**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

A blue and white logo

AI-generated content may be incorrect.

**BÀI TẬP THỰC HÀNH SỐ 4**

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**NỘI DUNG BỔ SUNG: ỨNG DỤNG VỚI CSDL**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 2251172523 | Nguyễn Minh Tiên | 64KTPM3 |

**Hà Nội, năm 2025**

**BÀI TẬP 1: SHARED PREFERENCE**

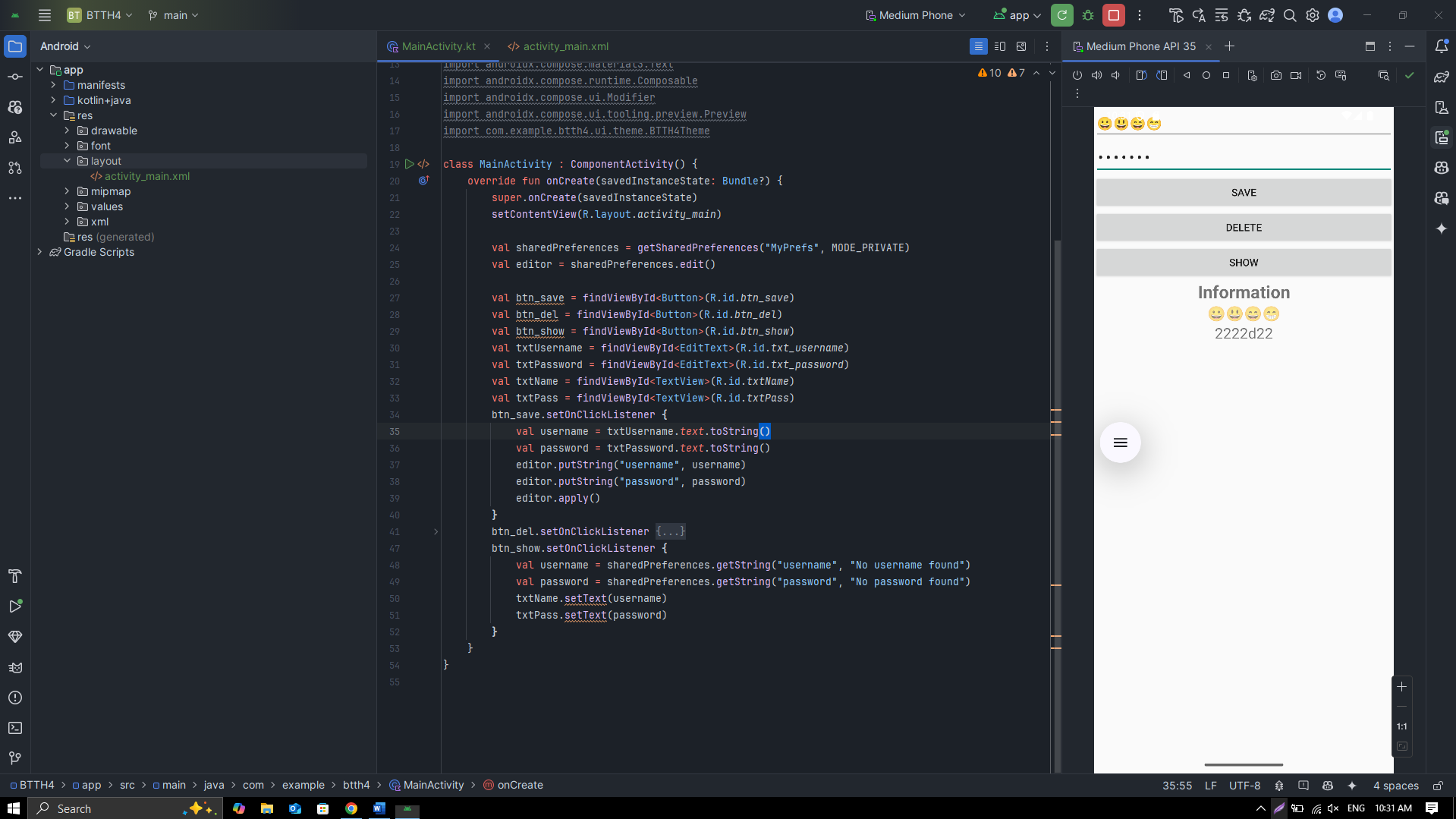
**Mục tiêu:**

* 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>>



**BÀI TẬP 2: SQLite**

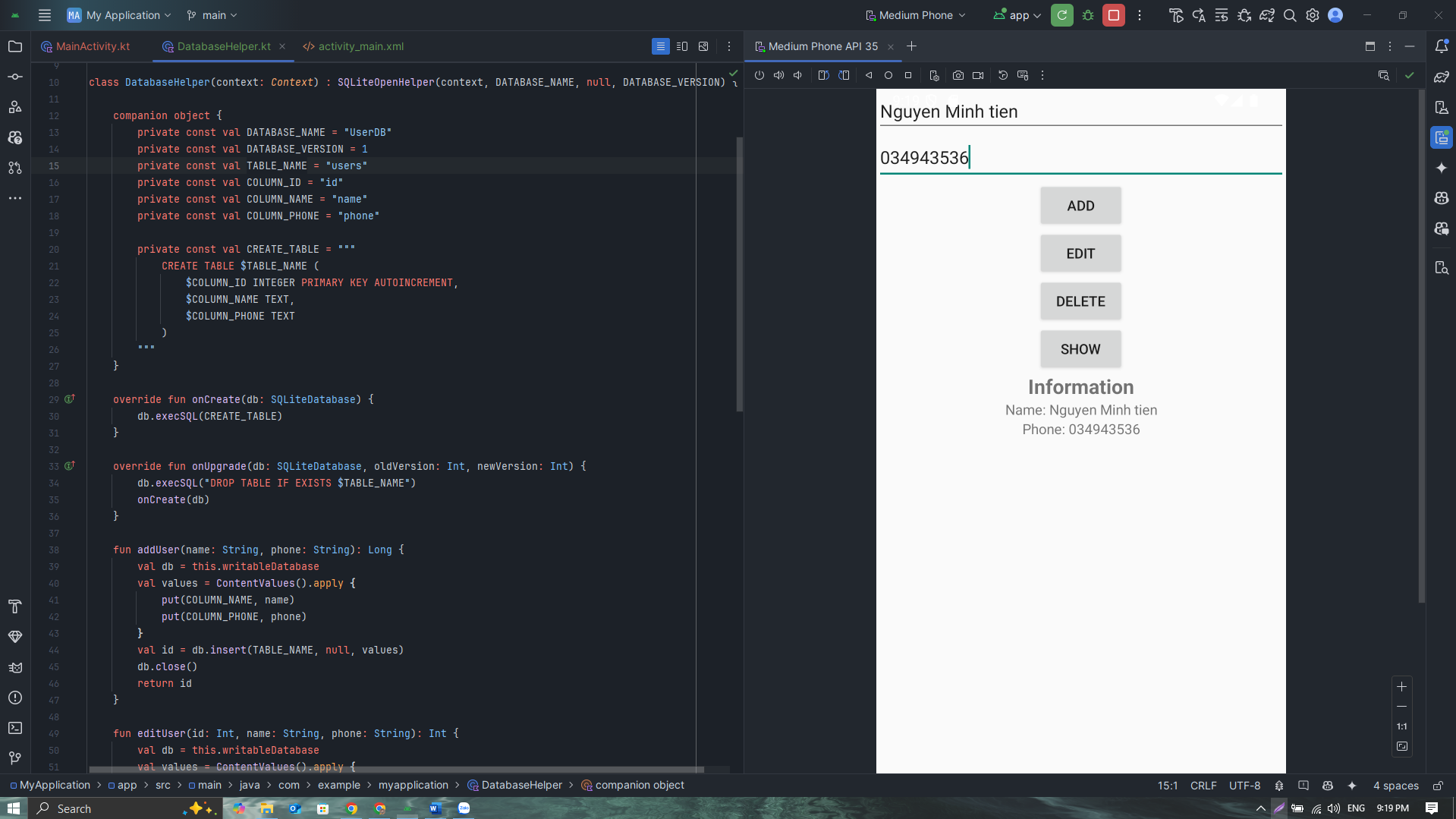
**Mục tiêu:**

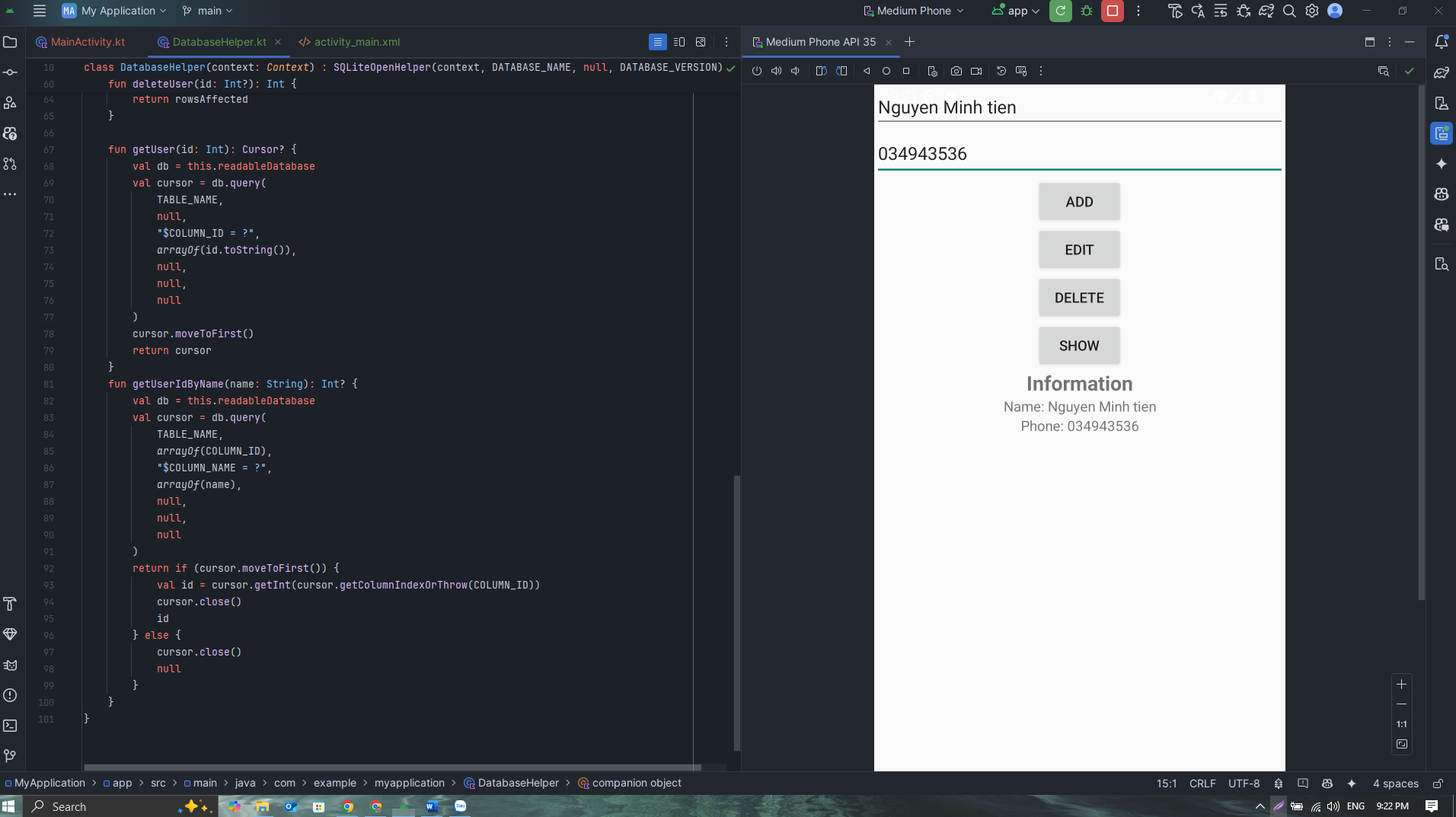
* Hiểu cách sử dụng SQLite để lưu trữ dữ liệu trong ứng dụng Android.
* Thực hành tạo cơ sở dữ liệu SQLite, thêm, sửa, xóa dữ liệu.

**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**





**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** |

1. **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**

