

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**  
**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP THỰC HÀNH SỐ 4**  
**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**  
**NỘI DUNG BỔ SUNG: ỨNG DỤNG VỚI CSDL**

<b>STT</b>	<b>Mã sinh viên</b>	<b>Họ và tên</b>	<b>Lớp</b>
1	2251061762	Vũ Xuân Duy	64CNTT2

**Hà Nội, năm 2025**

## BÀI TẬP 1: SHARED PREFERENCE

### Mục tiêu:

- Hiểu cách sử dụng Shared Preference để lưu trữ dữ liệu cục bộ trong ứng dụng Android.
- Thực hành lưu trữ và đọc dữ liệu từ Shared Preference.

### Yêu cầu:

#### 1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho tên người dùng và mật khẩu, và ba nút bấm: "Lưu", "Xóa", và "Hiển thị".

#### 2. Sử dụng Shared Preference:

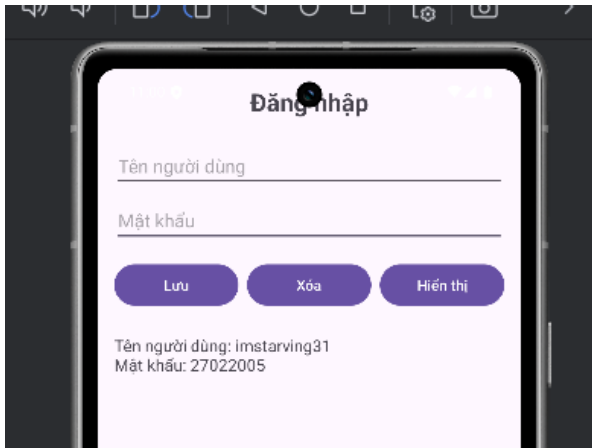
- Tạo một lớp helper **PreferenceHelper** để quản lý Shared Preference.
- Khi người dùng nhấn nút "Lưu", lưu tên người dùng và mật khẩu vào Shared Preference.
- Khi người dùng nhấn nút "Xóa", xóa dữ liệu đã lưu trong Shared Preference.
- Khi người dùng nhấn nút "Hiển thị", đọc dữ liệu từ Shared Preference và hiển thị lên màn hình.

#### 3. Thực hành:

- Viết mã Kotlin để thực hiện các chức năng trên.
- Sử dụng `getSharedPreferences` để truy cập Shared Preference và `edit()` để lưu dữ liệu.
- Sử dụng `commit()` hoặc `apply()` để lưu thay đổi.

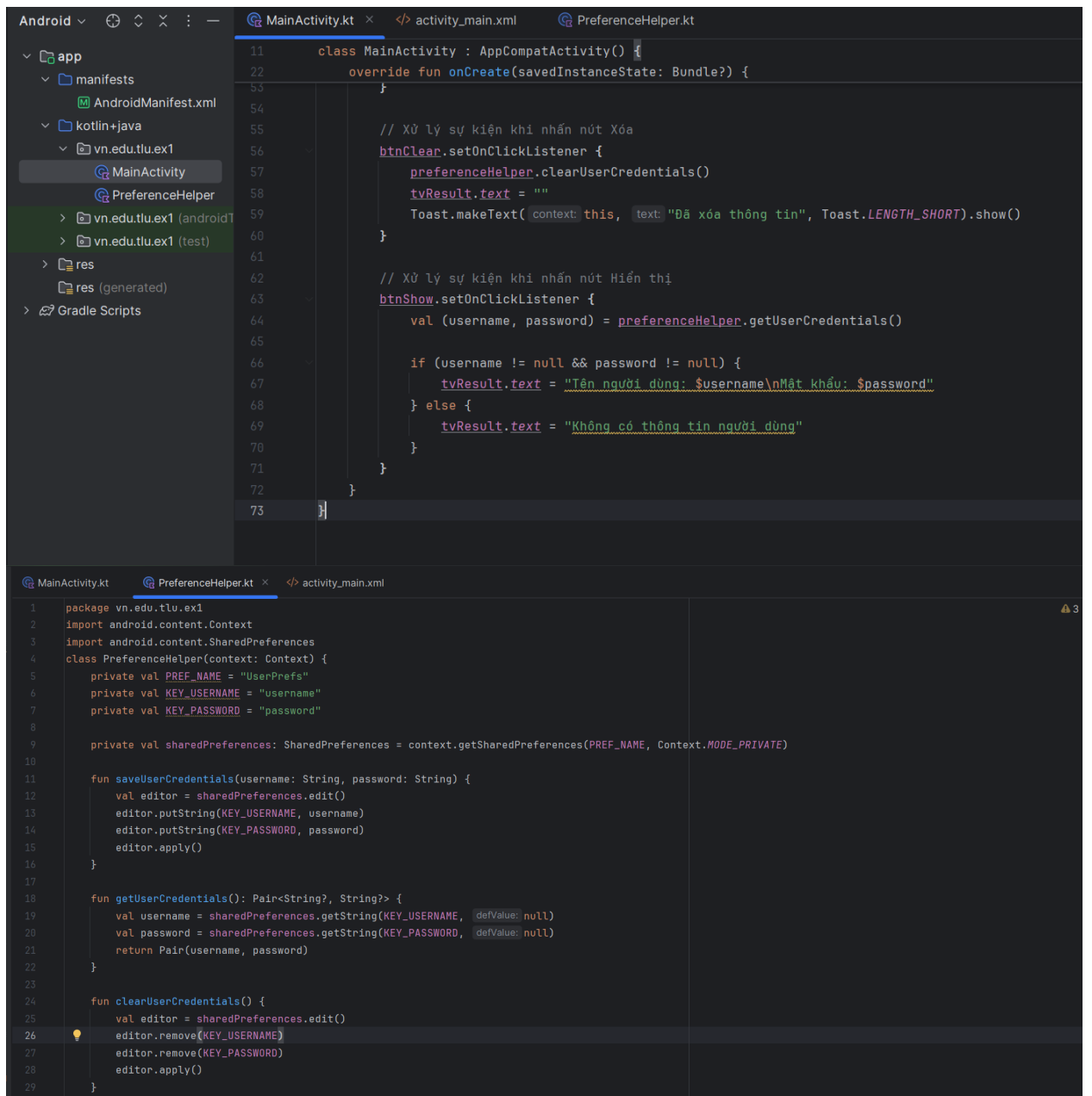
## 4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>



```
Android v
└─ app
  └─ manifests
    └─ AndroidManifest.xml
  └─ kotlin+java
    └─ vn.edu.tlu.ex1
      └─ MainActivity
      └─ PreferenceHelper
        └─ vn.edu.tlu.ex1 (android)
        └─ vn.edu.tlu.ex1 (test)
          └─ res
            └─ res (generated)
          └─ Gradle Scripts

MainActivity.kt  < activity_main.xml  PreferenceHelper.kt
1 package vn.edu.tlu.ex1
2
3
4 import android.os.Bundle
5 import android.widget.Button
6 import android.widget.EditText
7 import android.widget.TextView
8 import android.widget.Toast
9 import androidx.appcompat.app.AppCompatActivity
10
11 class MainActivity : AppCompatActivity() {
12
13     private lateinit var etUsername: EditText
14     private lateinit var etPassword: EditText
15     private lateinit var btnSave: Button
16     private lateinit var btnClear: Button
17     private lateinit var btnShow: Button
18     private lateinit var tvResult: TextView
19
20     private lateinit var preferenceHelper: PreferenceHelper
21
22     override fun onCreate(savedInstanceState: Bundle?) {
23         super.onCreate(savedInstanceState)
24         setContentView(R.layout.activity_main)
25
26         // Khởi tạo views
27         etUsername = findViewById(R.id.etUsername)
28         etPassword = findViewById(R.id.etPassword)
29
30         btnSave = findViewById(R.id.btnSave)
31         btnClear = findViewById(R.id.btnClear)
32         btnShow = findViewById(R.id.btnShow)
33         tvResult = findViewById(R.id.tvResult)
34
35         // Khởi tạo PreferenceHelper
36         preferenceHelper = PreferenceHelper(context = this)
37
38         // Xử lý sự kiện khi nhấn nút Lưu
39         btnSave.setOnClickListener {
40             val username = etUsername.text.toString()
41             val password = etPassword.text.toString()
42
43             if (username.isEmpty() || password.isEmpty()) {
44                 Toast.makeText(context, this, "Vui lòng nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show()
45                 return@setOnClickListener
46             }
47
48             preferenceHelper.saveUserCredentials(username, password)
49             Toast.makeText(context, this, "Đã lưu thông tin thành công", Toast.LENGTH_SHORT).show()
50
51             // Xóa trường nhập sau khi lưu
52             etUsername.setText("")
53             etPassword.setText("")
54         }
55     }
56 }
```



```
MainActivity.kt × PreferenceHelper.kt </> activity_main.xml ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:padding="16dp"
8     tools:context=".MainActivity">
9
10    <TextView
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="Đăng nhập"
14        android:textSize="24sp"
15        android:textStyle="bold"
16        android:gravity="center"
17        android:layout_marginBottom="24dp"/>
18
19    <EditText
20        android:id="@+id/etUsername"
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:hint="Tên người dùng"
24        android:inputType="text"
25        android:layout_marginBottom="8dp"/>
26
27    <EditText
28        android:id="@+id/etPassword"
29        android:layout_width="match_parent"
30        android:layout_height="wrap_content"
31        android:hint="Mật khẩu"
32        android:inputType="textPassword">
```

```
MainActivity.kt  PreferenceHelper.kt  </> activity_main.xml ×
2      <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
27      <EditText
33          android:layout_marginBottom="16dp"/>
34
35      <LinearLayout
36          android:layout_width="match_parent"
37          android:layout_height="wrap_content"
38          android:orientation="horizontal">
39
40          <Button
41              android:id="@+id/btnSave"
42              android:layout_width="0dp"
43              android:layout_height="wrap_content"
44              android:layout_weight="1"
45              android:text="Lưu"
46              android:layout_marginEnd="4dp"/>
47
48          <Button
49              android:id="@+id/btnClear"
50              android:layout_width="0dp"
51              android:layout_height="wrap_content"
52              android:layout_weight="1"
53              android:text="Xóa"
54              android:layout_marginStart="4dp"
55              android:layout_marginEnd="4dp"/>
56
57          <Button
58              android:id="@+id/btnShow"
59              android:layout_width="0dp"
60              android:layout_height="wrap_content"
61              android:layout_weight="1"
62              android:text="Hiển thị"
63              android:layout_marginStart="4dp"/>
64      </LinearLayout>
65
66      <TextView
67          android:id="@+id/tvResult"
68          android:layout_width="match_parent"
69          android:layout_height="wrap_content"
70          android:layout_marginTop="24dp"
71          android:textSize="16sp"/>
72
73  </LinearLayout>
```

## BÀI TẬP 2: SQLite

### Mục tiêu:

- Hiểu cách sử dụng SQLite để lưu trữ dữ liệu trong ứng dụng Android.
- Thực hành tạo cơ sở dữ liệu SQLite, thêm, sửa, xóa dữ liệu.

### Yêu cầu:

#### 1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho tên và số điện thoại, và bốn nút bấm: "Thêm", "Sửa", "Xóa", và "Hiển thị".

#### 2. Sử dụng SQLite:

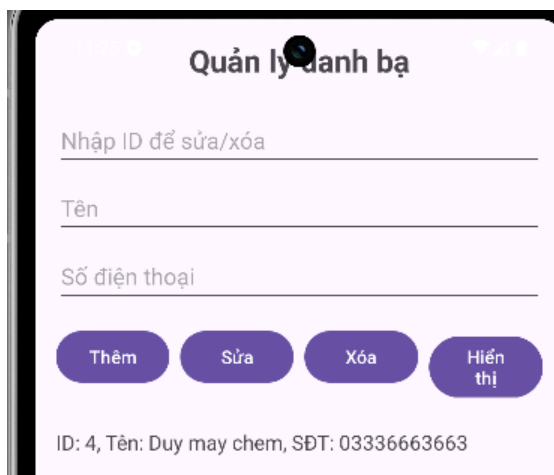
- Tạo một lớp helper để quản lý cơ sở dữ liệu SQLite.
- Tạo bảng dữ liệu với hai cột: tên và số điện thoại.
- Viết các hàm để thêm, sửa, xóa dữ liệu từ cơ sở dữ liệu.
- Khi người dùng nhấn nút "Hiển thị", đọc dữ liệu từ cơ sở dữ liệu và hiển thị lên màn hình.

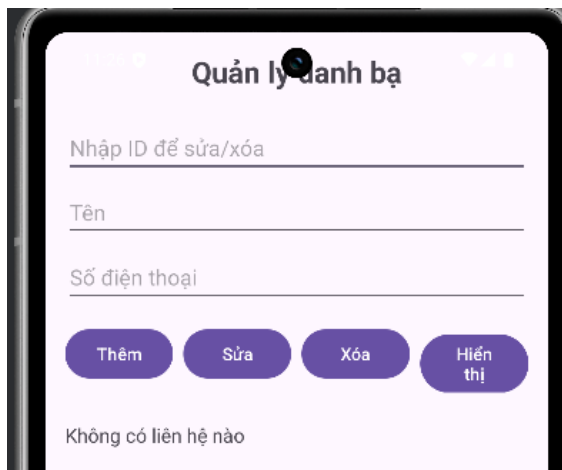
#### 3. Thực hành:

- Viết mã Kotlin để thực hiện các chức năng trên.
- Sử dụng SQLiteOpenHelper để tạo và quản lý cơ sở dữ liệu.

#### 4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>





```

<> activity_main.xml  MainActivity.kt  DatabaseHelper.kt
1      package vn.edu.tlu.ex2
2
3      > import ...
4
5
6
7
8
9
10     class MainActivity : AppCompatActivity() {
11
12         private lateinit var etId: EditText
13         private lateinit var etName: EditText
14         private lateinit var etPhone: EditText
15         private lateinit var btnAdd: Button
16         private lateinit var btnUpdate: Button
17         private lateinit var btnDelete: Button
18         private lateinit var btnShow: Button
19         private lateinit var tvResult: TextView
20
21         private lateinit var dbHelper: DatabaseHelper
22
23         override fun onCreate(savedInstanceState: Bundle?) {
24             super.onCreate(savedInstanceState)
25             setContentView(R.layout.activity_main)
26
27             // Khởi tạo views
28             etId = findViewById(R.id.etId)
29             etName = findViewById(R.id.etName)
30             etPhone = findViewById(R.id.etPhone)
31             btnAdd = findViewById(R.id.btnAdd)
32             btnUpdate = findViewById(R.id.btnUpdate)
33             btnDelete = findViewById(R.id.btnDelete)
34             btnShow = findViewById(R.id.btnShow)
35             tvResult = findViewById(R.id.tvResult)
36

```



```

<? activity_main.xml      MainActivity.kt  DatabaseHelper.kt

10  class MainActivity : AppCompatActivity() {
23      override fun onCreate(savedInstanceState: Bundle?) {
38          dbHelper = DatabaseHelper(context = this)
39
40          // Xử lý nút Thêm
41          btnAdd.setOnClickListener {
42              val name = etName.text.toString().trim()
43              val phone = etPhone.text.toString().trim()
44
45              if (name.isEmpty() || phone.isEmpty()) {
46                  Toast.makeText(context = this, text = "Vui lòng nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show()
47                  return@setOnClickListener
48              }
49
50              val id = dbHelper.addContact(name, phone)
51              if (id > 0) {
52                  Toast.makeText(context = this, text = "Đã thêm liên hệ", Toast.LENGTH_SHORT).show()
53                  clearInputs()
54              }
55          }
56
57          // Xử lý nút Sửa
58          btnUpdate.setOnClickListener {
59              val idText = etId.text.toString().trim()
60              val name = etName.text.toString().trim()
61              val phone = etPhone.text.toString().trim()
62
63              if (idText.isEmpty() || name.isEmpty() || phone.isEmpty()) {
64                  Toast.makeText(context = this, text = "Vui lòng nhập ID và đầy đủ thông tin", Toast.LENGTH_SHORT).show()
65                  return@setOnClickListener
66              }
67          }

```

```

Project Alt+1  .xml      MainActivity.kt  DatabaseHelper.kt

10  class MainActivity : AppCompatActivity() {
23      override fun onCreate(savedInstanceState: Bundle?) {
58          btnUpdate.setOnClickListener {
66              }
67
68              val id = idText.toLongOrNull()
69              if (id == null) {
70                  Toast.makeText(context = this, text = "ID không hợp lệ", Toast.LENGTH_SHORT).show()
71                  return@setOnClickListener
72              }
73
74              val rowsAffected = dbHelper.updateContact(id, name, phone)
75              if (rowsAffected > 0) {
76                  Toast.makeText(context = this, text = "Đã cập nhật liên hệ", Toast.LENGTH_SHORT).show()
77                  clearInputs()
78              } else {
79                  Toast.makeText(context = this, text = "Không tìm thấy liên hệ với ID: $id", Toast.LENGTH_SHORT).show()
80              }
81          }
82
83          // Xử lý nút Xóa
84          btnDelete.setOnClickListener {
85              val idText = etId.text.toString().trim()
86
87              if (idText.isEmpty()) {
88                  Toast.makeText(context = this, text = "Vui lòng nhập ID để xóa", Toast.LENGTH_SHORT).show()
89                  return@setOnClickListener
90              }
91
92              val id = idText.toLongOrNull()
93              if (id == null) {
94                  Toast.makeText(context = this, text = "ID không hợp lệ", Toast.LENGTH_SHORT).show()

```

```

94         btnDelete.setOnClickListener {
95             Toast.makeText(context: this, text: "ID không hợp lệ", Toast.LENGTH_SHORT).show()
96             return@setOnClickListener
97         }
98
99         val rowsAffected = dbHelper.deleteContact(id)
100         if (rowsAffected > 0) {
101             Toast.makeText(context: this, text: "Đã xóa liên hệ", Toast.LENGTH_SHORT).show()
102             clearInputs()
103         } else {
104             Toast.makeText(context: this, text: "Không tìm thấy liên hệ với ID: $id", Toast.LENGTH_SHORT).show()
105         }
106     }
107
108     // Xử lý nút Hiển thị
109     btnShow.setOnClickListener {
110         val contacts = dbHelper.getAllContacts()
111         if (contacts.isEmpty()) {
112             tvResult.text = "Không có liên hệ nào"
113         } else {
114             tvResult.text = contacts.joinToString(separator: "\n")
115         }
116     }
117
118     private fun clearInputs() {
119         etId.setText("")
120         etName.setText("")
121         etPhone.setText("")
122     }

```

```
activity_main.xml x MainActivity.kt DatabaseHelper.kt
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     android:padding="16dp">
7
8     <TextView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:text="Quản lý danh bạ"
12        android:textSize="24sp"
13        android:textStyle="bold"
14        android:gravity="center"
15        android:layout_marginBottom="24dp"/>
16
17    <EditText
18        android:id="@+id/etId"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:hint="Nhập ID để sửa/xóa"
22        android:inputType="number"
23        android:layout_marginBottom="8dp"/>
24
25    <EditText
26        android:id="@+id/etName"
27        android:layout_width="match_parent"
28        android:layout_height="wrap_content"
29        android:hint="Tên"
30        android:inputType="text"
31        android:layout_marginBottom="8dp"/>
32
```

```

2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
32
33 <EditText
34     android:id="@+id/etPhone"
35     android:layout_width="match_parent"
36     android:layout_height="wrap_content"
37     android:hint="Số điện thoại"
38     android:inputType="phone"
39     android:layout_marginBottom="16dp"/>
40
41 <LinearLayout
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:orientation="horizontal">
45
46 <Button
47     android:id="@+id/btnAdd"
48     android:layout_width="0dp"
49     android:layout_height="wrap_content"
50     android:layout_weight="1"
51     android:text="Thêm"
52     android:layout_marginEnd="4dp"/>
53
54 <Button
55     android:id="@+id/btnUpdate"
56     android:layout_width="0dp"
57     android:layout_height="wrap_content"
58     android:layout_weight="1"
59     android:text="Sửa"
60     android:layout_marginStart="4dp"
61     android:layout_marginEnd="4dp"/>
62

```

```
activity_main.xml x MainActivity.kt DatabaseHelper.kt
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
Resource Manager Layout
54 <Button
55     android:text="Sửa"
56     android:layout_marginStart="4dp"
57     android:layout_marginEnd="4dp"/>
58
59
60
61
62
63 <Button
64     android:id="@+id/btnDelete"
65     android:layout_width="0dp"
66     android:layout_height="wrap_content"
67     android:layout_weight="1"
68     android:text="Xóa"
69     android:layout_marginStart="4dp"
70     android:layout_marginEnd="4dp"/>
71
72
73 <Button
74     android:id="@+id/btnShow"
75     android:layout_width="0dp"
76     android:layout_height="wrap_content"
77     android:layout_weight="1"
78     android:text="Hiển thị"
79     android:layout_marginStart="4dp"/>
80
81 </LinearLayout>
82
83 <TextView
84     android:id="@+id/tvResult"
85     android:layout_width="match_parent"
86     android:layout_height="wrap_content"
87     android:layout_marginTop="24dp"
88     android:textSize="16sp"/>
89 </LinearLayout>
```

```

1 package vn.edu.tlu.ex2
2
3 import android.content.ContentValues
4 import android.content.Context
5 import android.database.sqlite.SQLiteDatabase
6 import android.database.sqlite.SQLiteOpenHelper
7
8 class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, factory: null, DATABASE_VERSION) {
9
10     companion object {
11         private const val DATABASE_NAME = "ContactDB"
12         private const val DATABASE_VERSION = 1
13         private const val TABLE_NAME = "contacts"
14         private const val KEY_ID = "id"
15         private const val KEY_NAME = "name"
16         private const val KEY_PHONE = "phone"
17     }
18
19     override fun onCreate(db: SQLiteDatabase?) {
20         val createTable = """
21             CREATE TABLE $TABLE_NAME (
22                 $KEY_ID INTEGER PRIMARY KEY AUTOINCREMENT,
23                 $KEY_NAME TEXT,
24                 $KEY_PHONE TEXT
25             )
26         """
27         db?.execSQL(createTable)
28     }
29
30     override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
31         db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
32         onCreate(db)
33     }
34 }

```

```

</> activity_main.xml    MainActivity.kt    DatabaseHelper.kt ×
8      class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, factory: null, DATABASE_VERSION) {
32          onCreate(db)
33      }
34
35      // Thêm liên hệ
36      fun addContact(name: String, phone: String): Long {
37          val db = this.writableDatabase
38          val values = ContentValues().apply {
39              put(KEY_NAME, name)
40              put(KEY_PHONE, phone)
41          }
42          val id = db.insert(TABLE_NAME, nullColumnHack: null, values)
43          db.close()
44          return id
45      }
46
47      // Sửa liên hệ (dựa trên ID)
48      fun updateContact(id: Long, name: String, phone: String): Int {
49          val db = this.writableDatabase
50          val values = ContentValues().apply {
51              put(KEY_NAME, name)
52              put(KEY_PHONE, phone)
53          }
54          val rowsAffected = db.update(TABLE_NAME, values, whereClause: "$KEY_ID = ?", arrayOf(id.toString()))
55          db.close()
56          return rowsAffected
57      }
58
59      // Xóa liên hệ (dựa trên ID)
60      fun deleteContact(id: Long): Int {
61          val db = this.writableDatabase
62          val rowsAffected = db.delete(TABLE_NAME, whereClause: "$KEY_ID = ?", arrayOf(id.toString()))

```

```

</> activity_main.xml    MainActivity.kt    DatabaseHelper.kt ×
8      class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, factory: null, DATABASE_VERSION) {
60          fun deleteContact(id: Long): Int {
62              val rowsAffected = db.delete(TABLE_NAME, whereClause: "$KEY_ID = ?", arrayOf(id.toString()))
63              db.close()
64              return rowsAffected
65          }
66
67          // Lấy tất cả liên hệ
68          fun getAllContacts(): List<String> {
69              val contactList = mutableListOf<String>()
70              val db = this.readableDatabase
71              val cursor = db.rawQuery("SELECT * FROM $TABLE_NAME", selectionArgs: null)
72
73              if (cursor.moveToFirst()) {
74                  do {
75                      val id = cursor.getLong(cursor.getColumnIndexOrThrow(KEY_ID))
76                      val name = cursor.getString(cursor.getColumnIndexOrThrow(KEY_NAME))
77                      val phone = cursor.getString(cursor.getColumnIndexOrThrow(KEY_PHONE))
78                      contactList.add("ID: $id, Tên: $name, SĐT: $phone")
79                  } while (cursor.moveToNext())
80              }
81
82              cursor.close()
83              db.close()
84              return contactList
85          }
86      }

```

## BÀI TẬP 3: HỆ SINH THÁI FIREBASE

### Mục tiêu:

- Hiểu rõ về các dịch vụ chính của Firebase.
- Biết cách tích hợp Firebase vào dự án phát triển ứng dụng.

### Yêu cầu:

#### 1. Tìm hiểu các dịch vụ chính của Firebase:

- Firebase Authentication: Xác thực người dùng.
- Firebase Realtime Database và Cloud Firestore: Cơ sở dữ liệu thời gian thực và NoSQL.
- Firebase Cloud Functions: Chạy mã backend serverless.
- Firebase Cloud Messaging (FCM): Gửi thông báo đẩy.
- Firebase Storage: Lưu trữ tệp tin trên đám mây.
- Firebase Machine Learning (ML): Tích hợp trí tuệ nhân tạo vào ứng dụng.

#### 2. Viết báo cáo:

- Giới thiệu tổng quan về Firebase và lịch sử phát triển.
- Mô tả chi tiết từng dịch vụ chính của Firebase.
- Thảo luận về lợi ích và ứng dụng của Firebase trong phát triển ứng dụng.

### Nội dung báo cáo viết ở đây

#### Tổng quan về Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Nó cung cấp một bộ công cụ và dịch vụ giúp các nhà phát triển xây dựng, quản lý và mở rộng ứng dụng một cách dễ dàng. Firebase được thiết kế để hỗ trợ cả ứng dụng Android, iOS và web, với các tính năng như cơ sở dữ liệu thời gian thực, xác thực người dùng, thông báo đẩy, lưu trữ đám mây, và nhiều hơn nữa. Firebase hoạt động theo mô hình **Backend-as-a-Service (BaaS)**, giúp giảm bớt gánh nặng quản lý backend cho các nhà phát triển.



## Lịch sử phát triển

**2011:** Firebase được thành lập bởi James Tamplin và Andrew Lee dưới dạng một công ty khởi nghiệp, ban đầu tập trung vào việc xây dựng một cơ sở dữ liệu thời gian thực (Realtime Database) để đồng bộ dữ liệu giữa các thiết bị.

**2014:** Google mua lại Firebase và tích hợp nó vào hệ sinh thái của mình, mở rộng các dịch vụ và cải thiện khả năng tích hợp với các sản phẩm khác của Google.

**2016:** Tại Google I/O 2016, Firebase được giới thiệu lại với nhiều dịch vụ mới như Cloud Messaging, Analytics, và Authentication, trở thành một nền tảng toàn diện cho phát triển ứng dụng.

**2017-2023:** Firebase tiếp tục phát triển với việc ra mắt Cloud Firestore (một cơ sở dữ liệu NoSQL mạnh mẽ hơn Realtime Database), Firebase Machine Learning, và các cải tiến liên tục cho các dịch vụ hiện có. Đến nay, Firebase đã trở thành một trong những nền tảng phổ biến nhất cho các nhà phát triển ứng dụng.

## Mô tả chi tiết từng dịch vụ chính của Firebase

### Firebase Authentication

**Chức năng:** Cung cấp các phương thức xác thực người dùng an toàn và dễ sử dụng.

**Tính năng:**

Hỗ trợ nhiều phương thức đăng nhập: email/mật khẩu, số điện thoại, tài khoản Google, Facebook, Twitter, GitHub, v.v.

Quản lý phiên đăng nhập, xác minh email, khôi phục mật khẩu.

Tích hợp dễ dàng với các dịch vụ khác của Firebase.

**Ứng dụng:** Đăng nhập/đăng ký người dùng trong ứng dụng, bảo mật dữ liệu dựa trên quyền truy cập của người dùng.

### Firebase Realtime Database

**Chức năng:** Cơ sở dữ liệu NoSQL thời gian thực, lưu trữ dữ liệu dưới dạng JSON.

**Tính năng:**

Đồng bộ dữ liệu theo thời gian thực giữa các thiết bị.

Hỗ trợ làm việc offline (dữ liệu được đồng bộ khi có kết nối).

Quy tắc bảo mật (Security Rules) để kiểm soát quyền truy cập.

**Ứng dụng:** Chat thời gian thực, ứng dụng cộng tác (như Google Docs), trò chơi đa người chơi.

### Cloud Firestore

**Chức năng:** Một cơ sở dữ liệu NoSQL hiện đại hơn Realtime Database, cũng được lưu trữ trên đám mây.

**Tính năng:**

Hỗ trợ truy vấn phức tạp hơn, khả năng mở rộng tốt hơn.

Đồng bộ thời gian thực và làm việc offline.

Tích hợp với các dịch vụ khác của Firebase.

**Ứng dụng:** Ứng dụng cần lưu trữ dữ liệu có cấu trúc phức tạp, như danh sách sản phẩm, bài viết, hoặc hồ sơ người dùng.

Firebase Cloud Functions

**Chức năng:** Chạy mã backend serverless trên đám mây.

**Tính năng:**

Tự động mở rộng theo nhu cầu, không cần quản lý server.

Kích hoạt bởi các sự kiện từ Firebase (ví dụ: khi người dùng đăng ký, khi dữ liệu thay đổi).

Viết bằng JavaScript/TypeScript (Node.js).

**Ứng dụng:** Xử lý logic phức tạp (như gửi email chào mừng), tích hợp với API bên thứ ba, hoặc tự động hóa tác vụ.

Firebase Cloud Messaging (FCM)

**Chức năng:** Gửi thông báo đẩy (push notifications) đến người dùng.

**Tính năng:**

Gửi thông báo đến một thiết bị, nhóm thiết bị, hoặc tất cả người dùng.

Hỗ trợ cả Android, iOS và web.

Tích hợp với Analytics để phân tích hiệu quả thông báo.

**Ứng dụng:** Gửi thông báo khuyến mãi, cập nhật trạng thái đơn hàng, hoặc tin nhắn trong ứng dụng chat.

Firebase Storage

**Chức năng:** Lưu trữ và quản lý tệp tin (hình ảnh, video, tài liệu) trên đám mây.

**Tính năng:**

Tích hợp với Authentication để kiểm soát quyền truy cập.

Hỗ trợ tải lên/tải xuống tệp với hiệu suất cao.

Tự động mở rộng dung lượng lưu trữ.

**Ứng dụng:** Lưu trữ ảnh đại diện người dùng, video, hoặc tài liệu trong ứng dụng.

Firestore Machine Learning (ML)

**Chức năng:** Tích hợp trí tuệ nhân tạo và máy học vào ứng dụng.

**Tính năng:**

Cung cấp các mô hình ML sẵn có (như nhận diện văn bản, phân loại hình ảnh).

Hỗ trợ triển khai mô hình ML tùy chỉnh (Custom Models) với TensorFlow Lite.

Tích hợp với các dịch vụ khác của Firebase.

**Ứng dụng:** Nhận diện khuôn mặt, dịch ngôn ngữ, hoặc phân tích cảm xúc trong ứng dụng.

## **Lợi ích và ứng dụng của Firebase trong phát triển ứng dụng**

Lợi ích

**Dễ sử dụng:** Firebase cung cấp SDK và tài liệu chi tiết, giúp nhà phát triển tích hợp nhanh chóng.

**Tích hợp toàn diện:** Các dịch vụ của Firebase hoạt động liền mạch với nhau (ví dụ: Authentication kết hợp với Firestore để bảo mật dữ liệu).

**Miễn phí ở mức cơ bản:** Firebase có gói miễn phí (Spark Plan) phù hợp cho các dự án nhỏ hoặc thử nghiệm.

**Khả năng mở rộng:** Tự động mở rộng theo nhu cầu, phù hợp cho cả ứng dụng nhỏ và lớn.

**Hỗ trợ đa nền tảng:** Hoạt động trên Android, iOS, web, và cả Unity (cho game).

Ứng dụng

**Ứng dụng chat:** Sử dụng Realtime Database hoặc Firestore để đồng bộ tin nhắn, FCM để gửi thông báo.

**Ứng dụng thương mại điện tử:** Firestore để lưu trữ sản phẩm, Storage để lưu ảnh sản phẩm, FCM để gửi thông báo khuyến mãi.

**Ứng dụng mạng xã hội:** Authentication để đăng nhập, Storage để lưu ảnh/video, Firestore để lưu bài viết.

**Trò chơi:** Realtime Database để đồng bộ trạng thái trò chơi, Analytics để phân tích hành vi người chơi.

### 3. Thực hành:

- Tạo một dự án Firebase mới trên Firebase Console.
- Đăng ký ứng dụng Android vào dự án Firebase.
- Sử dụng ít nhất hai dịch vụ của Firebase trong dự án (ví dụ: Authentication và Realtime Database).

### **Bài tập cụ thể: Tích hợp Firebase Authentication và Realtime Database**

**Yêu cầu:**

#### 1. Tạo ứng dụng mới:

- Tạo một dự án Android mới bằng Kotlin.
- Thiết kế giao diện người dùng với hai trường nhập (EditText) cho email và mật khẩu, và ba nút bấm: "Đăng ký", "Đăng nhập", và "Hiển thị dữ liệu".

#### 2. Tích hợp Firebase Authentication:

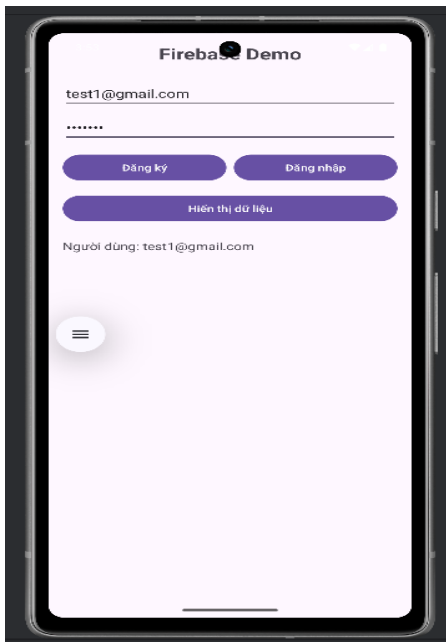
- Sử dụng Firebase Authentication để cho phép người dùng đăng ký và đăng nhập bằng email và mật khẩu.
- Viết mã để xử lý các sự kiện đăng ký và đăng nhập thành công hoặc thất bại.

#### 3. Tích hợp Firebase Realtime Database:

- Sau khi người dùng đăng nhập thành công, lưu trữ thông tin người dùng vào Firebase Realtime Database.
- Khi người dùng nhấn nút "Hiển thị dữ liệu", đọc dữ liệu từ Firebase Realtime Database và hiển thị lên màn hình.

#### 4. Kết quả

<<Sinh viên chụp Ảnh màn hình kết quả và mã nguồn chính tại đây>>



FirestoreDemo ▾

## Realtime Database

[Need help with Realtime Database? Ask Gemini](#)

[Data](#) | [Rules](#) | [Backups](#) | [Use](#) | [Extensions](#)

Protect your Realtime Database resources from misuse such as billing fraud and ph

<https://fir-demo-2ed02-default-rtdb.firebaseio.com>

<https://fir-demo-2ed02-default-rtdb.firebaseio.com/>

- users
  - S91u00tkikWJPZtQPnowqlgfdUj1
    - email: "test1@gmail.com"
  - gnlgb3RbGNMNq0hTuhv1VcbN2qV2

FirestoreDemo ▾

## Authentication

[Users](#) | [Connection method](#) | [Models](#) | [Use](#) | [Settings](#) | [Extensions](#)

The following authentication features will no longer be available when Firebase Dynamic Links is deprecated on August 25, 2025: email-based link authentication for mobile apps, and Cordova OAuth support for web apps. ▾

[Add a user](#)

Identifier	Suppliers	Creation date ▾	Last login	User UID
test1@gmail.com		24 mars 2025	24 mars 2025	S91u00tkikWJPZtQPnowqlgfd...

Lines per page: 50 ▾ 1 - 2 of 2 < >

```
</> activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)

1  package vn.edu.tlu.ex3
2
3  import android.os.Bundle
4  import android.widget.Button
5  import android.widget.EditText
6  import android.widget.TextView
7  import android.widget.Toast
8  import androidx.appcompat.app.AppCompatActivity
9  import com.google.firebase.auth.FirebaseAuth
10 import com.google.firebase.database.FirebaseDatabase
11
12 </> class MainActivity : AppCompatActivity() {
13
14     private lateinit var etEmail: EditText
15     private lateinit var etPassword: EditText
16     private lateinit var btnSignUp: Button
17     private lateinit var btnSignIn: Button
18     private lateinit var btnShowData: Button // Thêm nút Hiển thị dữ liệu
19     private lateinit var tvUserInfo: TextView
20
21     private lateinit var auth: FirebaseAuth
22     private lateinit var database: FirebaseDatabase
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27
28         // Khởi tạo views
29         etEmail = findViewById(R.id.etEmail)
30         etPassword = findViewById(R.id.etPassword)
31         btnSignUp = findViewById(R.id.btnSignUp)
32         btnSignIn = findViewById(R.id.btnSignIn)
```

```
</> activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)  build.gradle.kts (.app)

12 class MainActivity : AppCompatActivity() {
13     override fun onCreate(savedInstanceState: Bundle?) {
14         btnShowData = findViewById(R.id.btnShowData) // Khởi tạo nút Hiển thị dữ liệu
15         tvUserInfo = findViewById(R.id.tvUserInfo)
16
17         // Khởi tạo Firebase
18         auth = FirebaseAuth.getInstance()
19         database = FirebaseDatabase.getInstance()
20
21         // Xử lý nút Đăng ký
22         btnSignUp.setOnClickListener {
23             val email = etEmail.text.toString().trim()
24             val password = etPassword.text.toString().trim()
25
26             if (email.isEmpty() || password.isEmpty()) {
27                 Toast.makeText(context, this, text: "Vui lòng nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show()
28                 return@setOnClickListener
29             }
30
31             auth.createUserWithEmailAndPassword(email, password)
32                 .addOnCompleteListener { task ->
33                     if (task.isSuccessful) {
34                         val user = auth.currentUser
35                         // Lưu thông tin người dùng vào Realtime Database
36                         val userId = user?.uid ?: ""
37                         val userRef = database.getReference(path: "users").child(userId)
38                         userRef.child(pathString: "email").setValue(email)
39                         Toast.makeText(context, this, text: "Đăng ký thành công", Toast.LENGTH_SHORT).show()
40                         loadUserInfo()
41                     } else {
42                         Toast.makeText(context, this, text: "Đăng ký thất bại: ${task.exception?.message}", Toast.LENGTH_LONG).show()
43                     }
44                 }
45         }
46     }
47 }
```

```

<? activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)  build.gradle.kts (:app)
12  class MainActivity : AppCompatActivity() {
24  override fun onCreate(savedInstanceState: Bundle?) {
41  btnSignUp.setOnClickListener {
64  }
65
66  // Xử lý nút Đăng nhập
67  btnSignIn.setOnClickListener {
68  val email = etEmail.text.toString().trim()
69  val password = etPassword.text.toString().trim()
70
71  if (email.isEmpty() || password.isEmpty()) {
72  Toast.makeText(context, this, text: "Vui lòng nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show()
73  return@setOnClickListener
74  }
75
76  auth.signInWithEmailAndPassword(email, password)
77  .addOnCompleteListener { task ->
78  if (task.isSuccessful) {
79  Toast.makeText(context, this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show()
80  loadUserInfo()
81  } else {
82  Toast.makeText(context, this, text: "Đăng nhập thất bại: ${task.exception?.message}", Toast.LENGTH_LONG).show()
83  }
84  }
85  }
86
87  // Xử lý nút Hiển thị dữ liệu
88  btnShowData.setOnClickListener {
89  val user = auth.currentUser
90  if (user != null) {
91  val userId = user.uid
92  val userRef = database.getReference(path: "users").child(userId)

```

```

<? activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)  build.gradle.kts (:app)
12  class MainActivity : AppCompatActivity() {
24  override fun onCreate(savedInstanceState: Bundle?) {
88  btnShowData.setOnClickListener {
89  val user = auth.currentUser
90  if (user != null) {
91  val userId = user.uid
92  val userRef = database.getReference(path: "users").child(userId)
93  userRef.get().addOnSuccessListener { snapshot ->
94  if (snapshot.exists()) {
95  val email = snapshot.child(path: "email").getValue(String::class.java)
96  tvUserInfo.text = "Dữ liệu từ Database:\nEmail: $email"
97  } else {
98  tvUserInfo.text = "Không có dữ liệu cho người dùng này"
99  }
100  }.addOnFailureListener {
101  tvUserInfo.text = "Không thể đọc dữ liệu: ${it.message}"
102  }
103  } else {
104  tvUserInfo.text = "Vui lòng đăng nhập trước"
105  Toast.makeText(context, this, text: "Vui lòng đăng nhập để hiển thị dữ liệu", Toast.LENGTH_SHORT).show()
106  }
107  }
108  }
109
110  private fun loadUserInfo() {
111  val user = auth.currentUser
112  if (user != null) {
113  val userId = user.uid
114  val userRef = database.getReference(path: "users").child(userId)
115  userRef.get().addOnSuccessListener { snapshot ->
116  val email = snapshot.child(path: "email").getValue(String::class.java)
117  tvUserInfo.text = "Người dùng: $email"

```

```
</> activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)  build.gradle.kts (:ap

12      class MainActivity : AppCompatActivity() {
24          override fun onCreate(savedInstanceState: Bundle?) {
88              btnShowData.setOnClickListener {
102                  }
103              } else {
104                  tvUserInfo.text = "Vui lòng đăng nhập trước"
105                  Toast.makeText(context: this, text "Vui lòng đăng nhập để hiển thị dữ liệu", Toast.LENGTH_SHORT).show()
106              }
107          }
108      }
109
110      private fun loadUserInfo() {
111          val user = auth.currentUser
112          if (user != null) {
113              val userId = user.uid
114              val userRef = database.getReference(path: "users").child(userId)
115              userRef.get().addOnSuccessListener { snapshot ->
116                  val email = snapshot.child(path: "email").getValue(String::class.java)
117                  tvUserInfo.text = "Người dùng: $email"
118              }.addOnFailureListener {
119                  tvUserInfo.text = "Không thể tải thông tin người dùng"
120              }
121          } else {
122              tvUserInfo.text = "Chưa đăng nhập"
123          }
124      }
125  }
```

```
</> activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:tools="http://schemas.android.com/tools">
4      <uses-permission android:name="android.permission.INTERNET" />
5  </manifest>
```

```
</> activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)

You can use the Project Structure dialog to view and edit your project configuration

1  // Top-level build file where you can add configuration options common to all sub-projects/modules.
2  plugins {
3      alias(libs.plugins.android.application) apply false
4      alias(libs.plugins.kotlin.android) apply false
5  }
6  buildscript {
7      dependencies {
8          // Thêm Google Services plugin
9          classpath("com.google.gms:google-services:4.4.2")
10     }
11 }
12
```



```
activity_main.xml  AndroidManifest.xml  MainActivity.kt  build.gradle.kts (Ex3)  settings.gradle.kts (Ex3)

You can use the Project Structure dialog to view and edit your project configuration

1  pluginManagement {
2      repositories {
3          google {
4              content {
5                  includeGroupByRegex("com\\.android.*")
6                  includeGroupByRegex("com\\.google.*")
7                  includeGroupByRegex("androidx.*")
8              }
9          }
10         mavenCentral()
11         gradlePluginPortal()
12     }
13 }
14 dependencyResolutionManagement {
15     repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
16     repositories {
17         google()
18         mavenCentral()
19     }
20 }
21
22 rootProject.name = "Ex3"
23 include(...)projectPaths: ":app"
24
```

```
build.gradle.kts (:app) x

You can use the Project Structure dialog to view and edit your project configuration

1  plugins {
2      alias(libs.plugins.android.application)
3      alias(libs.plugins.kotlin.android)
4      id("com.google.gms.google-services")
5  }
6
```

```
48
49     // Firebase dependencies
50     implementation(platform("com.google.firebase:firebase-bom:33.1.2"))
51     implementation("com.google.firebase:firebase-auth-ktx")
52     implementation("com.google.firebase:firebase-database-ktx")
53 }
```

```
build.gradle.kts (:app) build.gradle.kts (Ex3) ×
You can use the Project Structure dialog to view and edit your project configuration

1 // Top-level build file where you can add configuration options co
2 plugins {
3     alias(libs.plugins.android.application) apply false
4     alias(libs.plugins.kotlin.android) apply false
5 }
6 buildscript {
7     dependencies {
8         // Thêm Google Services plugin
9         classpath("com.google.gms:google-services:4.4.2")
10    }
11 }
12
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Firebase Demo"
        android:textSize="24sp"
        android:textStyle="bold"
        android:gravity="center"
        android:layout_marginBottom="24dp"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="8dp"/>

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Mật khẩu"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>
</LinearLayout>
```

```
</> activity_main.xml ×
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
32
33     <LinearLayout
34         android:layout_width="match_parent"
35         android:layout_height="wrap_content"
36         android:orientation="horizontal">
37
38         <Button
39             android:id="@+id/btnSignUp"
40             android:layout_width="0dp"
41             android:layout_height="wrap_content"
42             android:layout_weight="1"
43             android:text="Đăng ký"
44             android:layout_marginEnd="4dp"/>
45
46         <Button
47             android:id="@+id/btnSignIn"
48             android:layout_width="0dp"
49             android:layout_height="wrap_content"
50             android:layout_weight="1"
51             android:text="Đăng nhập"
52             android:layout_marginStart="4dp"/>
53     </LinearLayout>
54
55     <Button
56         android:id="@+id/btnShowData"
57         android:layout_width="match_parent"
58         android:layout_height="wrap_content"
59         android:text="Hiển thị dữ liệu"
60         android:layout_marginTop="16dp"/>
61
62     <TextView
```

</> activity\_main.xml ×

```
2   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
33   <LinearLayout
44       android:layout_marginEnd="4dp"/>
45
46   <Button
47       android:id="@+id/btnSignIn"
48       android:layout_width="0dp"
49       android:layout_height="wrap_content"
50       android:layout_weight="1"
51       android:text="Đăng nhập"
52       android:layout_marginStart="4dp"/>
53   </LinearLayout>
54
55   <Button
56       android:id="@+id/btnShowData"
57       android:layout_width="match_parent"
58       android:layout_height="wrap_content"
59       android:text="Hiển thị dữ liệu"
60       android:layout_marginTop="16dp"/>
61
62   <TextView
63       android:id="@+id/tvUserInfo"
64       android:layout_width="match_parent"
65       android:layout_height="wrap_content"
66       android:layout_marginTop="24dp"
67       android:textSize="16sp"/>
68 </LinearLayout>
```