:layout\_constraintTop\_toBottomOf="@+id/searchTable" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintBottom\_toBottomOf="parent" android:layout\_marginTop="16dp" android:visibility="gone">

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Name" android:gravity="center" />

            <EditText android:id="@+id/name" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Brand" android:gravity="center" />

            <EditText android:id="@+id/brand" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Description" android:gravity="center" />

            <EditText android:id="@+id/desc" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="text" android:text="" />

        </TableRow>

        <TableRow android:layout\_height="wrap\_content">

            <TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="Product Price" android:gravity="center" />

            <EditText android:id="@+id/price" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:ems="10" android:inputType="numberDecimal" android:text="" />

        </TableRow>

    </TableLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

* ViewActivity.java

package com.example.products;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.LinearLayout;

import android.widget.ListView;

import android.widget.TextView;

import java.util.List;

public class ViewActivity extends AppCompatActivity {

    private DatabaseHelper databaseHelper;

    private LinearLayout rightSideLayout;

    @Override

    protected void **onCreate**(Bundle savedInstanceState) {

        super.**onCreate**(savedInstanceState);

**setContentView**(R.layout.view);

        ListView brandListView = **findViewById**(R.id.brandListView);

        rightSideLayout = **findViewById**(R.id.rightSideLayout);

        databaseHelper = new **DatabaseHelper**(this);

        List<String> brandList = databaseHelper.**getBrandList**();

        ArrayAdapter<String> adapter = new **ArrayAdapter**<>(this, android.R.layout.simple\_list\_item\_1, brandList);

        brandListView.**setAdapter**(adapter);

        brandListView.**setOnItemClickListener**(new AdapterView.**OnItemClickListener**() {

            @Override

            public void **onItemClick**(AdapterView<?> parent, android.view.View view, int position, long id) {

                String selectedBrand = brandList.**get**(position);

                List<Product> productList = databaseHelper.**getProductsByBrand**(selectedBrand);

                // Clear the existing product information

                rightSideLayout.**removeAllViews**();

                // ...

                if (!productList.**isEmpty**()) {

                    for (Product product : productList) {

                        // Create a TextView for each product

                        TextView productTextView = new **TextView**(ViewActivity.this);

                        productTextView.**setLayoutParams**(new ViewGroup.**LayoutParams**(

                                ViewGroup.LayoutParams.MATCH\_PARENT,

                                ViewGroup.LayoutParams.WRAP\_CONTENT // Set height to wrap\_content

                        ));

                        productTextView.**setText**("Product ID: " + product.**getId**() + "\n" +

                                "Product Name: " + product.**getName**() + "\n" +

                                "Product Price: " + product.**getPrice**());

                        // Add the TextView to the rightSideLayout

                        rightSideLayout.**addView**(productTextView);

                        // Add space (spacer TextView)

                        TextView spacer = new **TextView**(ViewActivity.this);

                        spacer.**setLayoutParams**(new ViewGroup.**LayoutParams**(

                                ViewGroup.LayoutParams.MATCH\_PARENT,

                                50

                        ));

                        rightSideLayout.**addView**(spacer);

                    }

                } else {

                    // Display a message if no products are available for the selected brand

                    TextView noProductsTextView = new **TextView**(ViewActivity.this);

                    noProductsTextView.**setLayoutParams**(new ViewGroup.**LayoutParams**(

                            ViewGroup.LayoutParams.MATCH\_PARENT,

                            ViewGroup.LayoutParams.WRAP\_CONTENT

                    ));

                    noProductsTextView.**setText**("No products available for this brand.");

                    // Add the message to the rightSideLayout

                    rightSideLayout.**addView**(noProductsTextView);

                }

            }

        });

    }

}

* view.xml

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

<androidx.core.widget.NestedScrollView 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:id="@+id/linearLayout" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

    <LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:orientation="vertical">

        <!-- Heading TextView -->

        <TextView android:id="@+id/Heading" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="View" android:textSize="35sp" android:layout\_gravity="center" android:layout\_marginTop="16dp" android:layout\_marginBottom="16dp" />

        <LinearLayout android:layout\_width="match\_parent" android:layout\_height="600dp" android:orientation="horizontal">

            <!-- Left side: List of Brands -->

            <ListView android:id="@+id/brandListView" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:divider="@android:color/darker\_gray" android:dividerHeight="1dp" />

            <View android:layout\_width="1dp" android:layout\_height="match\_parent" android:background="@android:color/darker\_gray" />

            <LinearLayout android:id="@+id/rightSideLayout" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:orientation="vertical" android:layout\_weight="1">

            </LinearLayout>

        </LinearLayout>

    </LinearLayout>

</androidx.core.widget.NestedScrollView>

* DatabaseHelper.java

// DatabaseHelper.java

package com.example.products;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

import java.util.ArrayList;

import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE\_NAME = "products";

    private static final int DATABASE\_VERSION = 1;

    public static final String TABLE\_NAME = "products";

    public static final String COLUMN\_PRODUCT\_ID = "product\_id";

    public static final String COLUMN\_PRODUCT\_NAME = "product\_name";

    public static final String COLUMN\_PRODUCT\_BRAND = "product\_brand";

    public static final String COLUMN\_PRODUCT\_DESCRIPTION = "product\_description";

    public static final String COLUMN\_PRODUCT\_PRICE = "product\_price";

    private static final String SQL\_CREATE\_PRODUCT\_TABLE =

            "CREATE TABLE " + TABLE\_NAME + " (" +

                    COLUMN\_PRODUCT\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +

                    COLUMN\_PRODUCT\_NAME + " TEXT," +

                    COLUMN\_PRODUCT\_BRAND + " TEXT," +

                    COLUMN\_PRODUCT\_DESCRIPTION + " TEXT," +

                    COLUMN\_PRODUCT\_PRICE + " REAL)";

    private static final String SQL\_DELETE\_PRODUCT\_TABLE =

            "DROP TABLE IF EXISTS " + TABLE\_NAME;

    public **DatabaseHelper**(Context context) {

        super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

    }

    @Override

    public void **onCreate**(SQLiteDatabase db) {

        db.**execSQL**(SQL\_CREATE\_PRODUCT\_TABLE);

    }

    @Override

    public void **onUpgrade**(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.**execSQL**(SQL\_DELETE\_PRODUCT\_TABLE);

**onCreate**(db);

    }

    public Product **searchProductById**(int productId) {

        SQLiteDatabase db = this.**getReadableDatabase**();

        String[] projection = {

                COLUMN\_PRODUCT\_NAME,

                COLUMN\_PRODUCT\_BRAND,

                COLUMN\_PRODUCT\_DESCRIPTION,

                COLUMN\_PRODUCT\_PRICE

        };

        String selection = COLUMN\_PRODUCT\_ID + " = ?";

        String[] selectionArgs = {String.**valueOf**(productId)};

        Cursor cursor = db.**query**(

                TABLE\_NAME,

                projection,

                selection,

                selectionArgs,

                null,

                null,

                null

        );

        Product product = null;

        if (cursor != null && cursor.**moveToFirst**()) {

            String name = cursor.**getString**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_NAME));

            String brand = cursor.**getString**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_BRAND));

            String description = cursor.**getString**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_DESCRIPTION));

            double price = cursor.**getDouble**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_PRICE));

            product = new **Product**(productId, name, brand, description, price);

            cursor.**close**();

        }

        db.**close**();

        return product;

    }

    public boolean **deleteProductById**(int productId) {

        SQLiteDatabase db = this.**getWritableDatabase**();

        String whereClause = COLUMN\_PRODUCT\_ID + " = ?";

        String[] whereArgs = {String.**valueOf**(productId)};

        int rowsDeleted = db.**delete**(TABLE\_NAME, whereClause, whereArgs);

        // Close the database

        db.**close**();

        // Check if any rows were deleted (deletion successful)

        return rowsDeleted > 0;

    }

    public boolean **updateProductPrice**(int productId, double newPrice) {

        SQLiteDatabase db = this.**getWritableDatabase**();

        ContentValues values = new **ContentValues**();

        values.**put**(COLUMN\_PRODUCT\_PRICE, newPrice);

        String whereClause = COLUMN\_PRODUCT\_ID + " = ?";

        String[] whereArgs = {String.**valueOf**(productId)};

        int rowsUpdated = db.**update**(TABLE\_NAME, values, whereClause, whereArgs);

        // Close the database

        db.**close**();

        // Check if any rows were updated (update successful)

        return rowsUpdated > 0;

    }

    public List<String> **getBrandList**() {

        List<String> brandList = new **ArrayList**<>();

        SQLiteDatabase db = this.**getReadableDatabase**();

        // SQL query to select distinct brands

        String query = "SELECT DISTINCT " + COLUMN\_PRODUCT\_BRAND + " FROM " + TABLE\_NAME;

        Cursor cursor = db.**rawQuery**(query, null);

        if (cursor != null && cursor.**moveToFirst**()) {

            do {

                String brand = cursor.**getString**(0); // 0 is the column index for brand

                brandList.**add**(brand);

                Log.**d**("DatabaseHelper", "Brand: " + brand); // Add this line for debugging

            } while (cursor.**moveToNext**());

            cursor.**close**();

        }

        db.**close**();

        return brandList;

    }

    // Add a method to select all items in a brand

    public List<Product> **getProductsByBrand**(String brandName) {

        List<Product> productList = new **ArrayList**<>();

        SQLiteDatabase db = this.**getReadableDatabase**();

        String[] projection = {

                COLUMN\_PRODUCT\_ID,

                COLUMN\_PRODUCT\_NAME,

                COLUMN\_PRODUCT\_PRICE

        };

        String selection = COLUMN\_PRODUCT\_BRAND + " = ?";

        String[] selectionArgs = {brandName};

        Cursor cursor = db.**query**(

                TABLE\_NAME,

                projection,

                selection,

                selectionArgs,

                null,

                null,

                null

        );

        if (cursor != null && cursor.**moveToFirst**()) {

            do {

                int productId = cursor.**getInt**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_ID));

                String name = cursor.**getString**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_NAME));

                double price = cursor.**getDouble**(cursor.**getColumnIndexOrThrow**(COLUMN\_PRODUCT\_PRICE));

                Product product = new **Product**(productId, name, brandName, null, price);

                productList.**add**(product);

            } while (cursor.**moveToNext**());

            cursor.**close**();

        }

        db.**close**();

        return productList;

    }

}

* DatabaseContract.java

// DatabaseContract.java

package com.example.products;

import android.provider.BaseColumns;

public final class DatabaseContract {

    private **DatabaseContract**() {}

    public static class ProductEntry implements BaseColumns {

        public static final String TABLE\_NAME = "products";

        public static final String COLUMN\_PRODUCT\_ID = "product\_id";

        public static final String COLUMN\_PRODUCT\_NAME = "product\_name";

        public static final String COLUMN\_PRODUCT\_BRAND = "product\_brand";

        public static final String COLUMN\_PRODUCT\_DESCRIPTION = "product\_description";

        public static final String COLUMN\_PRODUCT\_PRICE = "product\_price";

    }

}

* Product.java

// Product.java

package com.example.products;

public class Product {

    private int id;

    private String name;

    private String brand;

    private String description;

    private double price;

    public **Product**(int id, String name, String brand, String description, double price) {

        this.id = id;

        this.name = name;

        this.brand = brand;

        this.description = description;

        this.price = price;

    }

    public int **getId**() {

        return id;

    }

    public String **getName**() {

        return name;

    }

    public String **getBrand**() {

        return brand;

    }

    public String **getDescription**() {

        return description;

    }

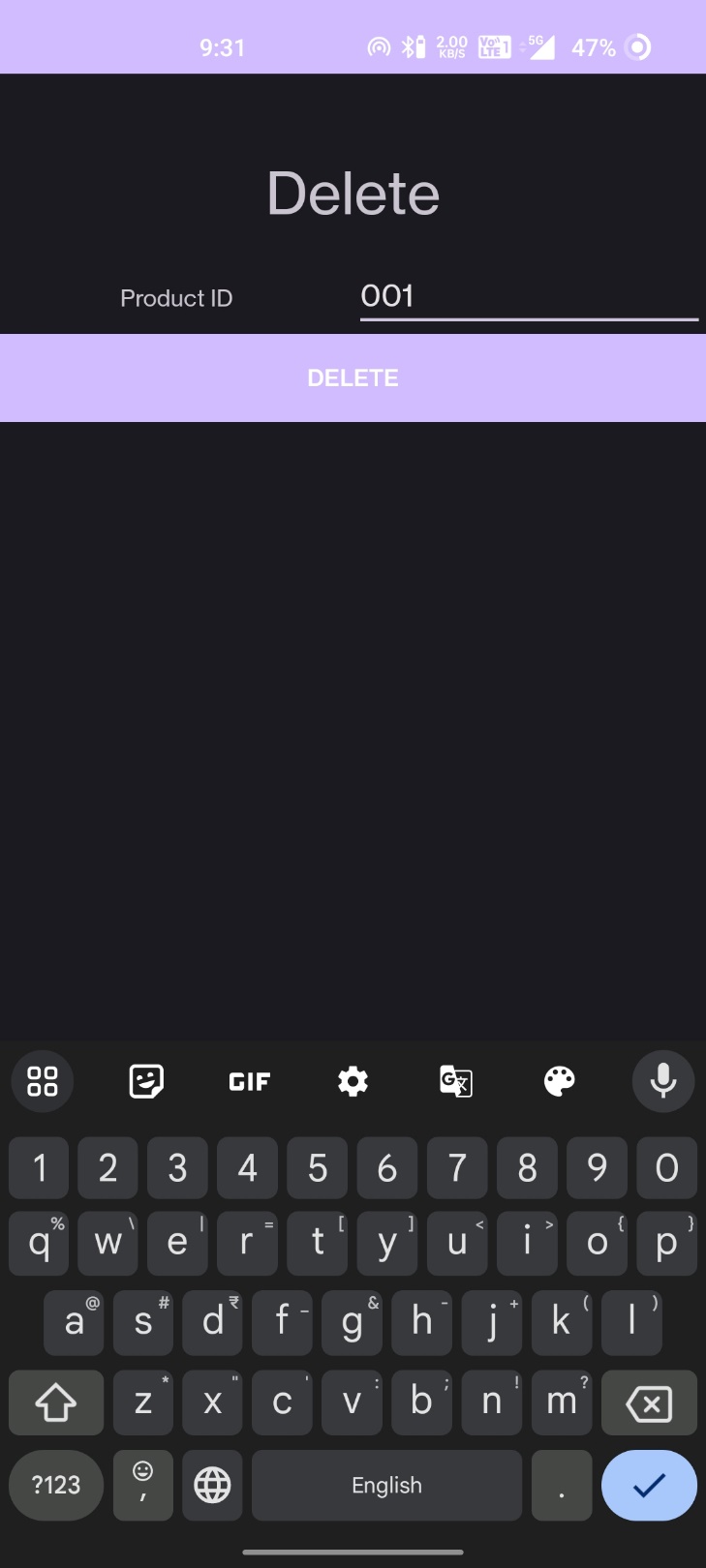
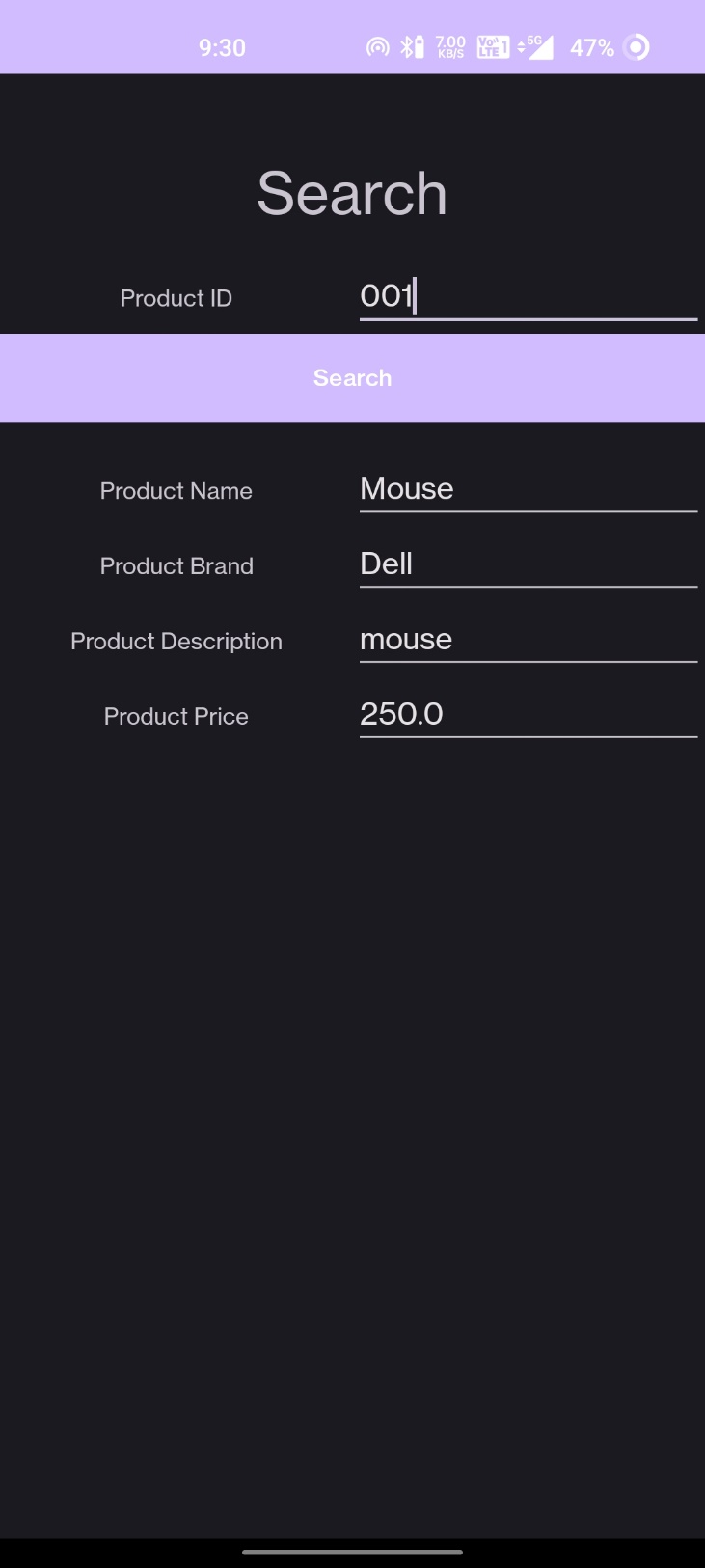
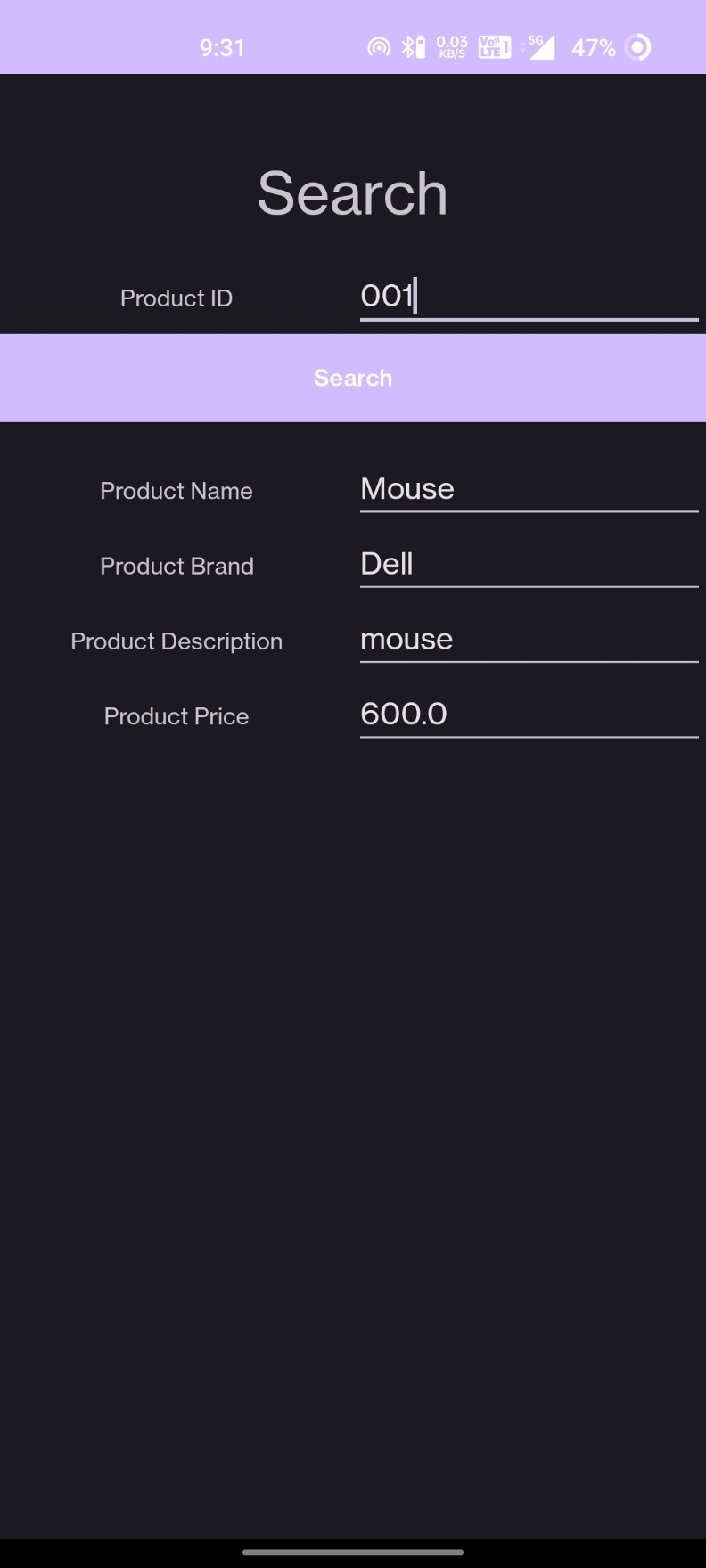
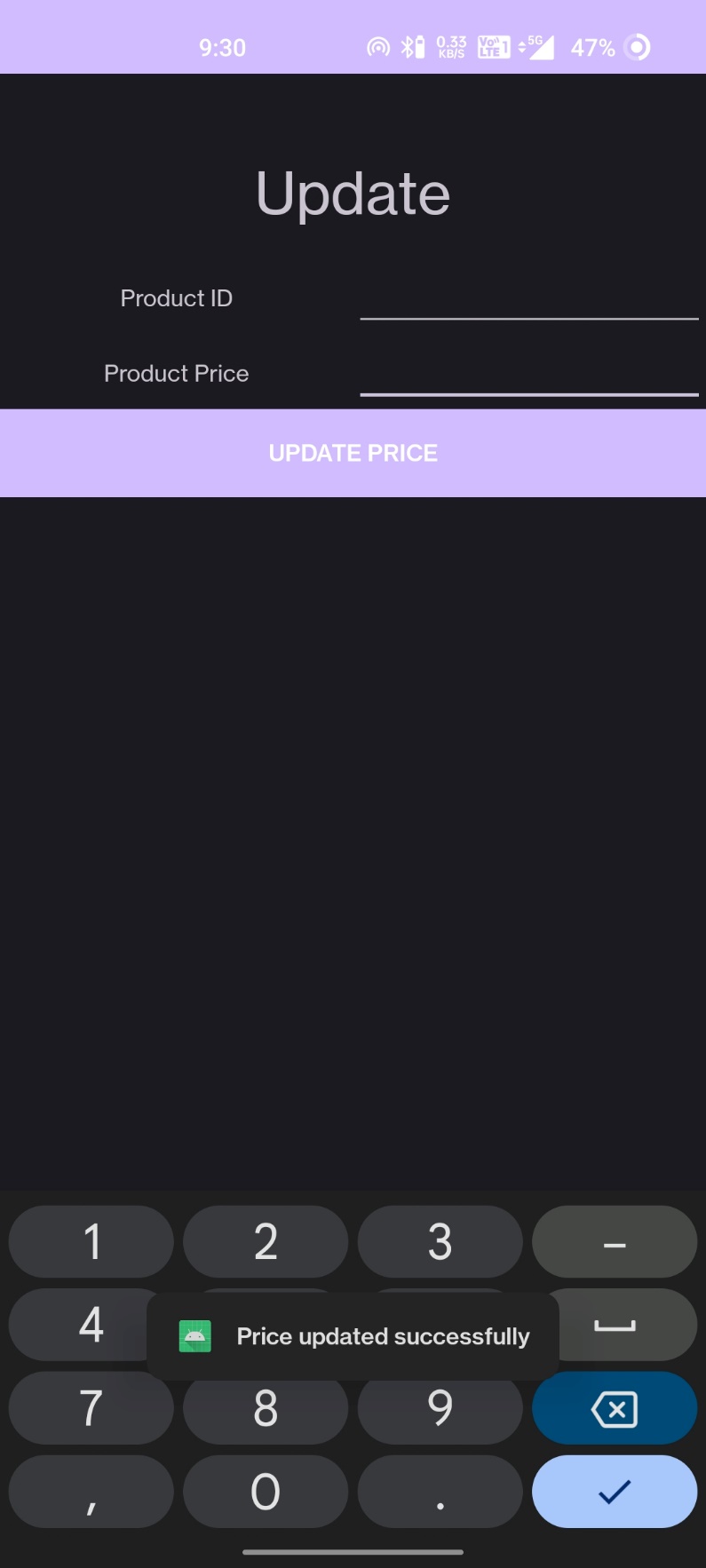
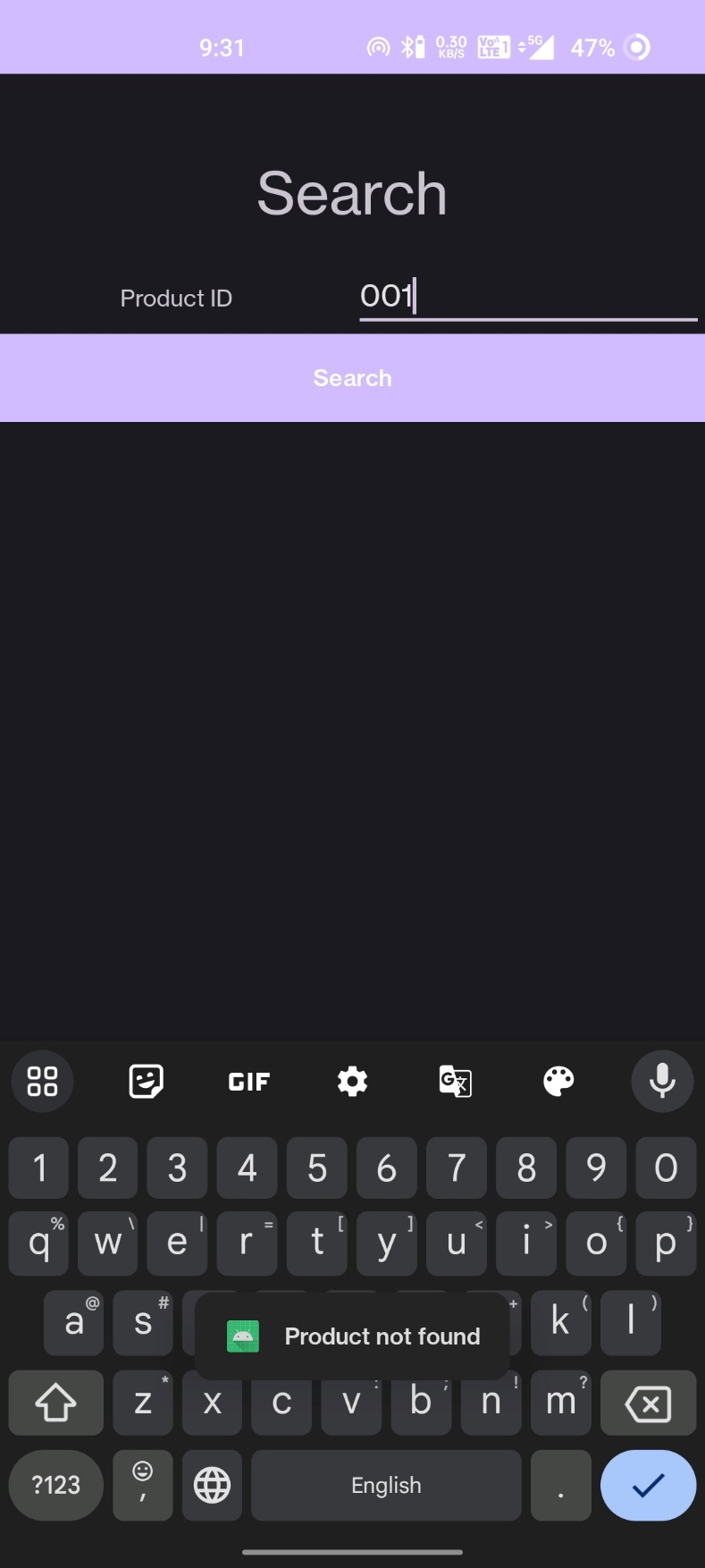
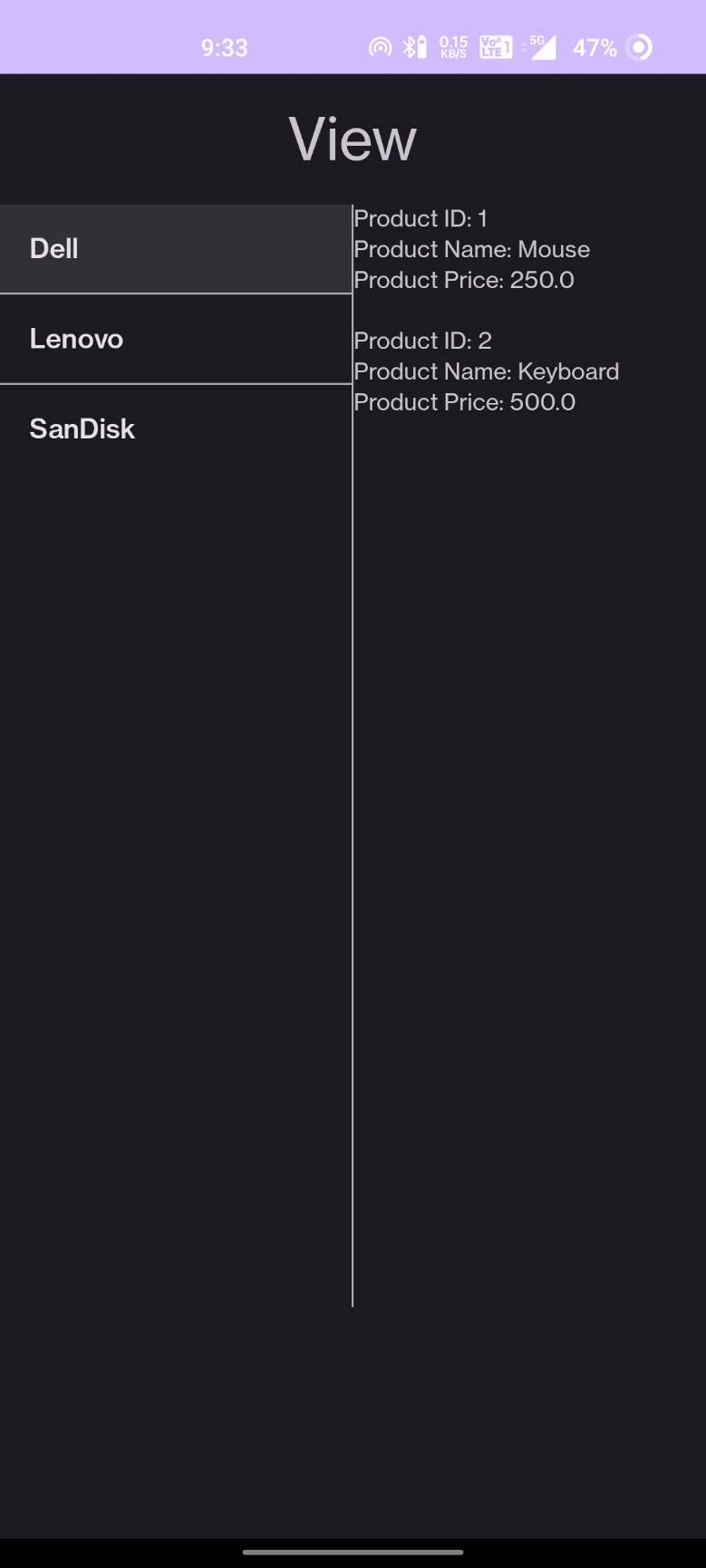
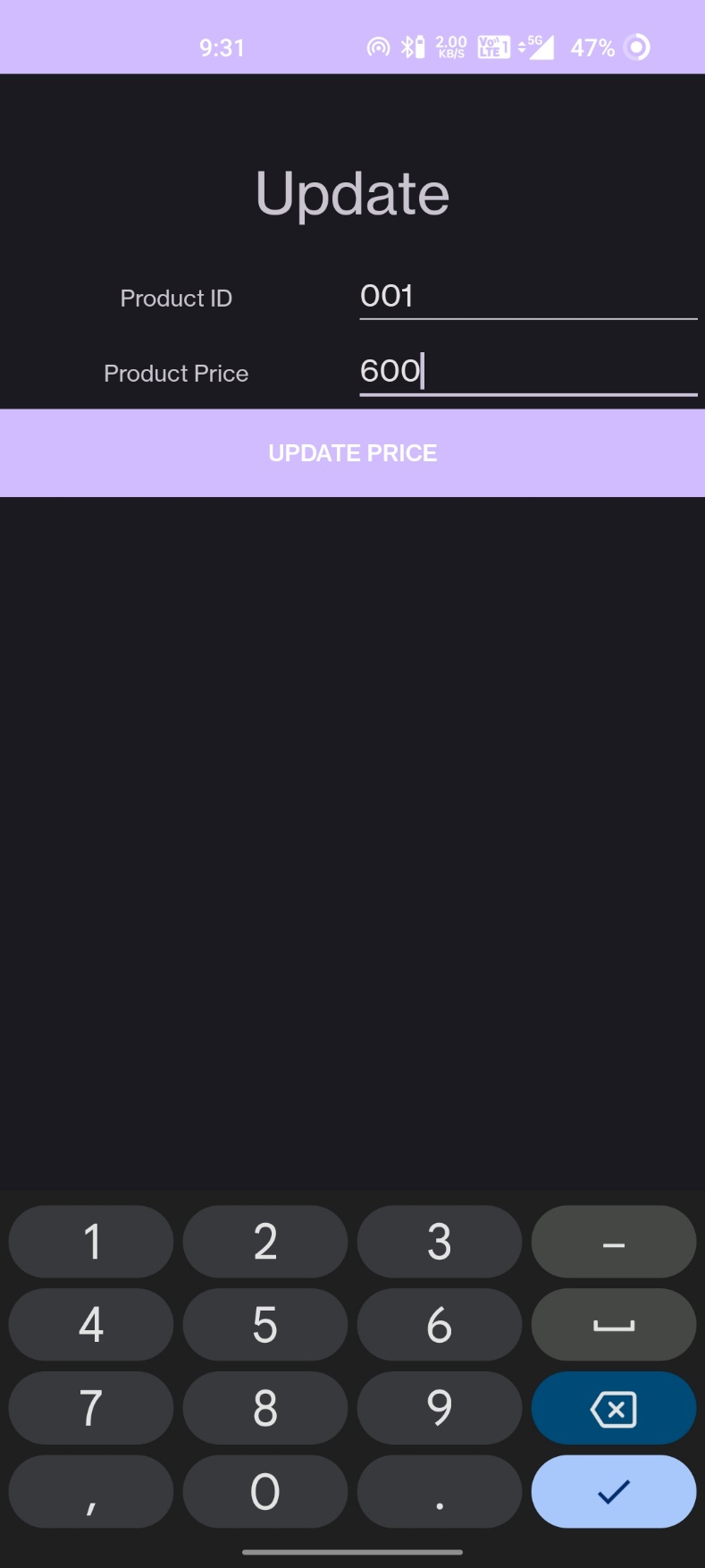
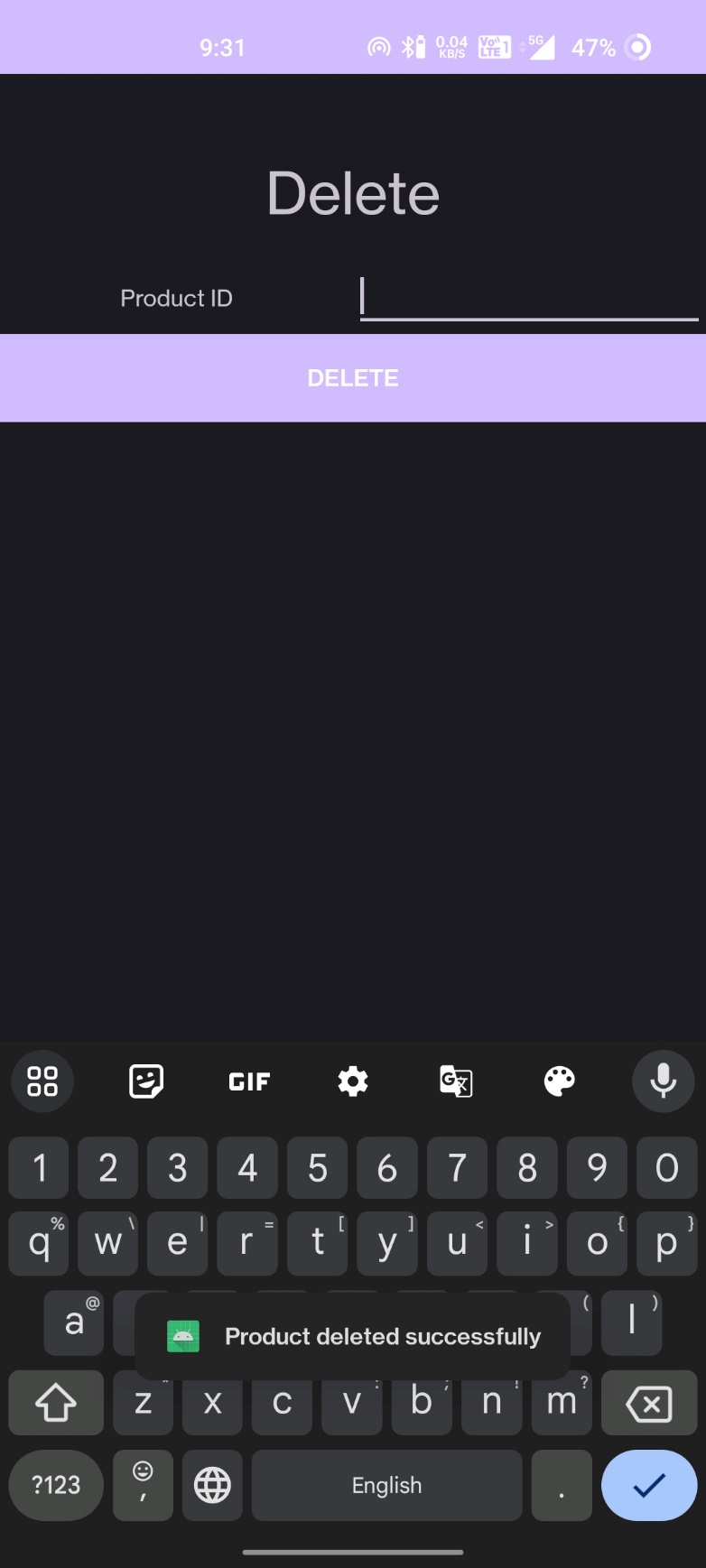
    public double **getPrice**() {

        return price;

    }

}

**Output:**

**Result:**

The mobile application was completed successfully

**Best Practices:**

* Using SQLite for local data storage.
* Input validation to ensure data integrity.
* Separation of concerns by using different activities for various operations.
* Reusable database helper class.

.

**Learning Outcomes:**

* Creating and managing Android activities.
* Working with SQLite databases in Android.
* Implementing input validation in Android applications.
* Navigating between different activities.
* Designing user interfaces with XML layout files.
* Handling user input and performing database operations.