**Tên: Lê Thanh Tịnh 1150080160**

# DỰ ÁN 4 – ỨNG DỤNG “QUẢN LÝ CHI TIÊU CÁ NHÂN”

Mục tiêu của dự án là xây dựng một ứng dụng Flutter giúp người dùng ghi lại, theo dõi   
và phân tích các khoản thu chi hằng ngày. Ứng dụng này được thiết kế với giao diện thân thiện,   
hoạt động offline hoàn toàn nhờ sử dụng cơ sở dữ liệu SQLite cục bộ.

## 1. Công nghệ sử dụng

- Flutter SDK 3.x (Dart)  
- Sqflite (cơ sở dữ liệu SQLite local)  
- Provider (quản lý trạng thái)  
- Path & Path Provider (lưu trữ tệp database)  
- Intl (định dạng tiền tệ, ngày tháng)  
- CustomPainter (vẽ biểu đồ Pie Chart)

## 2. Kiến trúc và cấu trúc thư mục

lib/  
 ├─ main.dart → Điểm khởi chạy ứng dụng  
 ├─ models/transaction.dart → Mô hình dữ liệu giao dịch  
 ├─ providers/transaction\_provider.dart → Lớp quản lý DB & trạng thái  
 ├─ screens/home\_screen.dart → Giao diện trang chính  
 ├─ widgets/add\_transaction\_form.dart → Form nhập giao dịch  
 └─ widgets/transaction\_chart.dart → Biểu đồ thống kê chi tiêu

## 3. Mô tả chức năng chính

• Thêm giao dịch: Người dùng nhập mô tả, số tiền, danh mục và ngày giao dịch thông qua form có kiểm tra dữ liệu hợp lệ.

• Lưu trữ dữ liệu: Toàn bộ thông tin được lưu vào SQLite cục bộ, đảm bảo hoạt động không cần Internet.

• Hiển thị danh sách: ListView.builder hiển thị toàn bộ lịch sử giao dịch, có định dạng tiền tệ Việt Nam (Intl).

• Phân tích chi tiêu: Biểu đồ tròn (Pie Chart) thể hiện tỷ lệ chi tiêu theo danh mục, được vẽ bằng CustomPainter.

• Quản lý trạng thái: Provider giúp đồng bộ danh sách giao dịch và biểu đồ khi có thay đổi.

## 4. Cấu trúc dữ liệu

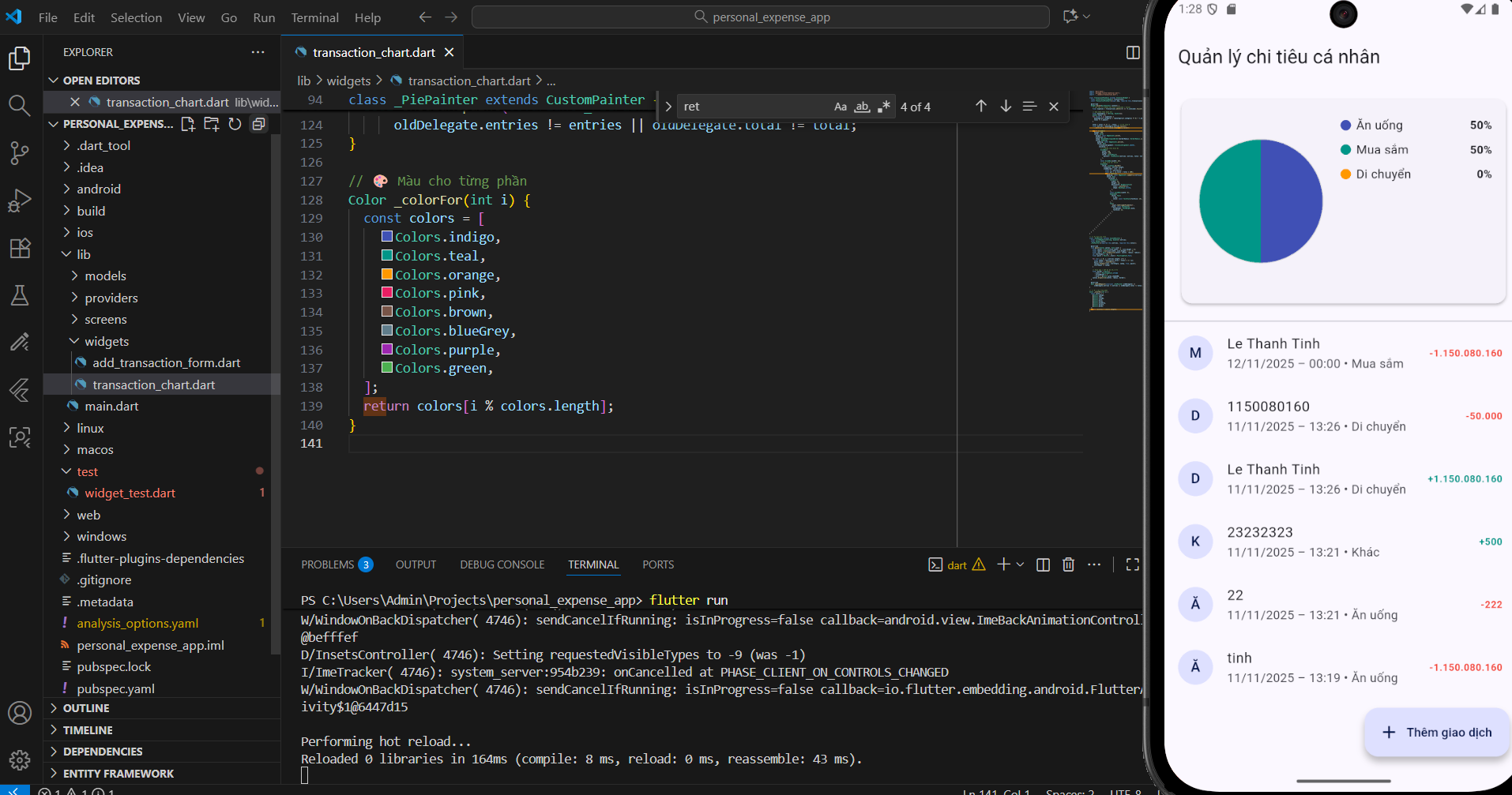
Bảng: transactions  
 ├─ id (INTEGER, PRIMARY KEY AUTOINCREMENT)  
 ├─ title (TEXT) – mô tả giao dịch  
 ├─ amount (REAL) – số tiền  
 ├─ date (TEXT ISO8601) – ngày tháng  
 ├─ category (TEXT) – danh mục  
 └─ isIncome (INTEGER) – 1 = Thu nhập, 0 = Chi tiêu

## 5. Giao diện người dùng (UI)

Giao diện ứng dụng được thiết kế Material 3, gồm thanh AppBar, biểu đồ tổng hợp ở đầu,   
và danh sách chi tiết bên dưới. Nút nổi (FloatingActionButton) giúp thêm giao dịch nhanh.

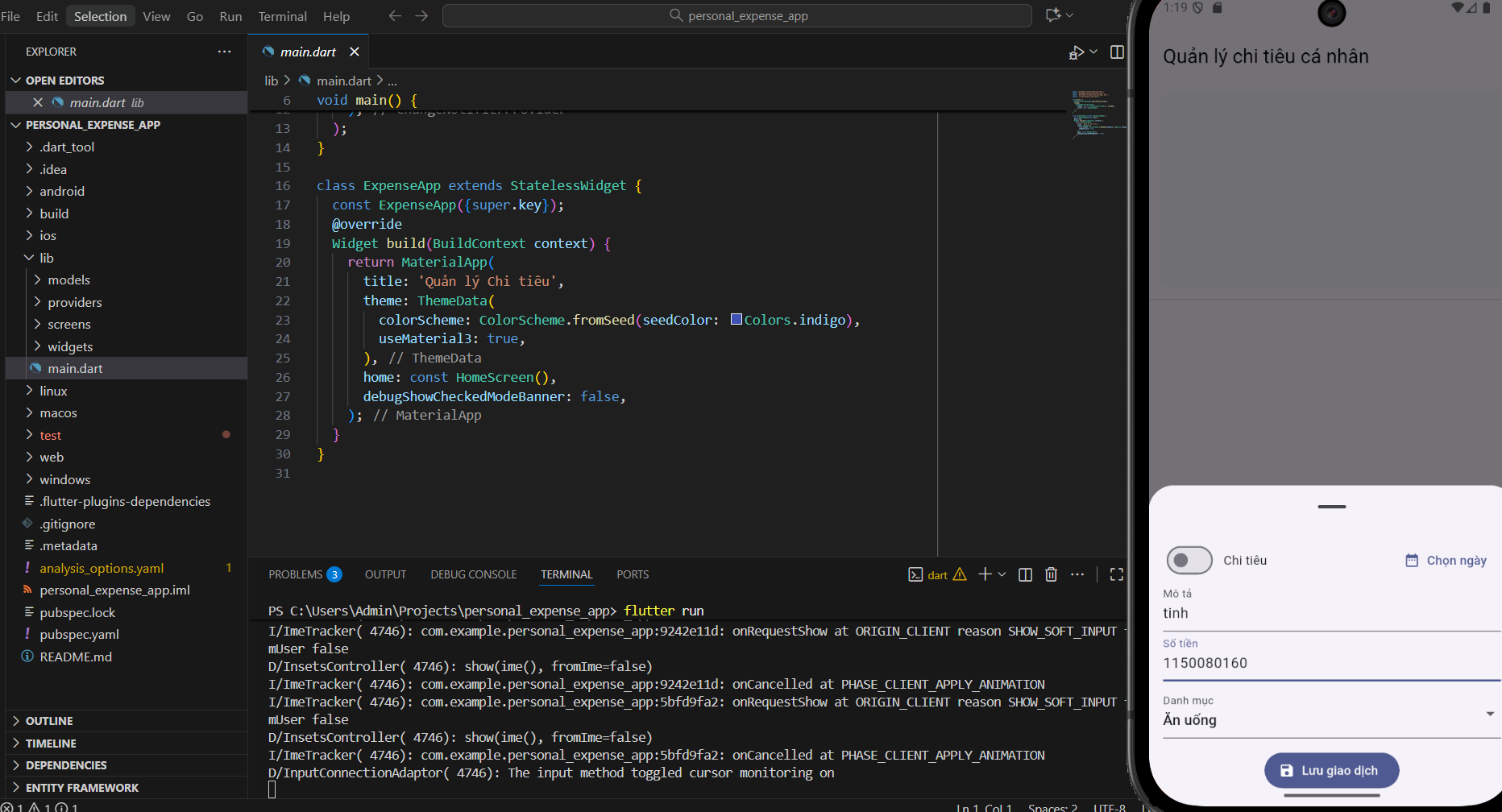
## 6. Biểu đồ thống kê

Biểu đồ tròn hiển thị tỷ lệ chi tiêu từng danh mục. CustomPainter được dùng để vẽ từng   
phân đoạn màu tương ứng với từng danh mục. Tổng chi tiêu được tính và phân bổ theo tỷ lệ phần trăm.

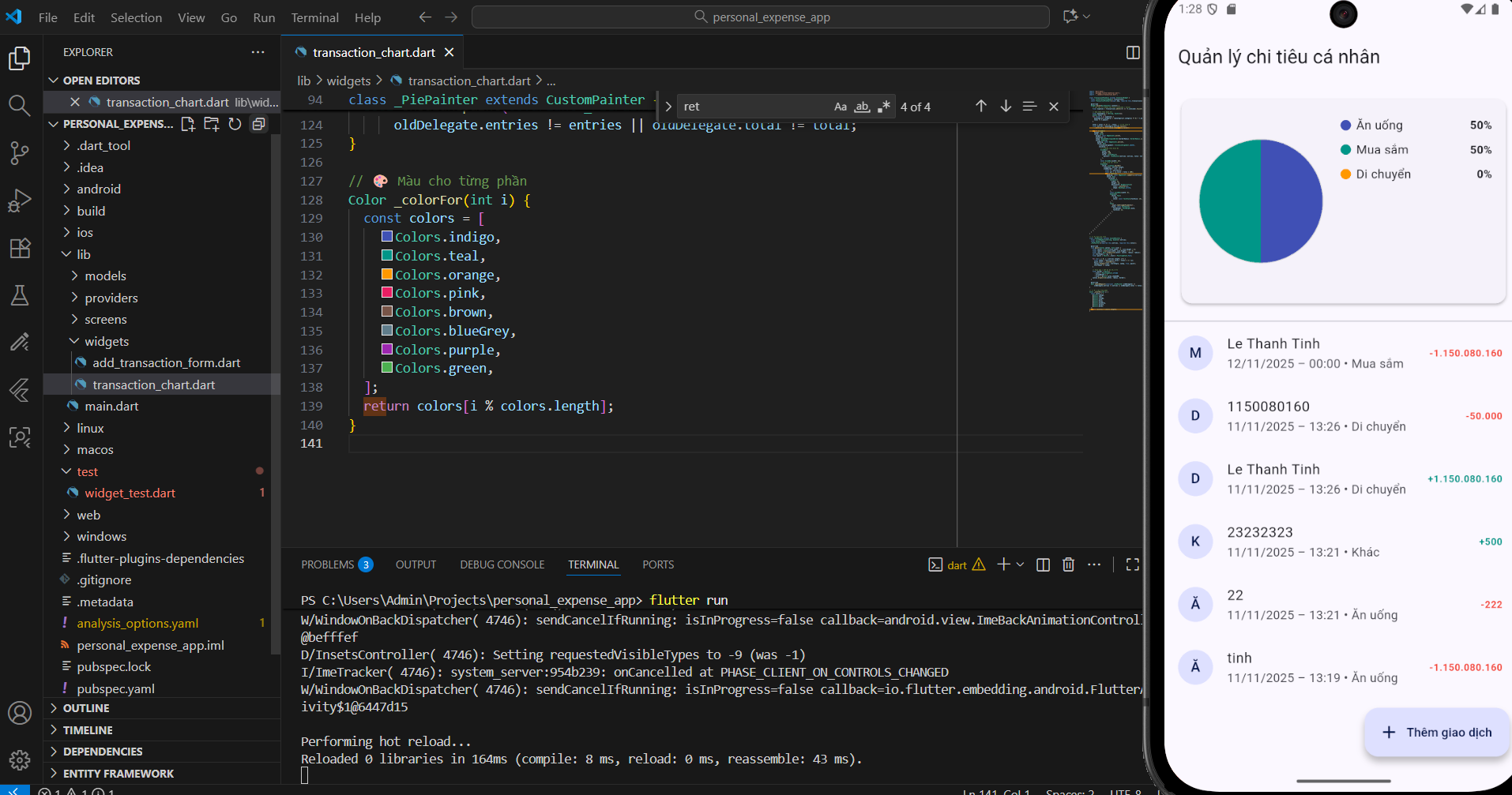


## 7. Nhận xét và đánh giá

Ứng dụng đáp ứng đầy đủ yêu cầu của đề bài: sử dụng lưu trữ cục bộ, xử lý form, hiển thị   
và phân tích dữ liệu trực quan. Dự án được tối ưu cho hiệu năng và dễ mở rộng, ví dụ có thể   
bổ sung biểu đồ cột (Bar Chart), xuất báo cáo PDF hoặc đồng bộ lên Cloud.

Màn hình thêm quản lí chi tiêu cá nhân  


Màn hình sau khi thêm



# DỰ ÁN 5 – ỨNG DỤNG “SÁCH ĐIỆN TỬ (E-BOOK READER)”

Ứng dụng “Sách điện tử” được xây dựng nhằm mô phỏng trải nghiệm đọc sách   
với hiệu ứng lật trang mượt mà, lưu cài đặt đọc cá nhân và chế độ sáng/tối.

## 1. Công nghệ sử dụng

- Flutter SDK 3.x  
- PageView (hiệu ứng lật trang)  
- CustomPainter (vẽ nội dung văn bản)  
- SharedPreferences (lưu cài đặt người dùng)  
- Asset Bundle (đọc file .txt trong assets)

## 2. Cấu trúc thư mục

assets/book.txt  
 lib/  
 main.dart # Khởi động app + lưu theme  
 theme.dart # Light/Dark Material 3  
 screens/reader\_screen.dart# PageView + CustomPainter + lưu trạng thái  
 widgets/reader\_controls.dart # Nút trang trước/sau + cỡ chữ  
 pubspec.yaml

## 3. Mô tả chức năng

• Đọc file: Ứng dụng đọc nội dung từ file `assets/book.txt` khi khởi động.

• Lật trang: Dùng PageView để chuyển trang mượt mà bằng thao tác vuốt ngang.

• Hiển thị nội dung: CustomPainter + TextPainter vẽ văn bản, giúp tuỳ chỉnh bố cục và font.

• Lưu cài đặt: SharedPreferences ghi nhớ trang cuối cùng, kích thước chữ và chế độ đọc.

• Thanh điều hướng: BottomAppBar có nút tăng/giảm cỡ chữ, chuyển trang, đổi chế độ sáng/tối.

## 4. Luồng hoạt động

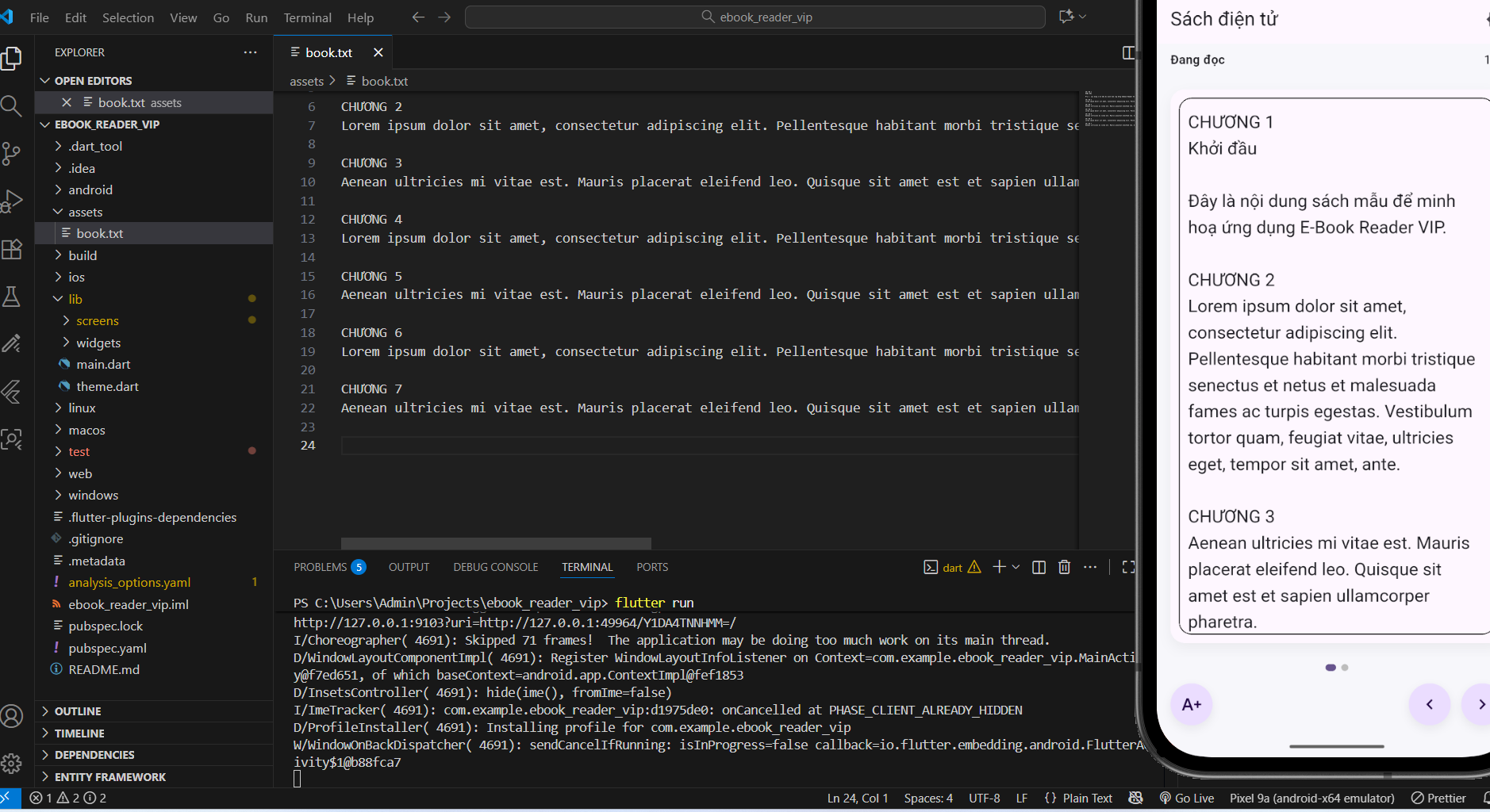
Khi ứng dụng mở, nó đọc nội dung file sách, chia nhỏ thành các trang (paginate),   
hiển thị từng trang bằng PageView. Khi người dùng thay đổi kích thước chữ hoặc lật trang,   
dữ liệu được lưu lại bằng SharedPreferences để lần mở tiếp theo đọc tiếp từ vị trí đó.

## 5. Giao diện người dùng

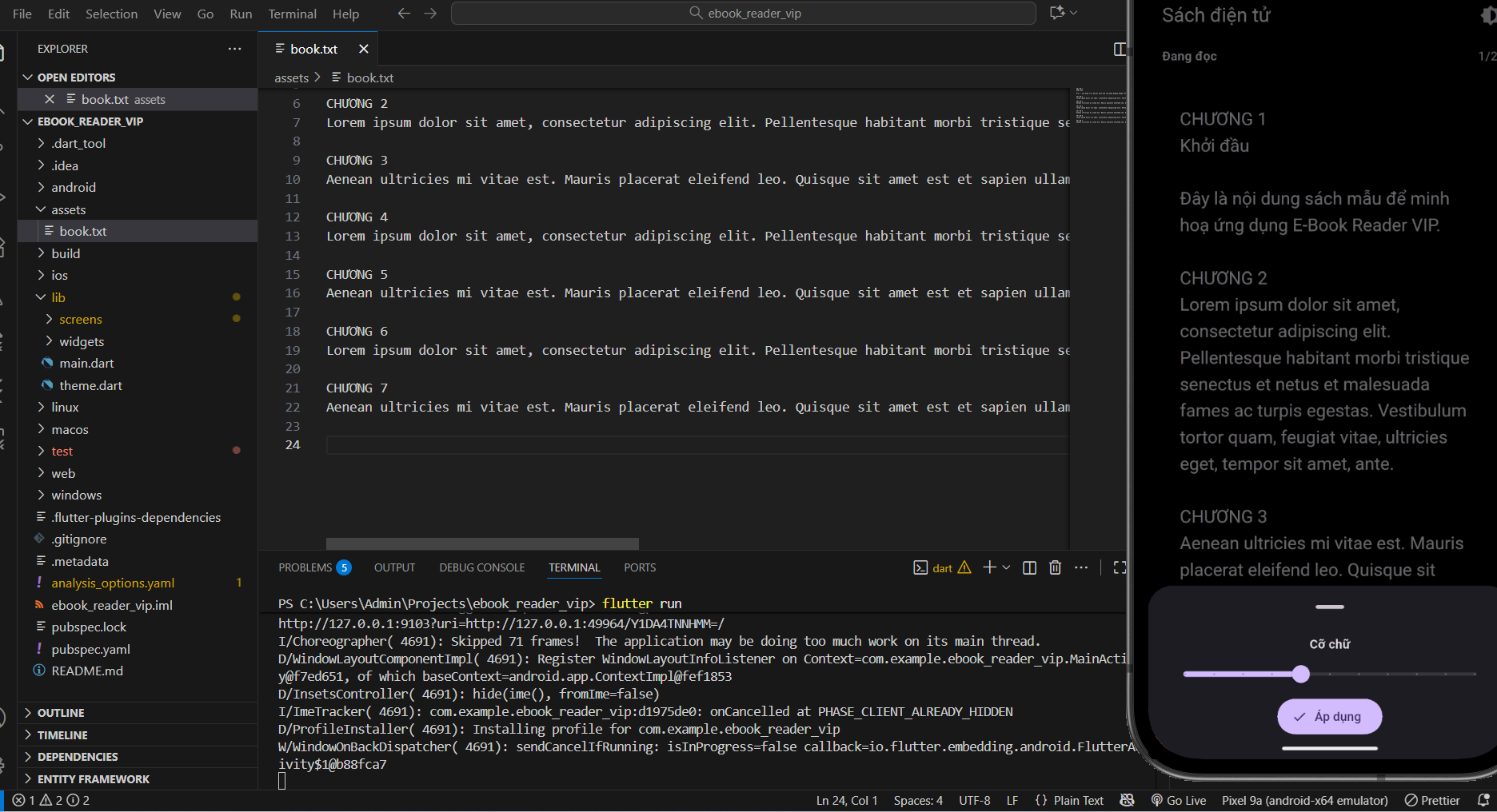
Thiết kế sử dụng Material 3 hiện đại. AppBar chứa tiêu đề và nút chuyển sáng/tối.  
BottomAppBar hiển thị các nút điều khiển cỡ chữ và trang. Phần nội dung là vùng đọc được   
vẽ bằng CustomPainter, giúp trình bày văn bản giống trang sách thật.

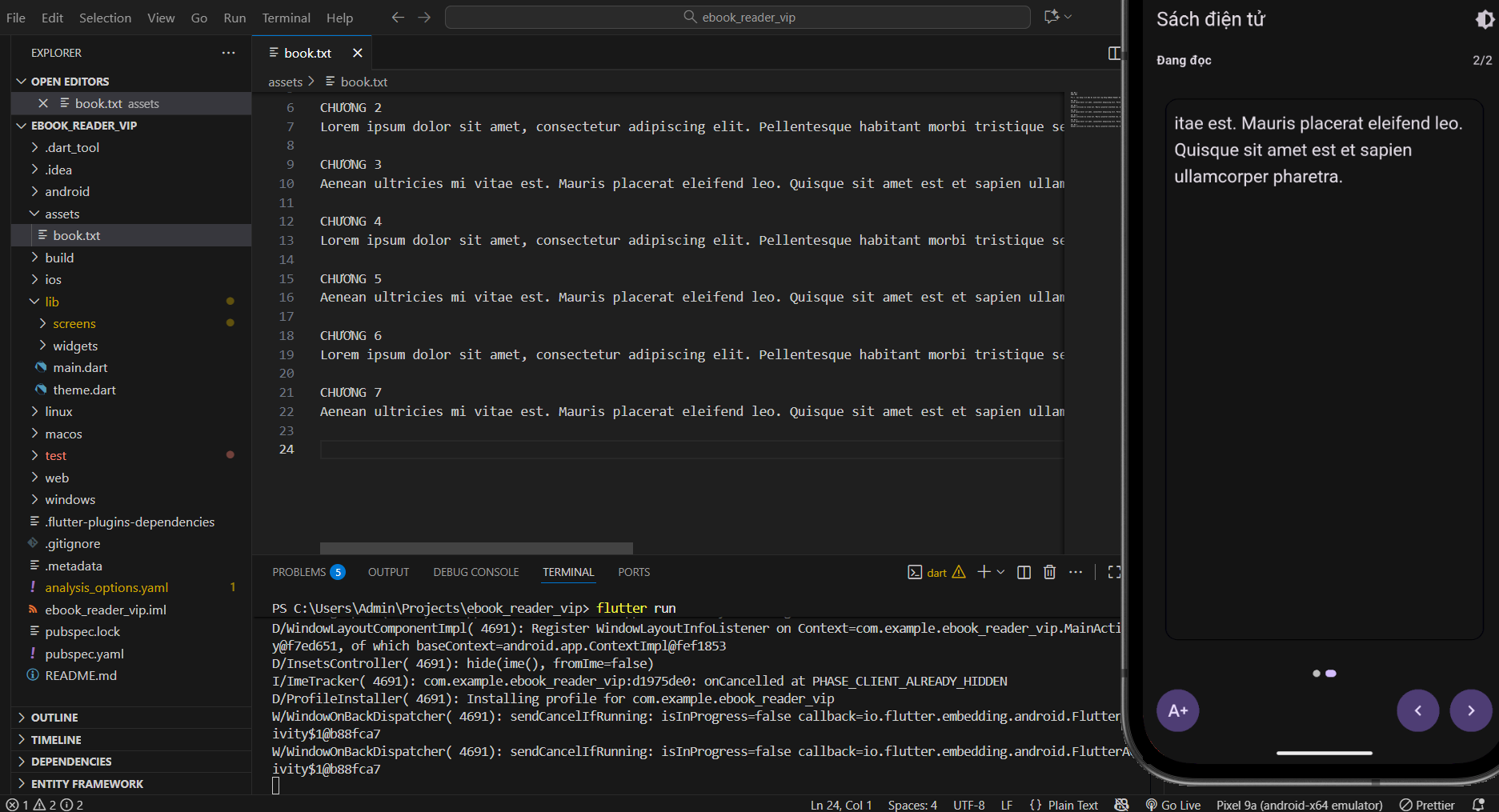
## 6. Nhận xét và đánh giá

Ứng dụng mang lại trải nghiệm đọc tự nhiên, tận dụng tốt khả năng của Flutter   
với PageView và CustomPainter. Việc lưu trạng thái người đọc giúp cá nhân hóa trải nghiệm,

Màn hình chế độ sáng và giao diện  


Màn hình chế độ tối và chỉnh cỡ chữ, qua trang





# DỰ ÁN 6: HỆ THỐNG ĐÁNH GIÁ NHÀ HÀNG (FIREBASE)

## 1. TỔNG QUAN DỰ ÁN

### 1.1 Mục tiêu & Phạm vi

• Xây dựng ứng dụng Flutter cho phép người dùng xem danh sách nhà hàng, đọc/gửi đánh giá kèm ảnh.

• Tích hợp Firebase: Authentication, Cloud Firestore, Cloud Storage, Cloud Messaging (FCM).

• UI thân thiện: ListView danh sách nhà hàng; bottom sheet gửi đánh giá; thông báo khi có đánh giá mới.

### 1.2. Công nghệ chính

- Firebase Authentication: Email/Password + Anonymous

- Cloud Firestore: Lưu nhà hàng, đánh giá, điểm trung bình

- Firebase Storage: Lưu ảnh đánh giá

- Firebase Cloud Messaging: Gửi/nhận thông báo

- image\_picker: Chọn ảnh từ thư viện

## 2. KIẾN TRÚC ỨNG DỤNG

### 2.1. Cấu trúc thư mục

lib/

├── main.dart # Entry point

├── firebase\_options.dart # Cấu hình Firebase

├── services/ # Business logic

│ ├── auth\_service.dart # Xác thực

│ ├── storage\_service.dart # Upload ảnh

│ └── fcm\_service.dart # Thông báo

├── repositories/ # Data access layer

│ └── restaurant\_repository.dart # CRUD operations

├── data/models/ # Data models

│ ├── restaurant.dart

│ └── review.dart

└── ui/ # Presentation layer

├── auth/sign\_in\_screen.dart

├── restaurants/

│ ├── restaurant\_list\_screen.dart

│ ├── restaurant\_detail\_screen.dart

│ ├── add\_review\_bottom\_sheet.dart

│ └── my\_reviews\_screen.dart

└── notifications/notifications\_screen.dart

### 2.2. Cơ sở dữ liệu Firestore

restaurants (collection)

└── {restaurantId}

├── name: String

├── photoUrl: String

├── avgRating: Double

└── reviews (subcollection)

└── {reviewId}

├── userId: String

├── content: String

├── rating: Integer (1-5)

├── imageUrl: String

└── createdAt: Timestamp

**Giải thích nhanh các đoạn code chính**

• main.dart: Khởi tạo Firebase (ưu tiên `firebase\_options.dart`), đăng ký handler FCM, điều hướng.

• AuthService: Đăng ký/đăng nhập email + ẩn danh.

• RestaurantRepository.addReview(): Ghi review vào subcollection và cập nhật `avgRating`.

• StorageService.uploadReviewImage(): Tải ảnh lên Storage theo thư mục `reviews/{restaurantId}/...`.

• FcmService.init(): Xin quyền, hiển thị local notification khi nhận message foreground, subscribe topic `new\_reviews`.

• UI: `RestaurantListScreen` hiển thị danh sách từ stream Firestore; `AddReviewBottomSheet` gửi review.

## 3. CÁC CHỨC NĂNG CHÍNH

### 3.1. Xác thực người dùng (Firebase Authentication)

**File**: services/auth\_service.dart

**Chức năng:**

* Đăng nhập/đăng ký bằng Email và Password
* Đăng nhập ẩn danh cho người dùng thử nghiệm
* Stream theo dõi trạng thái đăng nhập real-time
* Đăng xuất

**Triển khai**: Singleton pattern với các method signInWithEmail(), signUpWithEmail(), signInAnonymously(), signOut() và stream userChanges để theo dõi trạng thái authentication.

### 3.2. Quản lý dữ liệu nhà hàng (Cloud Firestore)

**File**: repositories/restaurant\_repository.dart

**Chức năng**:

* Lắng nghe danh sách nhà hàng theo thời gian thực
* Thêm đánh giá mới vào subcollection
* Tự động tính toán và cập nhật avgRating sau mỗi đánh giá
* Lưu thông báo về đánh giá mới

**Cơ chế tính avgRating**:

1. Lưu review vào subcollection restaurants/{id}/reviews
2. Đọc tất cả reviews của nhà hàng
3. Tính trung bình cộng các rating
4. Cập nhật field avgRating trong document nhà hàng

**Điểm quan trọng**: Sử dụng Stream để UI tự động cập nhật khi dữ liệu thay đổi, không cần refresh thủ công.

### 3.3. Upload hình ảnh (Firebase Storage + image\_picker)

**File**: services/storage\_service.dart, ui/restaurants/add\_review\_bottom\_sheet.dart

**Quy trình**:

1. Hiển thị modal bottom sheet cho người dùng chọn Camera hoặc Thư viện
2. Sử dụng image\_picker để chụp/chọn ảnh, giới hạn kích thước maxWidth: 1024px
3. Upload ảnh lên Firebase Storage theo đường dẫn: reviews/{restaurantId}/{timestamp}\_{userId}.jpg
4. Lấy URL công khai và lưu vào Firestore

**Xử lý đa nền tảng**: Hỗ trợ cả web (readAsBytes) và mobile (File path) thông qua package kIsWeb.

### 3.4. Thông báo Push (Firebase Cloud Messaging)

**File**: services/fcm\_service.dart, main.dart

**Cấu hình**:

**Foreground**: Hiển thị local notification bằng flutter\_local\_notifications khi app đang mở

**Background/Terminated**: Xử lý bởi background handler được đăng ký trong main.dart

**Topic subscription**: Tất cả người dùng subscribe topic "new\_reviews" để nhận thông báo chung

**Flow hoạt động:**

1. Khởi tạo: Xin quyền thông báo (iOS) và khởi tạo local notifications
2. Lắng nghe: FirebaseMessaging.onMessage stream cho foreground notifications
3. Subscribe: Tự động subscribe topic "new\_reviews" khi khởi động app
4. Hiển thị: Show local notification với title, body và high priority

### 3.5. Seed dữ liệu mẫu

**File**: main.dart

**Chức năng**: Tự động tạo 3 nhà hàng mẫu kèm 1 review nếu database trống.

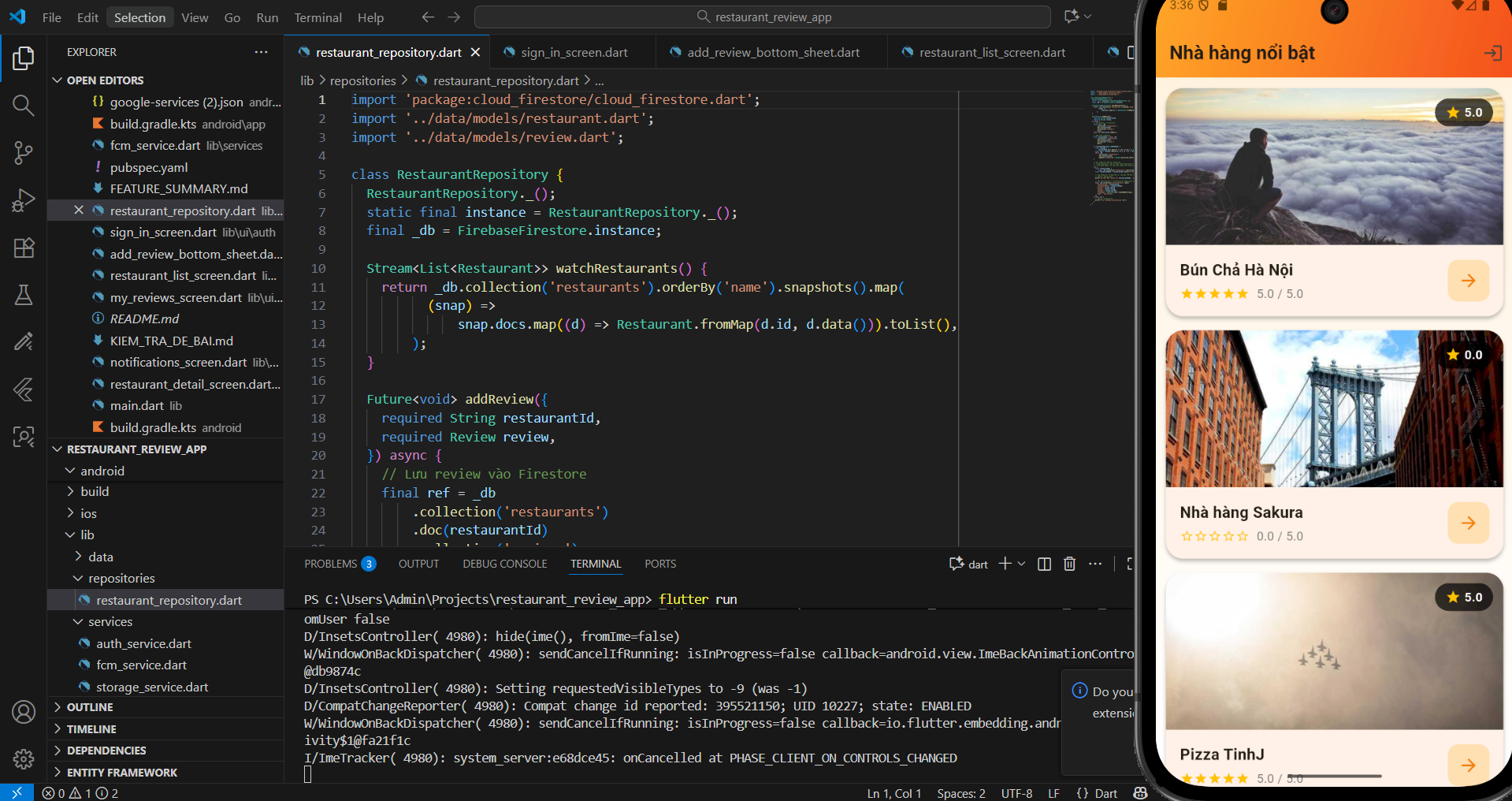
**Quy trình**:

1. Kiểm tra collection restaurants có dữ liệu chưa
2. Nếu trống, sử dụng batch write để tạo 3 nhà hàng với ảnh từ Picsum
3. Tạo 1 review mẫu cho nhà hàng đầu tiên
4. Tính toán và cập nhật avgRating

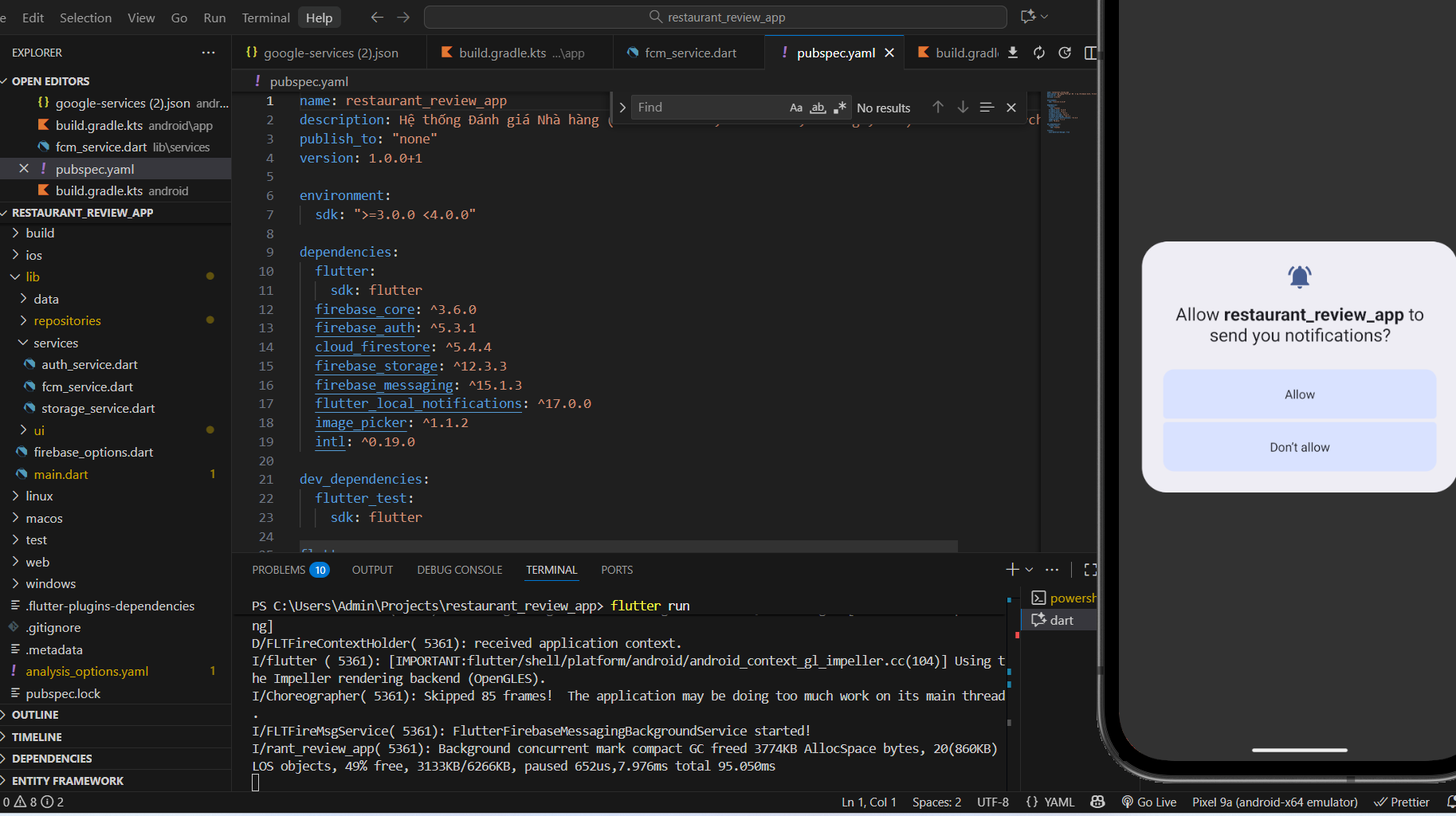
**Ưu điểm**: Giúp demo ứng dụng ngay lập tức mà không cần import data thủ công.

## 4. GIAO DIỆN NGƯỜI DÙNG

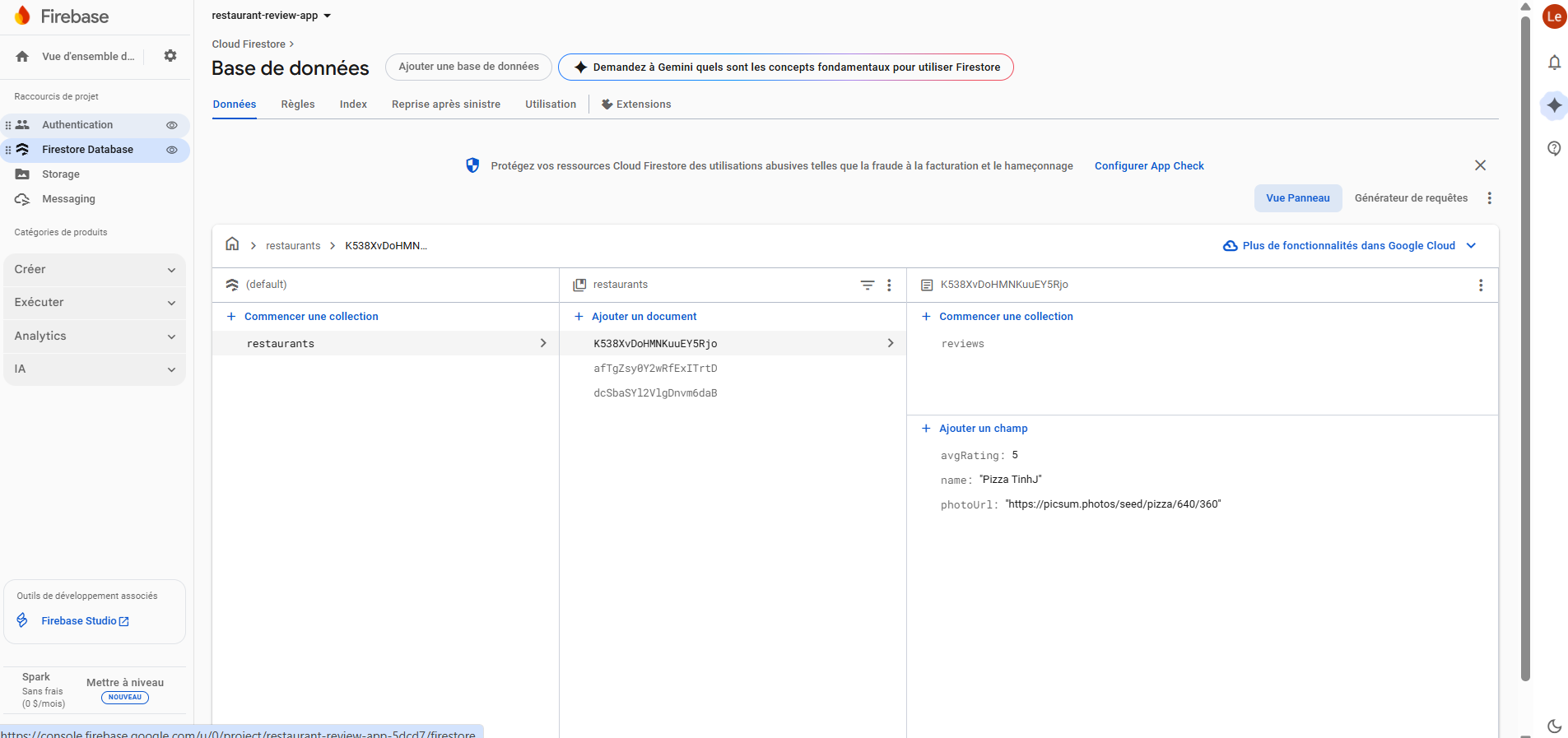
### 4.1. Màn hình danh sách nhà hàng



Gửi thông báo



Frebase lưu trữ

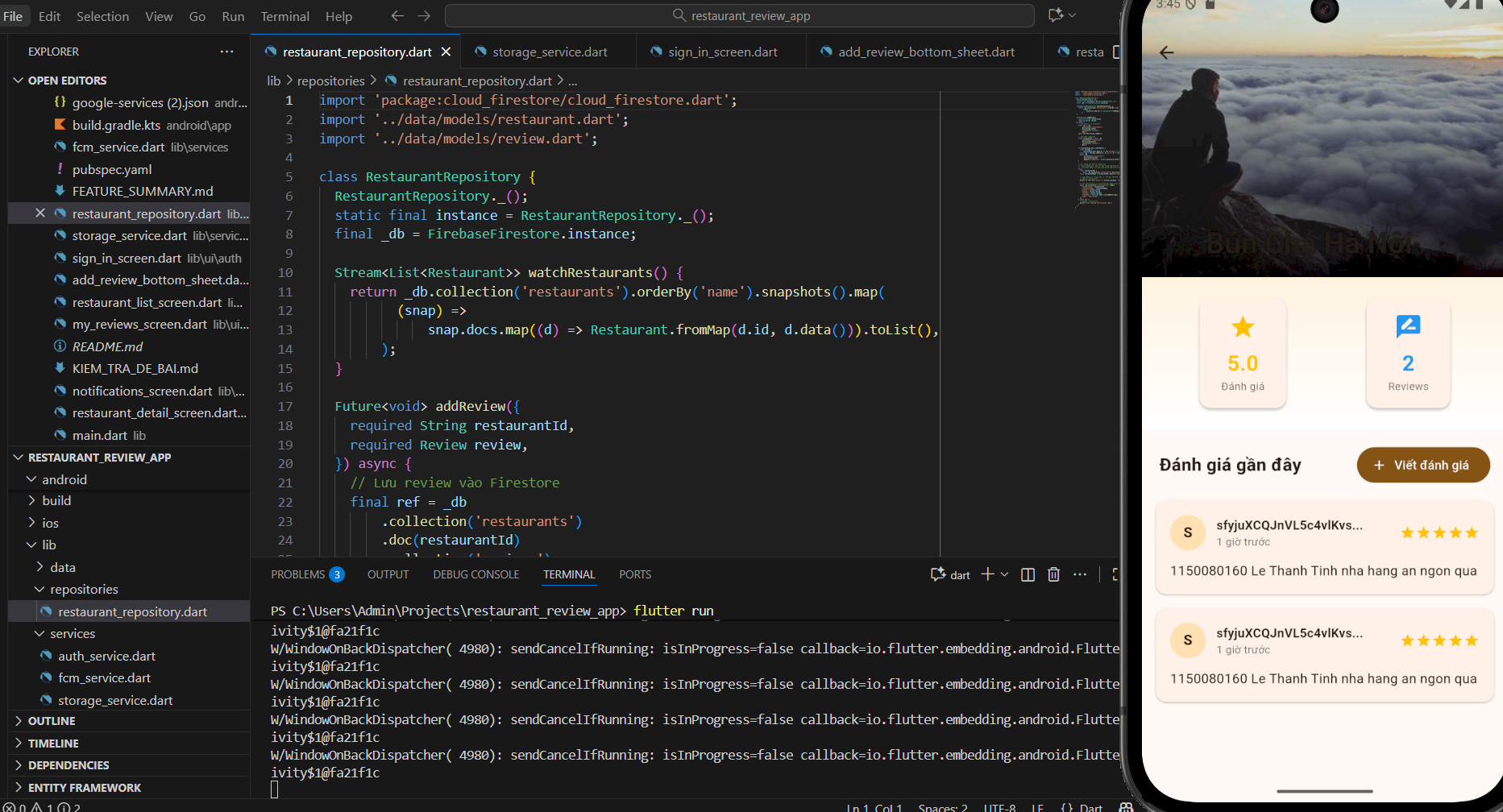


**File**: restaurant\_list\_screen.dart

**Đặc điểm**:

* StreamBuilder để cập nhật danh sách real-time
* Card layout với ảnh nhà hàng, tên, rating
* Hero animation khi chuyển sang màn hình chi tiết
* Badge rating hiển thị trên góc ảnh
* Gradient background màu cam
* Actions: Đăng nhập/xuất, Thông báo, Đánh giá của tôi

### 4.2. Màn hình chi tiết nhà hàng

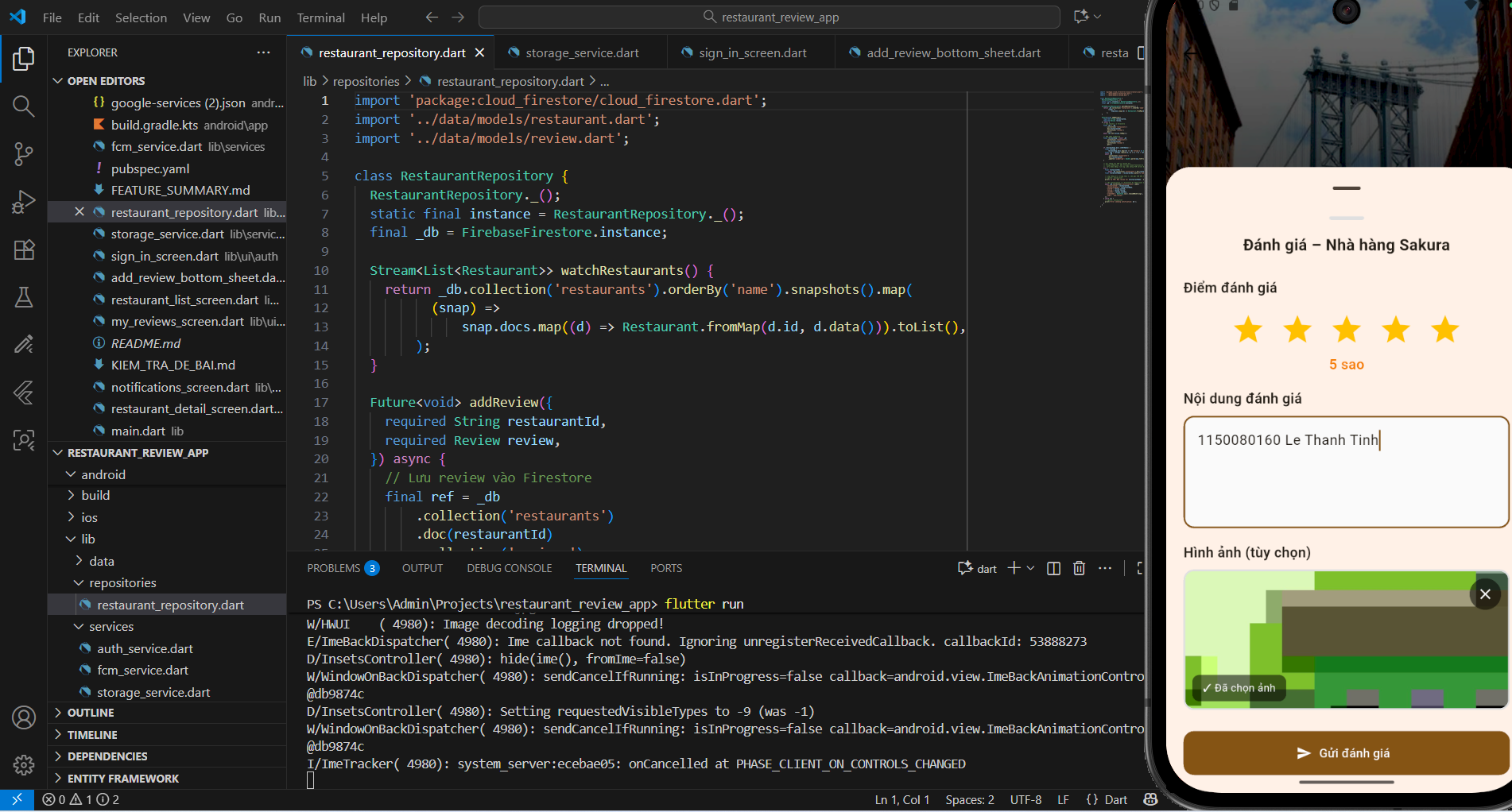


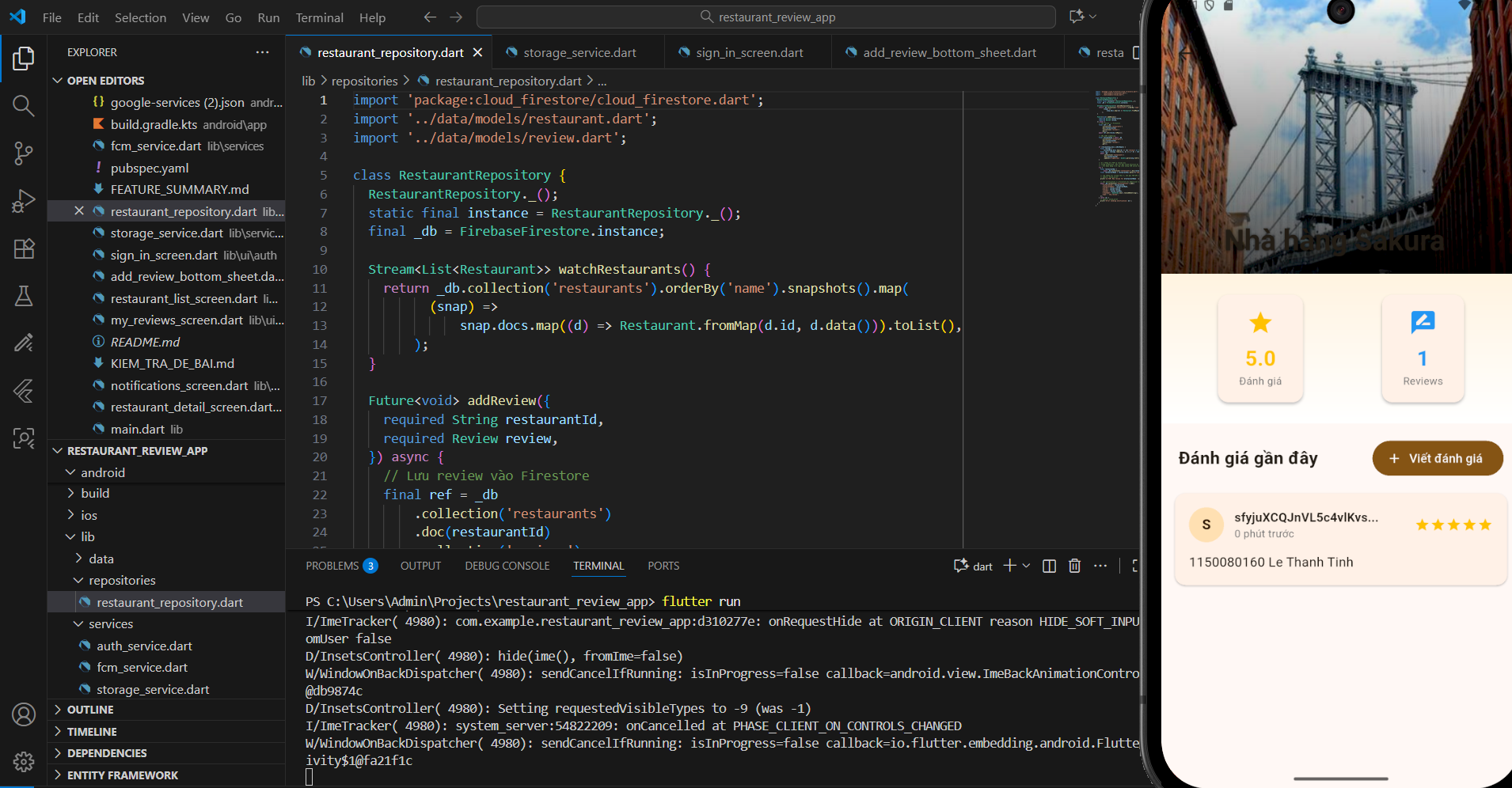
**File**: restaurant\_detail\_screen.dart

**Đặc điểm**:

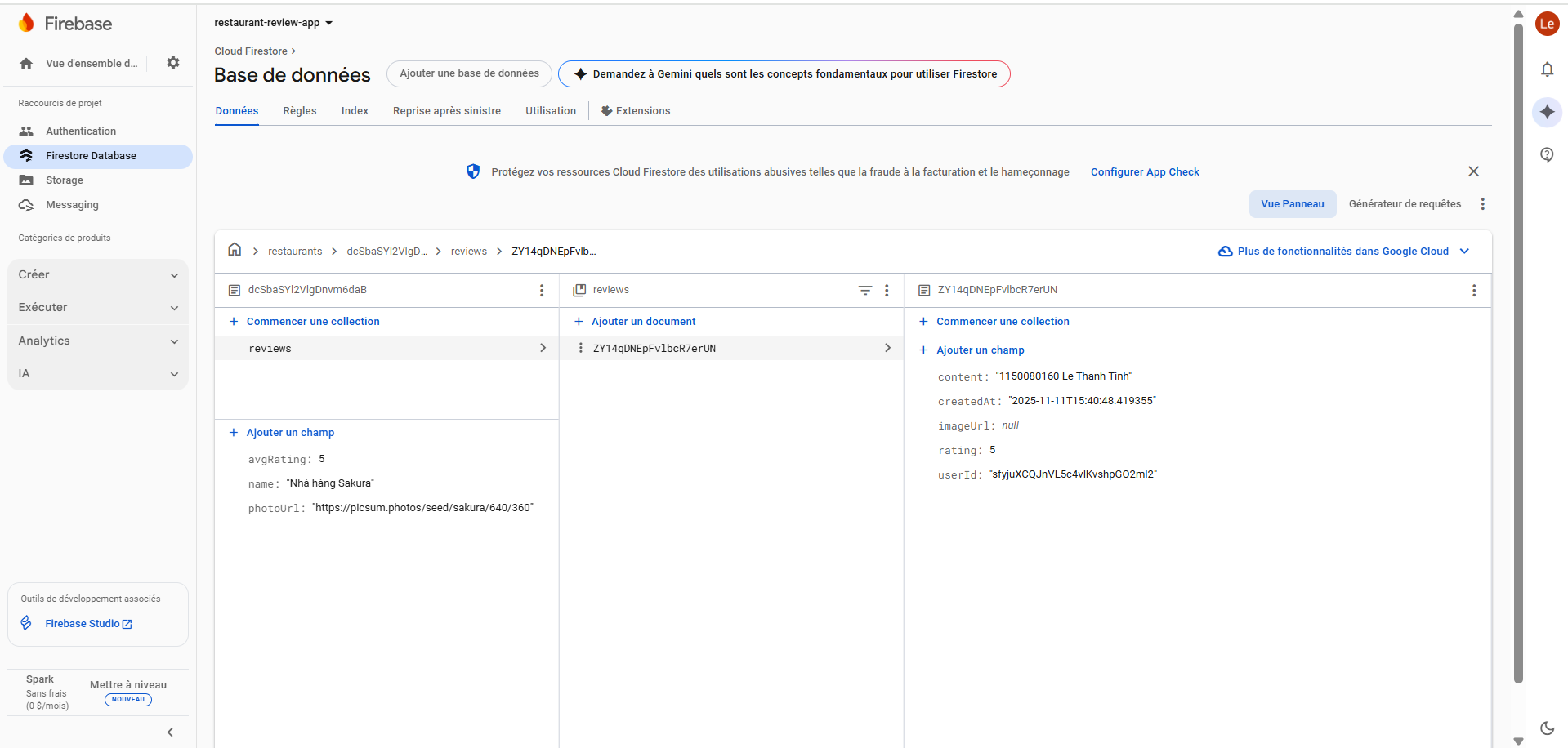
* SliverAppBar với ảnh full-width và gradient overlay
* Expandable header (280px) thu nhỏ khi scroll
* Info cards hiển thị avgRating và số lượng reviews
* StreamBuilder hiển thị danh sách reviews real-time
* Nút "Viết đánh giá" mở bottom sheet

### 4.3. Bottom sheet đánh giá





Firebase lưu trữ



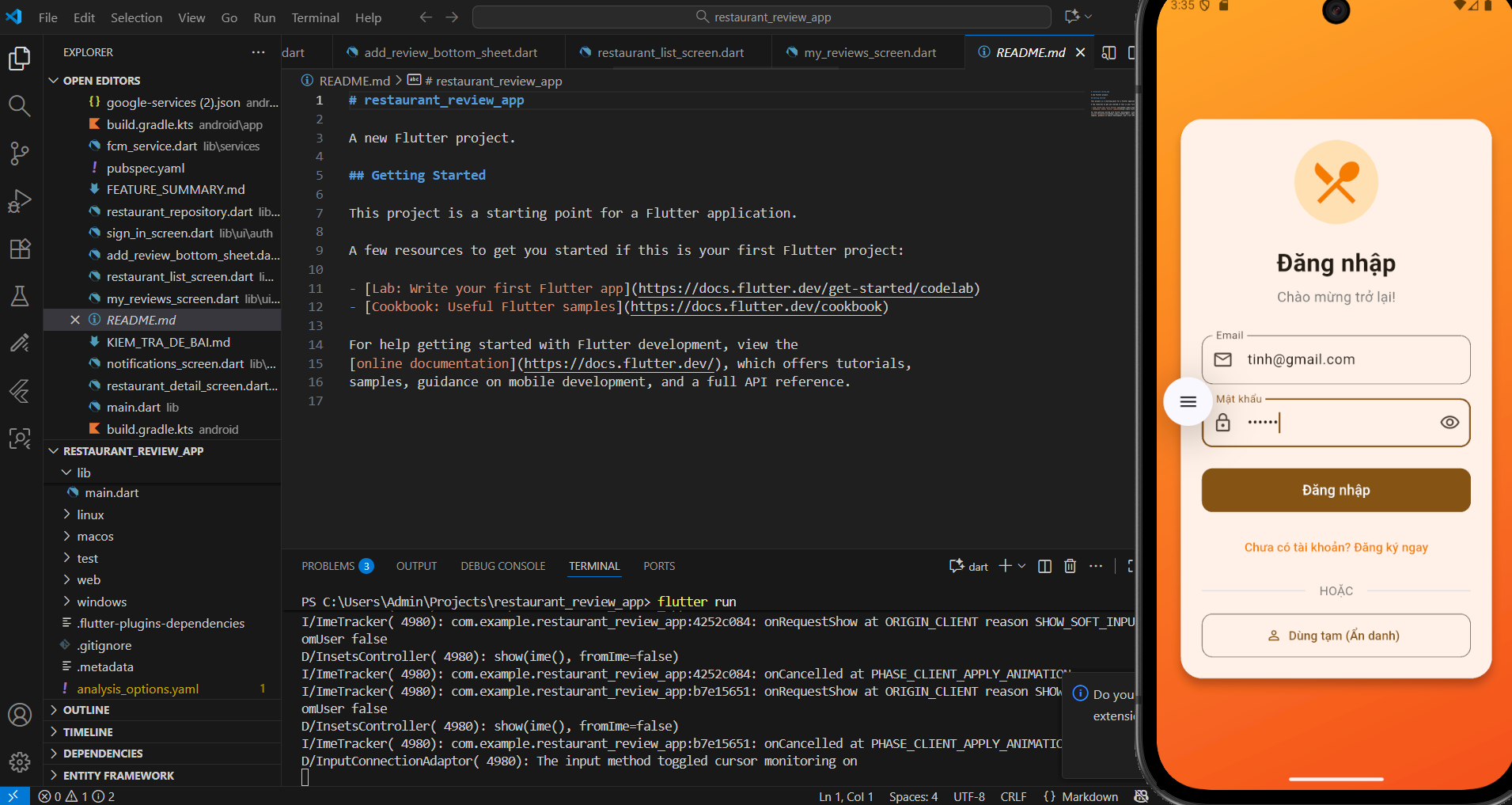
**File**: add\_review\_bottom\_sheet.dart

**Các thành phần**:

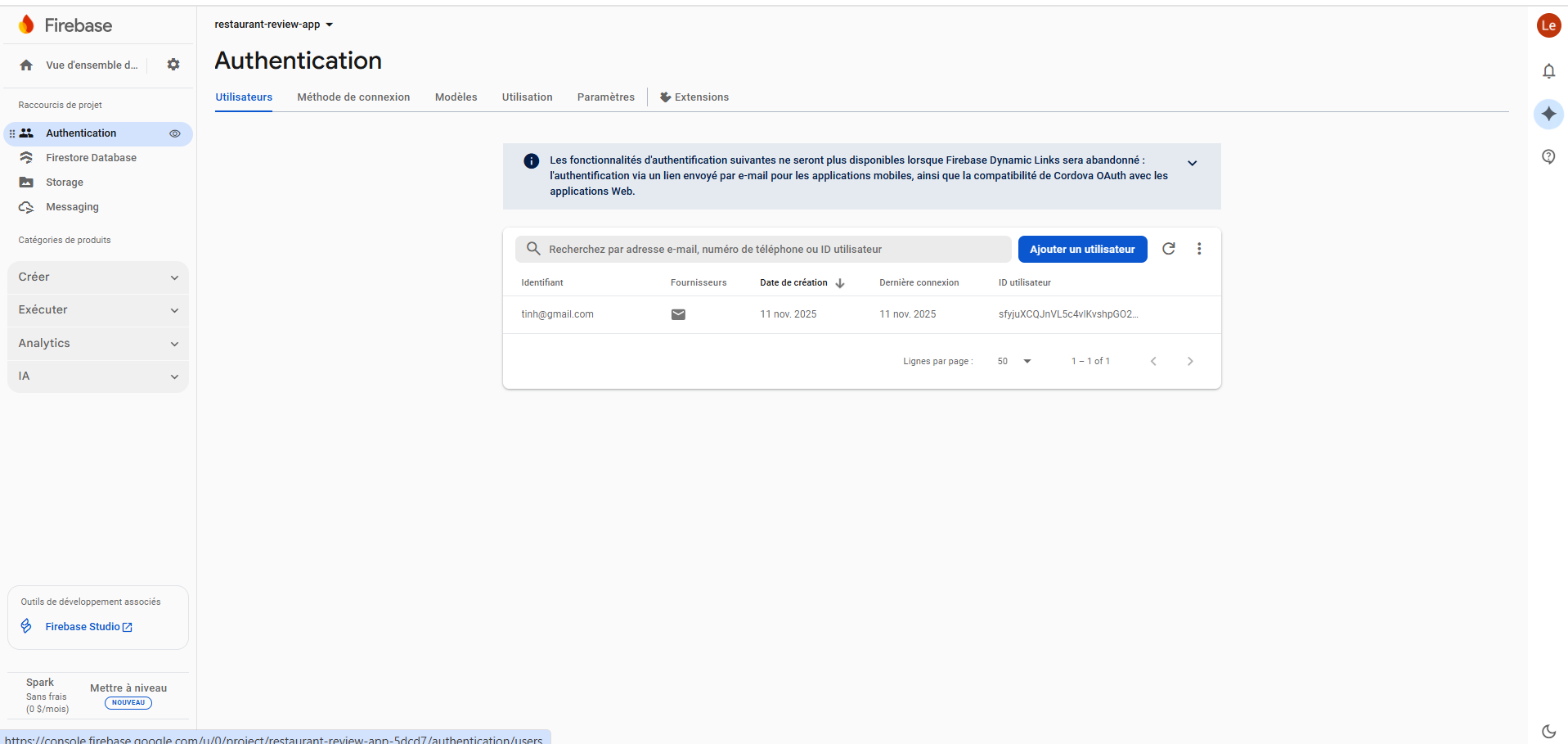
* Star selector: 5 icon stars để chọn rating (1-5)
* Text field đa dòng cho nội dung đánh giá
* Image picker: Chụp ảnh hoặc chọn từ thư viện
* Preview ảnh với nút xóa
* Loading indicator khi đang submit
* Validation: Bắt buộc nhập nội dung

**UX**: isScrollControlled và showDragHandle để bottom sheet responsive với keyboard.

### 4.4. Màn hình đăng nhập



Firebase lưu trữ

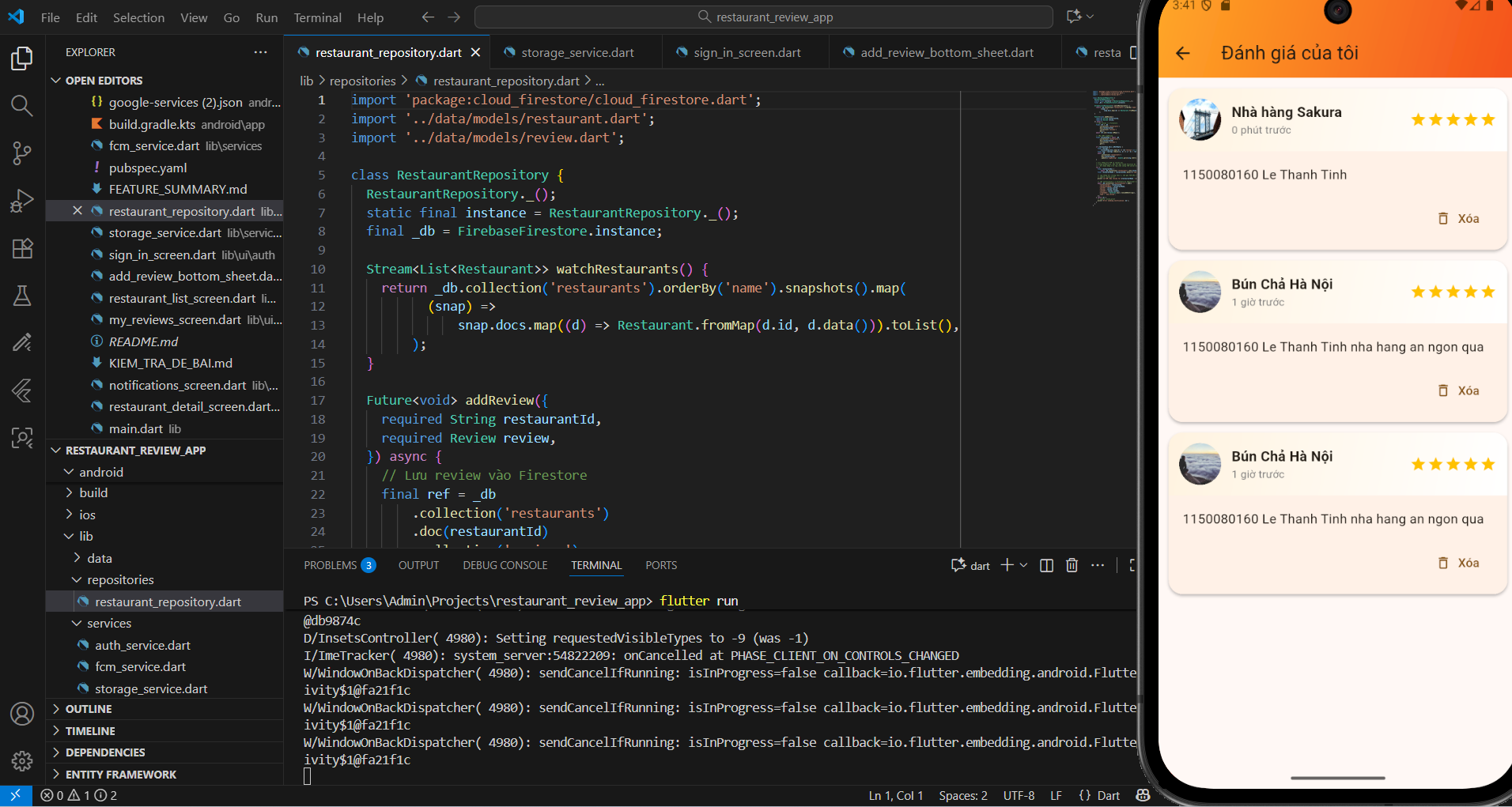


**File**: sign\_in\_screen.dart

**Tính năng**:

* Toggle giữa form đăng nhập và đăng ký
* Validation email và password (min 6 ký tự)
* Show/hide password
* Nút "Đăng nhập ẩn danh" cho trải nghiệm nhanh
* Gradient background và card design hiện đại

### 4.5. Màn hình đánh giá của tôi

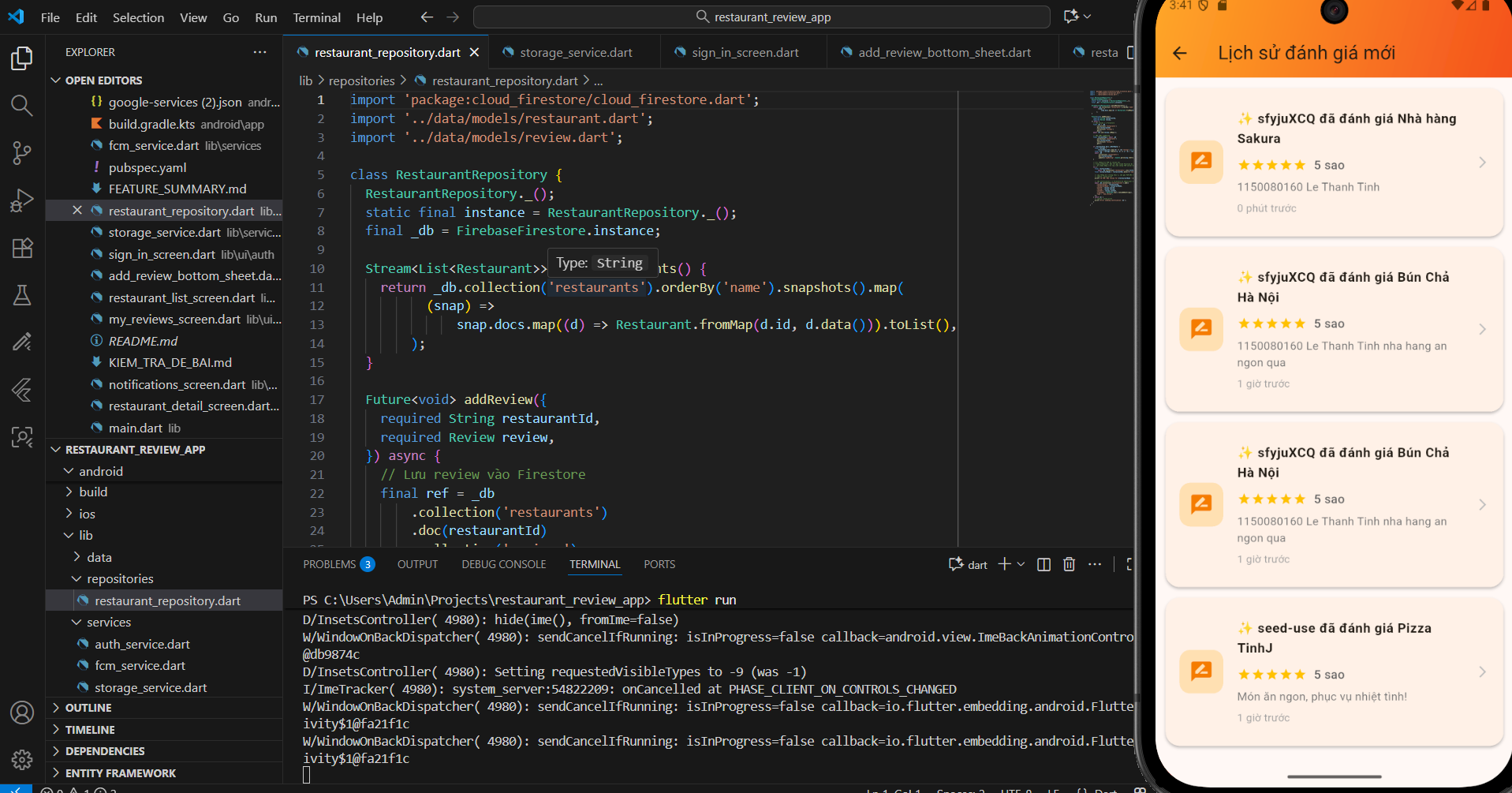


**File**: my\_reviews\_screen.dart

**Chức năng**:

* Hiển thị tất cả đánh giá của user hiện tại từ mọi nhà hàng
* Load data bằng FutureBuilder (query reviews theo userId)
* Sắp xếp theo thời gian giảm dần
* Chức năng xóa đánh giá với dialog xác nhận
* Tự động cập nhật avgRating sau khi xóa
* Pull-to-refresh để tải lại danh sách

### 4.6. Màn hình lịch sử đánh giá



**File**: notifications\_screen.dart

**Chức năng**:

* Hiển thị 20 đánh giá mới nhất từ tất cả nhà hàng
* Query tất cả subcollections reviews và merge
* Hiển thị tên nhà hàng, user, rating, nội dung, thời gian
* Format thời gian relative (X phút/giờ/ngày trước)
* Icon và styling phù hợp với notification

## 5. ĐIỂM NỔI BẬT VÀ KỸ THUẬT

### 5.1. Real-time Updates

Sử dụng Stream và StreamBuilder để UI tự động cập nhật khi dữ liệu Firestore thay đổi, không cần setState() thủ công.

### 5.2. Optimized Queries

Subcollection để tổ chức reviews theo nhà hàng

OrderBy và limit để tối ưu performance

Batch write khi seed data

### 5.3. Error Handling

Try-catch cho tất cả Firebase operations

Error builder cho Image.network

Loading states với CircularProgressIndicator

### 5.4. Responsive Design

MediaQuery để lấy keyboard height cho bottom sheet

SingleChildScrollView để tránh overflow

Gradient và shadow cho UI đẹp mắt

### 5.5. State Management

Singleton pattern cho services

StreamController trong repository

setState() đơn giản cho local state

## 6. KẾT LUẬN

Ứng dụng hoàn chỉnh với đầy đủ CRUD operations

Tích hợp 4 Firebase services chính

UI/UX hiện đại với Material Design 3

Real-time updates và push notifications