CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

XÂY DỰNG ỨNG DỤNG MYSHOP - PHẦN 1

File báo cáo cần nộp là file PDF trong đó có ghi thông tin **mã sinh viên, họ tên, lớp học phần** cùng với **hình minh họa tại các bước kiểm tra kết quả thực thi, các chức năng (không cần chụp hình mã nguồn)**. Cuối file báo cáo ghi đường link đến GitHub mã nguồn của dự án mà bạn tạo.

Ứng dụng MyShop có các chức năng chính sau:

- Hiển thị và cập nhật danh mục các sản phẩm
- Xem chi tiết một sản phẩm
- Đánh dấu sản phẩm được yêu thích
- Thêm, xóa sản phẩm trong giỏ hàng
- Thực hiện đặt hàng, xem lại đơn hàng, chi tiết đơn hàng
- Đăng ký, đăng nhập
- Lưu trữ dữ liệu trên Firebase

Sinh viên chỉ cần tạo MỘT báo cáo duy nhất cho tất cả các buổi thực hành. Nộp báo cáo khi đến hạn của mỗi buổi để báo cáo tiến độ, không tính điểm. Chỉ bài báo cáo nộp vào buổi thực hành cuối cùng sẽ được chấm điểm. Tuy nhiên, không nộp bài báo cáo tiến độ thì không tính điểm buổi đó.

Bước 0: Chuẩn bị môi trường làm việc

- Cài đặt Flutter, xem hướng dẫn tại: https://docs.flutter.dev/get-started/install. Cài đầy đủ Android Studio, Android SDK, Android SDK Command-line Tools, và Android SDK Build-Tools.
- Kiểm tra cài đặt môi trường thành công, vào Terminal/Command Prompt, chạy lệnh `flutter doctor`:

```
D:\> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.0.1, on Microsoft Windows [Version 10.0.22000.708], locale vi-VN)
[✓] Android toolchain - develop for Android devices (Android SDK version 32.1.0-rc1)
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.1.1)
[✓] Android Studio (version 2021.2)
[✓] VS Code (version 1.68.0)
[✓] Connected device (1 available)
[✓] HTTP Host Availability

• No issues found!
```

  Kiểm tra rằng **Flutter** và **Android toolchain** không có vấn đề (dấu check xanh).
- Mở *Android Studio* > *Virtual Device Manager* để tạo một máy ảo di động.
- Tải và cài đặt git: https://git-scm.com/download/win.
- Kiểm tra rằng có thể gõ lệnh `node` và `git` trên Terminal/Command Prompt:

```
D:\> git --version
git version 2.35.2.windows.1
```

(Trong trường hợp không thể chạy lệnh từ Terminal thì cần thêm đường dẫn đến thư mục chứa chương trình `git` vào biến môi trường PATH trên máy của bạn).

- Trình soạn thảo mã nguồn (IDE): có thể dùng Android Studio hoặc Visual Studio Code (VS Code). Nếu sử dụng VS Code thì cần cài đặt thêm phần mở rộng cho Dart và Flutter.

Bước 1: Khởi tạo dự án

- Để tạo dự án flutter, có thể sử dụng lệnh flutter trên Terminal: `flutter create myshop`.
- Dùng VS Code, mở dự án vừa tạo, hiệu chỉnh nội dung tập tin *lib/main.dart* như sau:

```dart
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'My Shop',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        fontFamily: 'Lato',
        colorScheme: ColorScheme.fromSwatch(
          primarySwatch: Colors.purple,
        ).copyWith(
          secondary: Colors.deepOrange,
        ),
      ),
      home: Container(
        color: Colors.green,
      ),
    );
  }
}
```

- Sao chép thư mục *assets* đã cho vào thư mục gốc của dự án. Hiệu chỉnh tập tin *pubspec.yaml*, khai báo sử dụng các font trong thư mục *assets*:

```
flutter:
  uses-material-design: true
  fonts:
    - family: Lato
      fonts:
        - asset: assets/fonts/Lato-Regular.ttf
        - asset: assets/fonts/Lato-Bold.ttf
          weight: 700
    - family: Anton
      fonts:
        - asset: assets/fonts/Anton-Regular.ttf
```

- Mở thiết bị giả lập sau đó tiến hành chạy ứng dụng: Run > Run Without Debugging. Kiểm tra ứng dụng chạy thành công.
- Tiếp đến, khởi tạo repo git để quản lý mã nguồn dự án: mở Terminal tại thư mục gốc của dự án và chạy lệnh `git init`. Gõ `git status` sẽ thấy các thư mục, tập tin của dự án được hiển thị. Lưu các thư mục, tập tin này vào cho git quản lý:

```
git add -A
git commit -m "Khoi tao du an"
```

- Đăng ký tài khoản (nếu chưa có) và tạo một repo trên GitHub. Liên kết repo git cục bộ với repo git trên GitHub như sau:

```
git remote add origin <github-repository-url>
git push origin master
```

Kiểm tra rằng các tập tin mã nguồn dự án được đưa lên repo trên GitHub.

Bước 2: Xây dựng trang hiển thị thông tin chi tiết sản phẩm

- Định nghĩa lớp **Product** miêu tả thông tin của một sản phẩm (*lib/models/product.dart*):

```dart
class Product {
  final String? id;
  final String title;
  final String description;
  final double price;
  final String imageUrl;
  final bool isFavorite;

  Product({
    this.id,
    required this.title,
    required this.description,
    required this.price,
    required this.imageUrl,
    this.isFavorite = false,
  });

  Product copyWith({
    String? id,
    String? title,
    String? description,
    double? price,
    String? imageUrl,
    bool? isFavorite,
  }) {
    return Product(
      id: id ?? this.id,
      title: title ?? this.title,
      description: description ?? this.description,
      price: price ?? this.price,
      imageUrl: imageUrl ?? this.imageUrl,
      isFavorite: isFavorite ?? this.isFavorite,
    );
  }
}

```

- Danh sách các sản phẩm ví dụ (*_items*) được cho trong tập tin *products.txt*. Định nghĩa lớp **ProductsManager** quản lý các sản phẩm (*lib/ui/products/products_manager.dart*):

```dart
import '../../models/product.dart';

class ProductsManager {
  final List<Product> _items = […

  int get itemCount {
    return _items.length;
  }

  List<Product> get items {
    return [..._items];
  }

  List<Product> get favoriteItems {
    return _items.where((prodItem) => prodItem.isFavorite).toList();
  }
}
```

- Định nghĩa trang thông tin chi tiết sản phẩm (*lib/ui/products/product_detail_screen.dart*):

```dart
import 'package:flutter/material.dart';

import '../../models/product.dart';

class ProductDetailScreen extends StatelessWidget {
  const ProductDetailScreen(
    this.product, {
    super.key,
  });

  final Product product;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(product.title),
      ),
      body: SingleChildScrollView(
        child: Column(
          children: <Widget>[
            SizedBox(
              height: 300,
              width: double.infinity,
              child: Image.network(
                product.imageUrl,
                fit: BoxFit.cover,
              ),
            ),
            const SizedBox(height: 10),
            Text(
              '\$${product.price}',
              style: const TextStyle(
                color: Colors.grey,
                fontSize: 20,
              ),
            ),
            const SizedBox(
              height: 10,
            ),
            Container(
              padding: const EdgeInsets.symmetric(horizontal: 10),
              width: double.infinity,
              child: Text(
                product.description,
                textAlign: TextAlign.center,
                softWrap: true,
              ),
            )
          ],
        ),
      ),
    );
  }
}

```

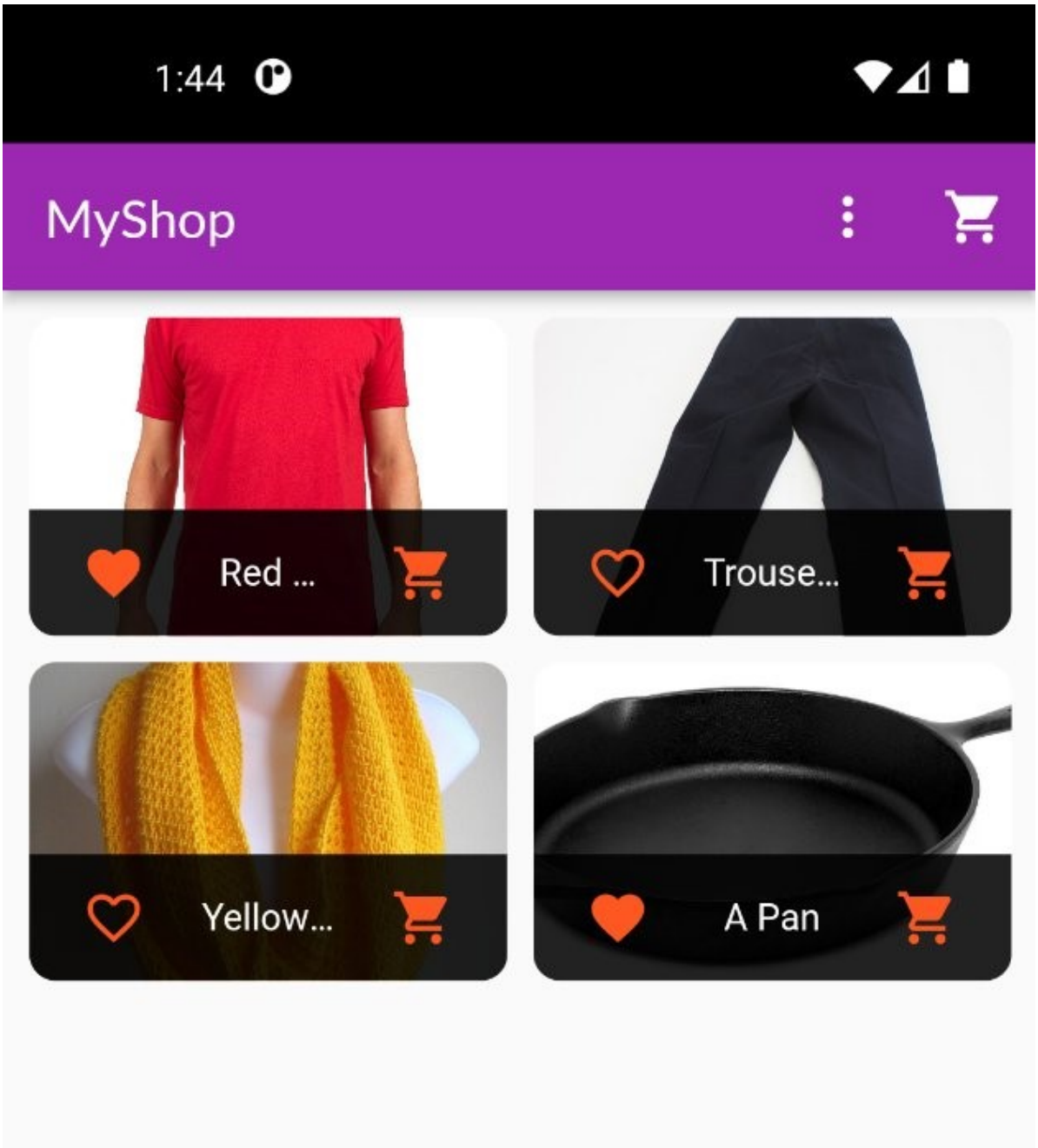
- Hiệu chỉnh *lib/main.dart* để kiểm tra trang hiển thị thông tin chi tiết sản phẩm:

```
...
import 'ui/products/products_manager.dart';
import 'ui/products/product_detail_screen.dart';
...
class MyApp extends StatelessWidget {
    ...
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            ...
            home: SafeArea(
                child: ProductDetailScreen(
                    ProductsManager().items[0],
                ),
            ),
        );
    }
}
```

- Sau khi kiểm tra, lưu mã lệnh vào repo git và lên GitHub:

```
git add -u
git add lib/models lib/ui
git commit -m "Xay dung trang chi tiet san pham"
git push origin master
```

Bước 3: Xây dựng trang tổng quan các sản phẩm



- Định nghĩa widget **ProductGridTile** trình bày thông tin một sản phẩm (*lib/ui/products/product_grid_tile.dart*):

```dart
import 'package:flutter/material.dart';

import '../../models/product.dart';

import 'product_detail_screen.dart';

class ProductGridTile extends StatelessWidget {
  const ProductGridTile(
    this.product, {
    super.key,
  });

  final Product product;

  @override
  Widget build(BuildContext context) {
    return ClipRRect(
      borderRadius: BorderRadius.circular(10),
      child: GridTile(
        footer: buildGridFooterBar(context),
        child: GestureDetector(
          onTap: () {
            print('Go to product detail screen');
          },
          child: Image.network(
            product.imageUrl,
            fit: BoxFit.cover,
          ),
        ),
      ),
    );
  }

  Widget buildGridFooterBar(BuildContext context) {
    return GridTileBar(
      ...
    );
  }
}
```

Hàm *buildGridFooterBar*():

```
Widget buildGridFooterBar(BuildContext context) {
  return GridTileBar(
    backgroundColor: Colors.black87,
    leading: IconButton(
      icon: Icon(
        product.isFavorite ? Icons.favorite : Icons.favorite_border,
      ),
      color: Theme.of(context).colorScheme.secondary,
      onPressed: () {
        print('Toggle a favorite product');
      },
    ),
    title: Text(
      product.title,
      textAlign: TextAlign.center,
    ),
    trailing: IconButton(
      icon: const Icon(
        Icons.shopping_cart,
      ),
      onPressed: () {
        print('Add item to cart');
      },
      color: Theme.of(context).colorScheme.secondary,
    ),
  );
}
```

- Định nghĩa widget **ProductsGrid** hiển thị các sản phẩm dạng lưới (*lib/ui/products/products_grid.dart*):

```dart
import 'package:flutter/material.dart';

import 'product_grid_tile.dart';
import 'products_manager.dart';

class ProductsGrid extends StatelessWidget {
  final bool showFavorites;

  const ProductsGrid(this.showFavorites, {super.key});

  @override
  Widget build(BuildContext context) {
    final productsManager = ProductsManager();
    final products =
        showFavorites ? productsManager.favoriteItems : productsManager.items;
    return GridView.builder(
      padding: const EdgeInsets.all(10.0),
      itemCount: products.length,
      itemBuilder: (ctx, i) => ProductGridTile(products[i]),
      gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
        childAspectRatio: 3 / 2,
        crossAxisSpacing: 10,
        mainAxisSpacing: 10,
      ),
    );
  }
}

```

- Định nghĩa trang tổng quan các sản phẩm (*lib/ui/products/product_overview_screen.dart*):

```dart
import 'package:flutter/material.dart';

import 'products_grid.dart';

enum FilterOptions { favorites, all }

class ProductsOverviewScreen extends StatefulWidget {
  const ProductsOverviewScreen({super.key});

  @override
  State<ProductsOverviewScreen> createState() => _ProductsOverviewScreenState();
}

class _ProductsOverviewScreenState extends State<ProductsOverviewScreen> {
  var _showOnlyFavorites = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('MyShop'),
        actions: <Widget>[
          buildProductFilterMenu(),
          buildShoppingCartIcon(),
        ],
      ),
      body: ProductsGrid(_showOnlyFavorites),
    );
  }

  Widget buildShoppingCartIcon() {
    return IconButton(
      ...
    );
  }

  Widget buildProductFilterMenu() {
    return PopupMenuButton(
      ...
    );
  }
}
```

Hàm *buildShoppingCartIcon()* và *buildProductFilterMenu()*:

```
Widget buildShoppingCartIcon() {
  return IconButton(
    icon: const Icon(
      Icons.shopping_cart,
    ),
    onPressed: () {
      print('Go to cart screen');
    },
  );
}

Widget buildProductFilterMenu() {
  return PopupMenuButton(
    onSelected: (FilterOptions selectedValue) {
      setState(() {
        if (selectedValue == FilterOptions.favorites) {
          _showOnlyFavorites = true;
        } else {
          _showOnlyFavorites = false;
        }
      });
    },
    icon: const Icon(
      Icons.more_vert,
    ),
    itemBuilder: (ctx) => [
      const PopupMenuItem(
        value: FilterOptions.favorites,
        child: Text('Only Favorites'),
      ),
      const PopupMenuItem(
        value: FilterOptions.all,
        child: Text('Show All'),
      ),
    ],
  );
}
```

- Hiệu chỉnh *lib/main.dart* kiểm tra trang tổng quan các sản phẩm:

```
...
import 'ui/products/products_overview_screen.dart';
...
class MyApp extends StatelessWidget {
    ...
    @override
```

```
    Widget build(BuildContext context) {
        return MaterialApp(
            ...
            home: const SafeArea(
                child: ProductsOverviewScreen(),
            ),
        );
    }
}
```

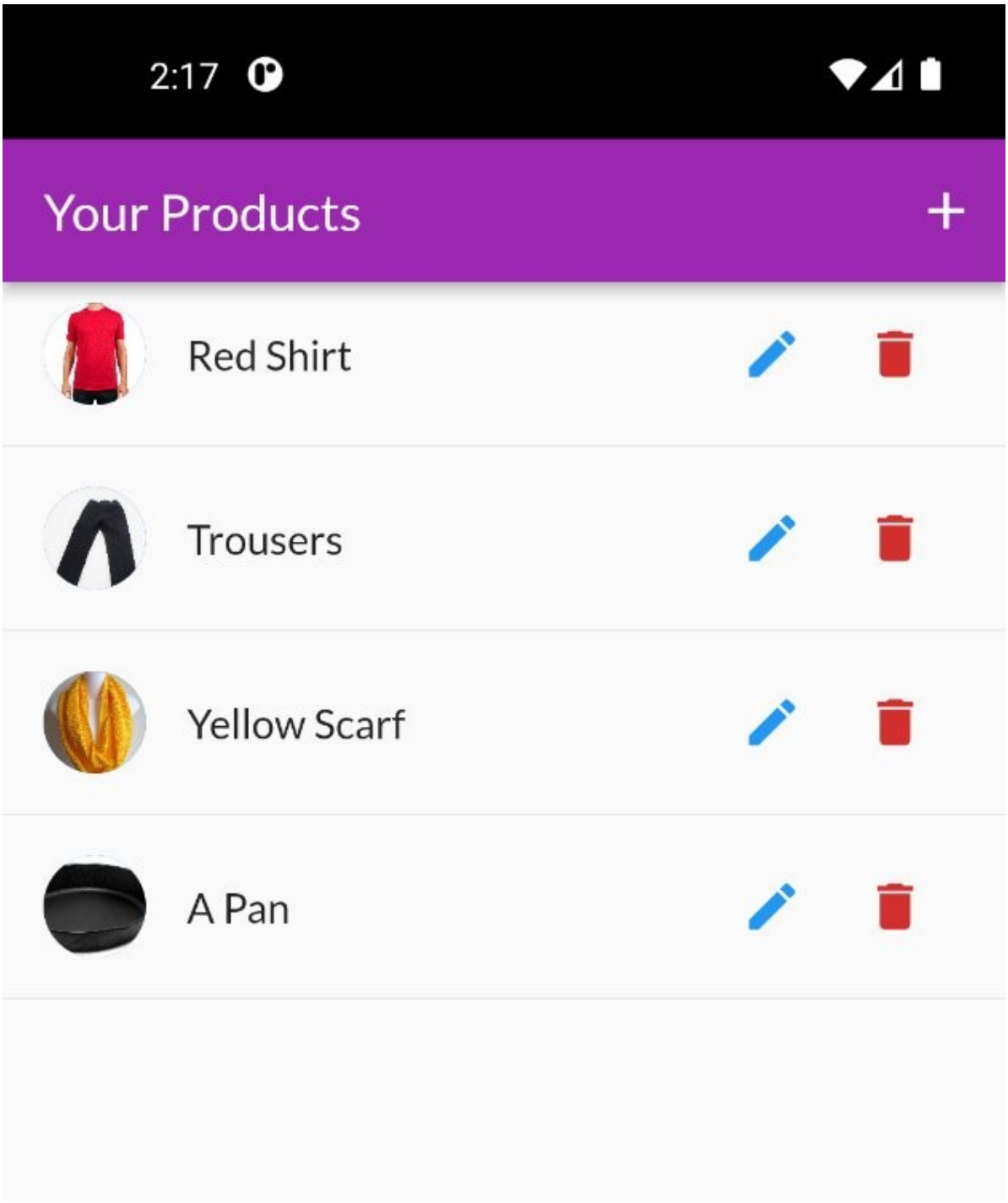
- Hiệu chỉnh widget **ProductGridTile** (*lib/ui/products/product_grid_tile.dart*) để thực hiện liên kết đến trang thông tin chi tiết sản phẩm:

```
...
import 'product_detail_screen.dart';
...
child: GestureDetector(
    onTap: () {
        Navigator.of(context).push(
            MaterialPageRoute(
                builder: (ctx) => ProductDetailScreen(product),
            ),
        );
    },
    child: Image.network(
        product.imageUrl,
        fit: BoxFit.cover,
    ),
),
...
```

- Sau khi kiểm tra, lưu mã lệnh vào repo git và lên GitHub:

```
git add -u
git add lib/ui
git commit -m "Xay dung trang tong quan cac san pham"
git push origin master
```

Bước 4: Xây dựng trang các sản phẩm của người dùng



- Định nghĩa widget **UserProductListTile** hiển thị thông tin một sản phẩm cùng với các thao tác sửa/xóa (*lib/ui/products/user_product_list_tile.dart*):

```dart
import 'package:flutter/material.dart';

import '../../models/product.dart';

class UserProductListTile extends StatelessWidget {
  final Product product;

  const UserProductListTile(
    this.product, {
    super.key,
  });

  @override
  Widget build(BuildContext context) {
    return ListTile(
      title: Text(product.title),
      leading: CircleAvatar(
        backgroundImage: NetworkImage(product.imageUrl),
      ),
      trailing: SizedBox(
        width: 100,
        child: Row(
          children: <Widget>[
            buildEditButton(context),
            buildDeleteButton(context),
          ],
        ),
      ),
    );
  }

  Widget buildDeleteButton(BuildContext context) {
    return IconButton(
      ...
    );
  }

  Widget buildEditButton(BuildContext context) {
    return IconButton(
      ...
    );
  }
}
```

Hàm *buildDeleteButton*() và *buildEditButton*():

```
Widget buildDeleteButton(BuildContext context) {
  return IconButton(
    icon: const Icon(Icons.delete),
    onPressed: () async {
      print('Delete a product');
    },
    color: Theme.of(context).errorColor,
  );
}

Widget buildEditButton(BuildContext context) {
  return IconButton(
    icon: const Icon(Icons.edit),
    onPressed: () {
      print('Go to edit product screen');
    },
    color: Theme.of(context).primaryColor,
  );
}
```

- Định nghĩa trang hiển thị các sản phẩm của người dùng (*lib/ui/products/user_products_screen.dart*):

```dart
import 'package:flutter/material.dart';

import 'user_product_list_tile.dart';
import 'products_manager.dart';

class UserProductsScreen extends StatelessWidget {
  const UserProductsScreen({super.key});

  @override
  Widget build(BuildContext context) {
    final productsManager = ProductsManager();
    return Scaffold(
      appBar: AppBar(
        title: const Text('Your Products'),
        actions: <Widget>[
          buildAddButton(),
        ],
      ),
      body: RefreshIndicator(
        onRefresh: () async => print('refresh products'),
        child: buildUserProductListView(productsManager),
      ),
    );
  }

  Widget buildUserProductListView(ProductsManager productsManager) {
    return ListView.builder(
      ...
    );
  }

  Widget buildAddButton() {
    return IconButton(
      ...
    );
  }
}
```

Hàm *buildUserProductListView*() và *buildAddButton*():

```
Widget buildUserProductListView(ProductsManager productsManager) {
  return ListView.builder(
    itemCount: productsManager.itemCount,
    itemBuilder: (ctx, i) => Column(
      children: [
        UserProductListTile(
          productsManager.items[i],
        ),
        const Divider(),
      ],
    ),
  );
}

Widget buildAddButton() {
  return IconButton(
    icon: const Icon(Icons.add),
    onPressed: () {
      print('Go to edit product screen');
    },
  );
}
```

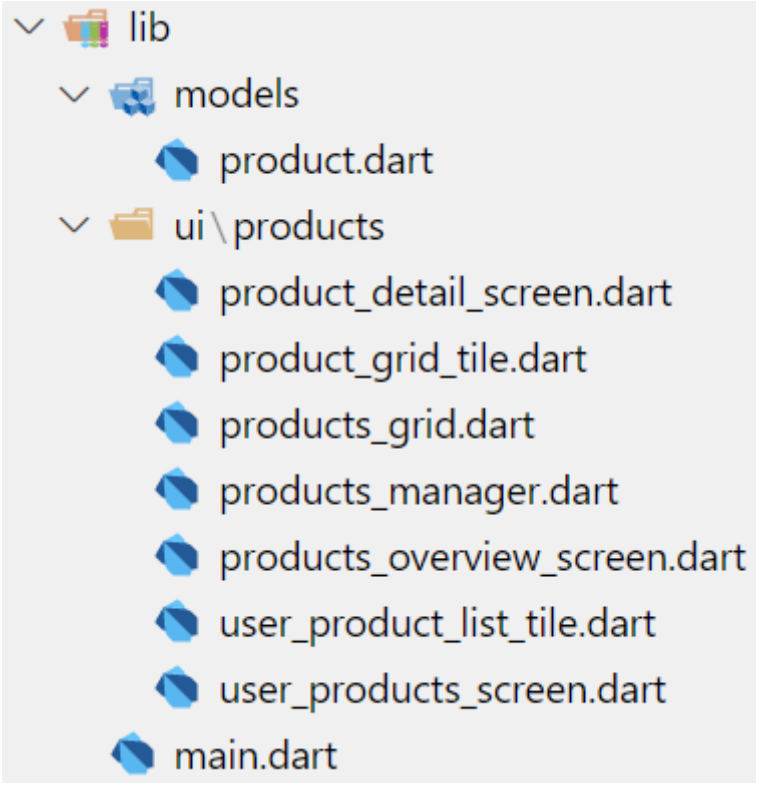
- Hiệu chỉnh *lib/main.dart* kiểm tra trang các sản phẩm của người dùng:

```
...
import 'ui/products/user_products_screen.dart';
...
class MyApp extends StatelessWidget {
    ...
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            ...
            home: const SafeArea(
                child: UserProductsScreen(),
            ),
        );
    }
}
```

- Sau khi kiểm tra, lưu mã lệnh vào repo git và lên GitHub:

```
git add -u
git add lib/ui
git commit -m "Xay dung trang cac san pham nguoi dung"
git push origin master
```

Cấu trúc thư mục mã nguồn hiện tại như sau:


lib
models
product.dart
ui\products
product_detail_screen.dart
product_grid_tile.dart
products_grid.dart
products_manager.dart
products_overview_screen.dart
user_product_list_tile.dart
user_products_screen.dart
main.dart