**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

A blue and white logo

AI-generated content may be incorrect.

**ÀI TẬP THỰC HÀNH SỐ 4**

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 |  |  |  |

**Hà Nội, năm 2025**

**BÀI TẬP 1: SHARED PREFERENCE**

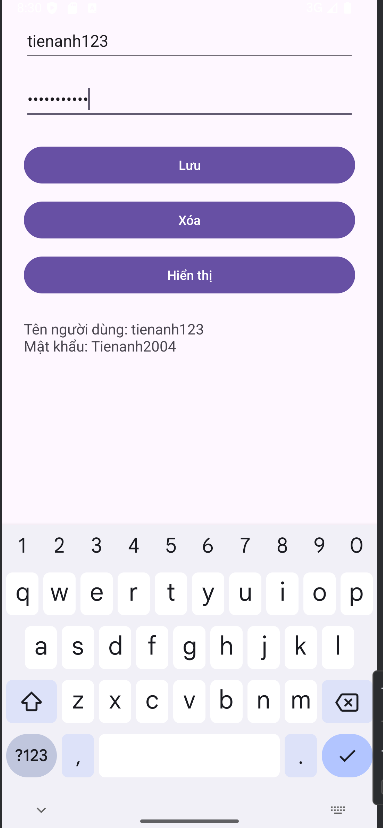
**Mục tiêu:**

* 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>>

**A screenshot of a phone

AI-generated content may be incorrect.** **A screenshot of a phone

AI-generated content may be incorrect.**

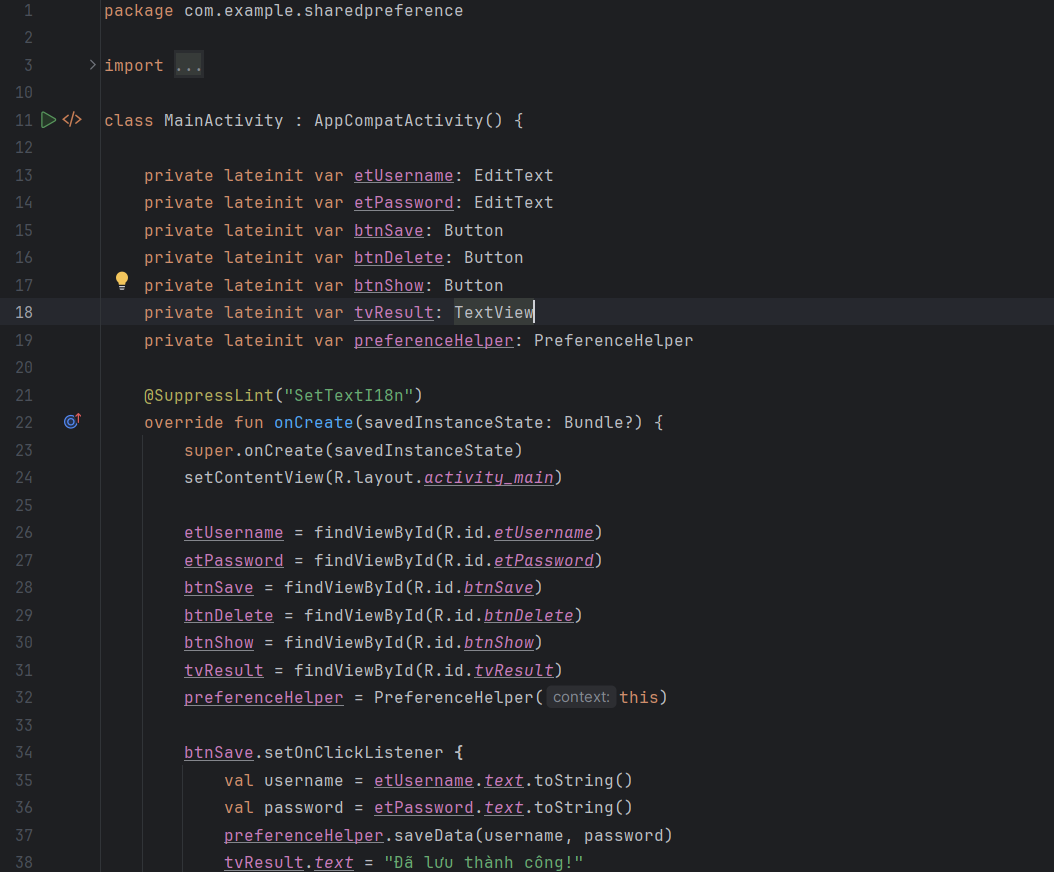
**Activity\_main.xml**

****

**A screen shot of a computer program

AI-generated content may be incorrect.**

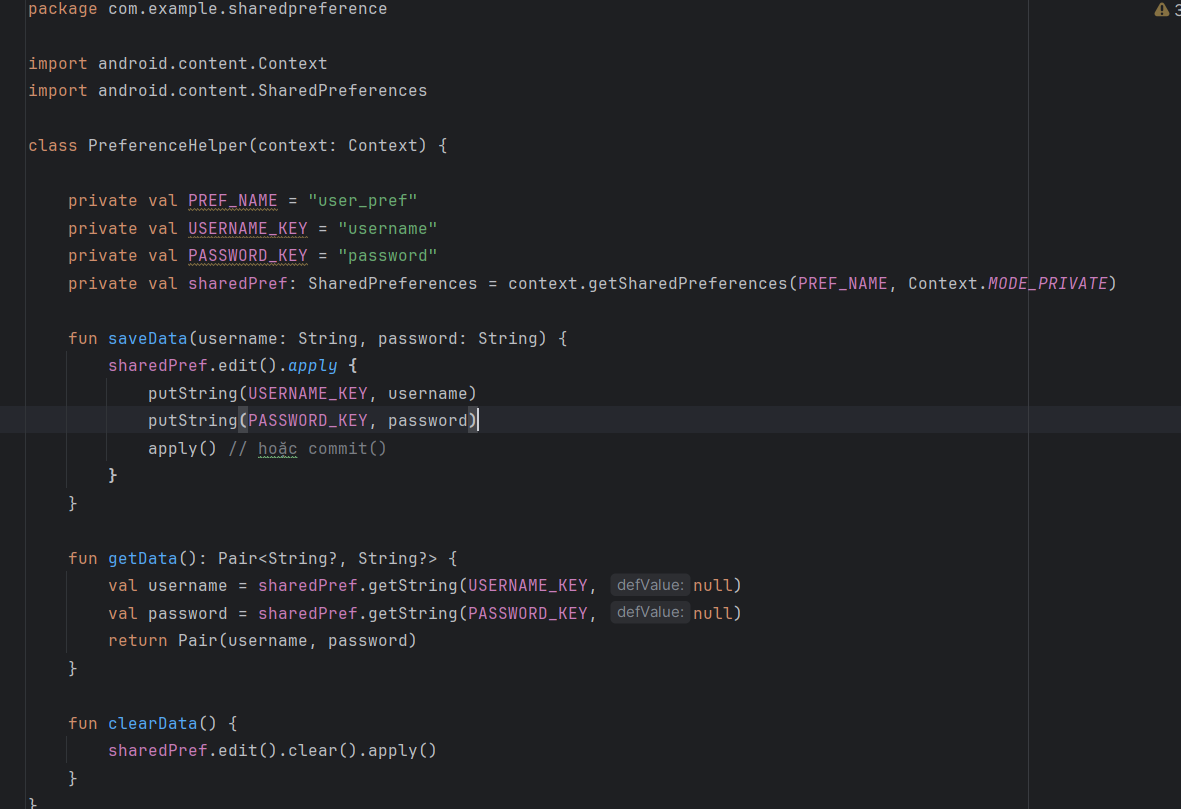
**MainActivity.kt**

****

**A computer screen with colorful text

AI-generated content may be incorrect.**

**Lớp PreferenceHelper.kt**

****

**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>>

**A screenshot of a phone

AI-generated content may be incorrect.**

**A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.  
A screenshot of a phone

AI-generated content may be incorrect.**

**Activity\_main.xml**

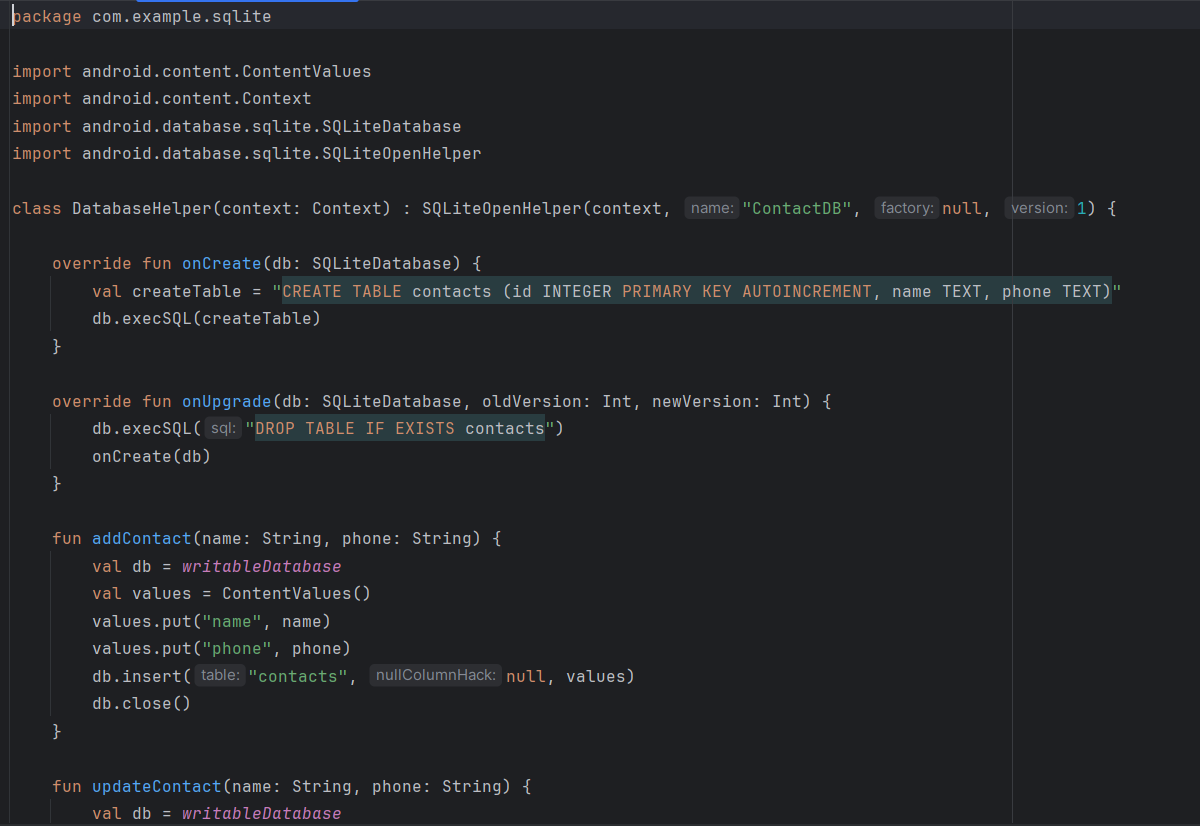
**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Lớp helper**

****

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**MainActivity.kt**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**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.

|  |
| --- |
| **Giới thiệu tổng quan**  **Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Firebase cung cấp một tập hợp các công cụ và dịch vụ hỗ trợ các lập trình viên xây dựng các ứng dụng chất lượng cao, tăng trưởng người dùng và tối ưu hóa lợi nhuận. Firebase tích hợp nhiều dịch vụ như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ đám mây, phân tích và nhiều tính năng khác giúp rút ngắn thời gian phát triển và vận hành ứng dụng.**  **Firebase được thiết kế để dễ dàng tích hợp với các ứng dụng Android, iOS, web và cả các framework đa nền tảng như Flutter, React Native.**  **Lịch sử phát triển**   * **2011: Firebase bắt đầu như một công ty khởi nghiệp được sáng lập bởi James Tamplin và Andrew Lee. Sản phẩm đầu tiên của họ là Firebase Realtime Database, một cơ sở dữ liệu NoSQL cung cấp khả năng đồng bộ dữ liệu theo thời gian thực giữa các thiết bị.** * **2014: Google mua lại Firebase, mở rộng các dịch vụ và tích hợp sâu vào hệ sinh thái Google Cloud.** * **2016: Firebase chính thức ra mắt bộ nền tảng hoàn chỉnh tại Google I/O với hàng loạt dịch vụ mới như Firebase Authentication, Firebase Cloud Messaging, Firebase Hosting, Firebase Analytics...** * **Từ 2017 đến nay: Firebase không ngừng mở rộng và nâng cấp, tích hợp thêm các tính năng như Firebase ML (máy học), Performance** * **Monitoring, Test Lab, và Crashlytics. Hiện tại, Firebase là một trong những nền tảng phổ biến nhất được các lập trình viên sử dụng để phát triển ứng dụng.** * **2. Mô tả chi tiết từng dịch vụ chính của Firebase** * **2.1. Firebase Realtime Database** * **Là cơ sở dữ liệu NoSQL lưu trữ và đồng bộ dữ liệu theo thời gian thực giữa các client và máy chủ.** * **Dữ liệu được lưu dưới dạng JSON và mọi thay đổi sẽ được đồng bộ ngay lập tức đến tất cả người dùng đang kết nối.** * **2.2. Cloud Firestore** * **Cơ sở dữ liệu NoSQL thế hệ mới, cải tiến từ Realtime Database.** * **Hỗ trợ cấu trúc dữ liệu linh hoạt hơn với các collection và document.** * **Cung cấp khả năng truy vấn mạnh mẽ, đồng bộ thời gian thực và tích hợp chặt chẽ với Google Cloud.** * **2.3. Firebase Authentication** * **Dịch vụ xác thực người dùng qua email/password, số điện thoại, tài khoản Google, Facebook, Twitter, GitHub và các hệ thống OAuth khác.** * **Cung cấp SDK dễ dàng tích hợp cho web và mobile.** * **2.4. Firebase Cloud Messaging (FCM)** * **Nền tảng gửi thông báo đẩy miễn phí đến các thiết bị Android, iOS và web.** * **Hỗ trợ cả tin nhắn dữ liệu (data message) và thông báo hiển thị (notification message).** * **2.5. Firebase Hosting** * **Dịch vụ lưu trữ và phân phối nội dung tĩnh (HTML, CSS, JS...) qua CDN toàn cầu.** * **Hỗ trợ HTTPS mặc định, dễ dàng triển khai các ứng dụng web hoặc Progressive Web App (PWA).** * **2.6. Firebase Cloud Functions** * **Cho phép viết các đoạn mã backend dưới dạng serverless functions, chạy trên hạ tầng Google Cloud.** * **Được kích hoạt bởi các sự kiện từ Firebase (ví dụ như ghi dữ liệu vào Firestore, người dùng đăng ký mới).** * **2.7. Firebase Crashlytics** * **Công cụ theo dõi lỗi ứng dụng (crash reporting) theo thời gian thực.** * **Giúp lập trình viên phát hiện và sửa lỗi nhanh chóng, ưu tiên xử lý các sự cố ảnh hưởng lớn đến người dùng.** * **2.8. Firebase Performance Monitoring** * **Giúp theo dõi hiệu suất ứng dụng, xác định các vấn đề về tốc độ load, thời gian phản hồi API hoặc các thao tác chậm trên thiết bị người dùng.** * **2.9. Firebase Analytics (Google Analytics for Firebase)** * **Hệ thống phân tích hành vi người dùng miễn phí, cho phép theo dõi các sự kiện trong ứng dụng, phân khúc người dùng và đo lường hiệu quả chiến dịch marketing.** * **2.10. Firebase ML (Machine Learning)** * **Dịch vụ tích hợp các mô hình máy học như nhận diện văn bản, nhận diện khuôn mặt, dịch ngôn ngữ...** * **Hỗ trợ cả các mô hình custom (tùy chỉnh) do lập trình viên tự huấn luyện.** * **3. Lợi ích và ứng dụng của Firebase trong phát triển ứng dụng** * **3.1. Lợi ích** * **Tiết kiệm thời gian phát triển: Firebase cung cấp sẵn các dịch vụ backend, giúp lập trình viên tập trung vào giao diện và tính năng ứng dụng mà không phải xây dựng server phức tạp.** * **Dễ dàng tích hợp: Firebase cung cấp SDK cho đa nền tảng (Android, iOS, Web, Flutter...), giúp triển khai nhanh chóng.** * **Mở rộng linh hoạt: Firebase dễ dàng tích hợp với các dịch vụ của Google Cloud hoặc bên thứ ba.** * **Hiệu suất cao và bảo mật: Các dịch vụ của Firebase vận hành trên hạ tầng Google Cloud, đảm bảo hiệu suất và bảo mật cao.** * **Quản lý và phân tích dữ liệu người dùng: Nhờ Firebase Analytics và các công cụ đo lường khác, lập trình viên có thể hiểu rõ hành vi người dùng và tối ưu ứng dụng hiệu quả.** * **3.2. Ứng dụng thực tế** * **Ứng dụng chat: Firebase Realtime Database hoặc Firestore giúp xây dựng ứng dụng nhắn tin thời gian thực.** * **Ứng dụng thương mại điện tử: Firebase Authentication hỗ trợ đăng nhập, còn Firestore lưu trữ sản phẩm và đơn hàng.** * **Ứng dụng mạng xã hội: Firebase Hosting, Cloud Functions và Firestore kết hợp để xây dựng nền tảng mạng xã hội quy mô nhỏ.** * **Ứng dụng tin tức/PWA: Firebase Hosting và FCM được dùng để tạo các trang web tin tức nhanh, tối ưu SEO và đẩy thông báo bài viết mới.** * **Ứng dụng tích hợp AI: Firebase ML giúp tích hợp các tính năng thông minh như OCR, nhận diện hình ảnh hoặc dịch ngôn ngữ vào ứng dụng.** |

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**

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