

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO DỰ ÁN CUỐI KỲ
PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG - CT484**

**Đề tài
ỨNG DỤNG BÁN MỸ PHẨM - LA BEAUTÉ**

Lớp học phần: CT48403

Nhóm sinh viên thực hiện:

Họ và tên: Nguyễn Lê Sắc

MSSV: B2014605

Họ và tên: Nguyễn Thu Thảo

MSSV: B2014613

HỌC KỲ 2, NĂM HỌC 2023-2024

MỤC LỤC

I. Tổng quan	3
II. Chi tiết các chức năng	5
1. Chức năng/giao diện 1: Đăng nhập, đăng ký	5
2. Chức năng/giao diện 2: Giao diện trang chủ	7
3. Chức năng/giao diện 3: Giỏ hàng	10
4. Chức năng/giao diện 4: Thông tin người dùng	13
5. Chức năng/giao diện 5: Quản lý đơn đặt hàng	15
6. Chức năng/giao diện 6: Hiển thị chi tiết đơn đặt hàng	17
7. Chức năng/giao diện 7: Hiển thị chi tiết sản phẩm	20
8. Chức năng/giao diện 8: Hiển thị sản phẩm yêu thích	23
9. Chức năng/giao diện 9: Giao diện quản trị viên	26
10. Chức năng/giao diện 10: Quản lý sản phẩm	27
Link Github	29

BÁO CÁO DỰ ÁN CUỐI KỲ

HỌC KỲ 2, NĂM HỌC 2023-2024

CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

- Tên dự án/ứng dụng: La Beauté
- Link GitHub mã nguồn:
<https://github.com/23-24Sem2-Courses/ct48403-project-lesacIT.git>
- MSSV 1: B2014605
- Họ tên SV 1: Nguyễn Lê Sắc
- MSSV 2: B204613
- Họ tên SV 2: Nguyễn Thu Thảo
- Lớp học phần: CT48403

I. Tổng quan

- **Miêu tả dự án/ứng dụng:** La Beauté là ứng dụng di động cho phép người dùng đăng kí một tài khoản và đăng nhập. Ứng dụng bán các sản phẩm là các loại mỹ phẩm, cho phép người dùng thêm vào giỏ hàng và đặt hàng. Cũng như hiển thị các đơn hàng mà người dùng đã đặt.

- *Bảng phân công công việc*

Người thực hiện	Tên công việc
Nguyễn Lê Sắc	<ul style="list-style-type: none">- Xây dựng giao diện giỏ hàng- Xây dựng giao diện đặt hàng- Xây dựng giao diện chi tiết đơn hàng- Xây dựng chức năng thêm sản phẩm vào giỏ hàng- Xây dựng giao diện trang quản lí- Xây dựng giao diện chi tiết đơn hàng- Xây dựng trang sản phẩm yêu thích

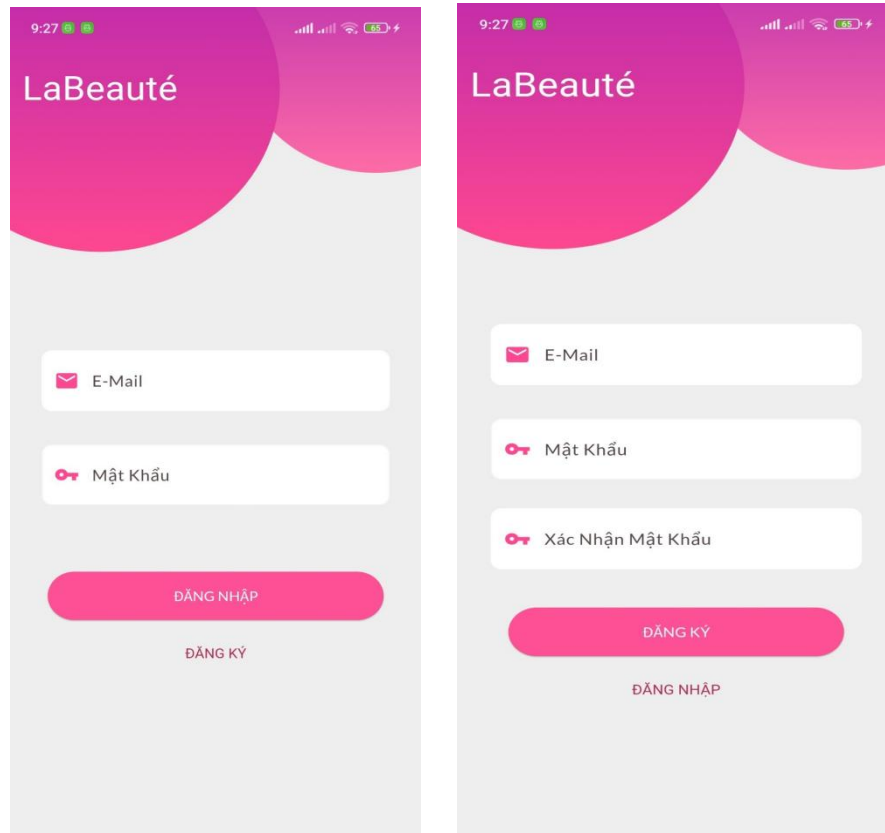
Nguyễn Thu Thảo	<ul style="list-style-type: none">- Xây dựng giao diện đăng nhập- Xây dựng giao diện đăng kí- Xây dựng trang hiển thị sản phẩm- Xây dựng giao diện trang thông tin người dùng- Xây dựng chức năng và giao diện trang quản lí sản phẩm- Xây giao diện quản lí đơn hàng- Xây dựng giao diện trang tổng quan
-----------------	---

II. Chi tiết các chức năng

1. Chức năng/giao diện 1: Đăng nhập, đăng ký

- **Miêu tả chức năng/giao diện:** Cho phép người dùng đăng ký một tài khoản với địa chỉ email cũng như là cho phép người dùng đăng nhập vào hệ thống

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, Container, Form, Center, Card, Column, SingleChildScrollView, Text, SizedBox, Padding, ValueListenableBuilder, CircularProgressIndicator, TextFormField, InputDecoration, Icon, IconButton, ElevatedButton, Stack, Positioned

+ Chức năng này có sử dụng các thư viện/plugin như Provider, Http, Convert, Shared_preferences, Foundation:

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.
- Http: giúp thực hiện các yêu cầu thông qua giao thức http.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Shared_preferences: giúp cung cấp một không gian lưu trữ cục bộ để lưu lại token của người dùng.
- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

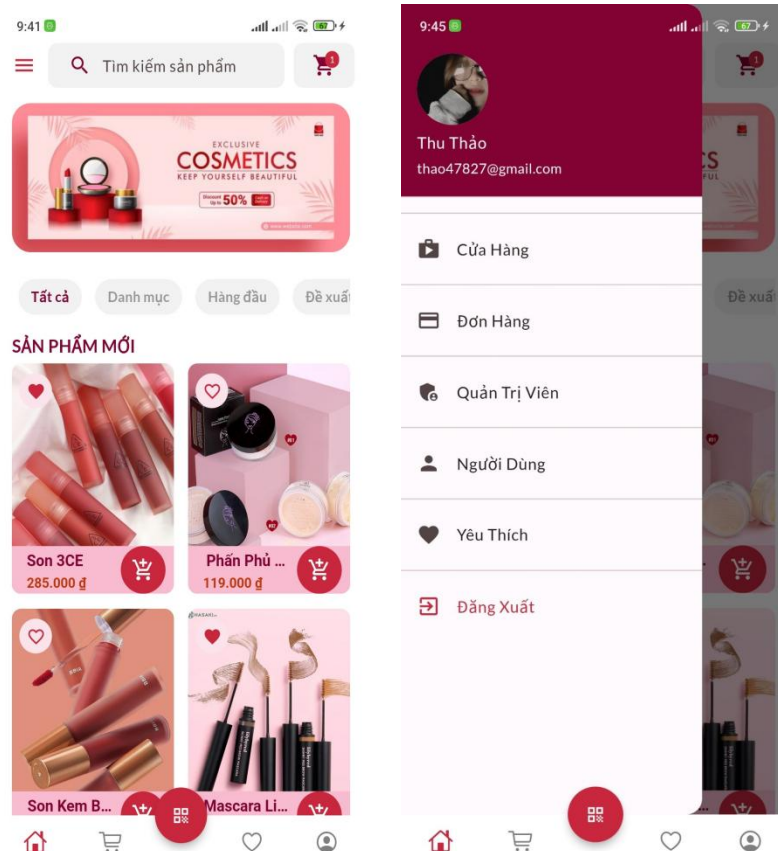
+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng AuthManager (with ChangeNotifier) lưu trữ thông tin xác thực người dùng và thực hiện việc xác thực cũng như đăng ký. Tại chức năng này sử dụng `context.read<AuthManager>()` để truy cập đối tượng AuthManager được cung cấp.

+ Chức năng này thực hiện việc xác thực và lưu trữ thông tin người dùng thông qua dịch vụ Firebase Authentication.

2.Chức năng/giao diện 2: Giao diện trang chủ

- **Miêu tả chức năng/giao diện:** Hiển thị tất cả loại mỹ phẩm của shop. Cho phép người dùng thêm sản phẩm vào giỏ hàng, hiện danh mục các sản phẩm của shop, hiện thị banner để quảng cáo các sản phẩm nổi bật của shop.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, ClipRRect, BottomNavigationBar, PopupMenuButton, PopupMenuItem, Icon, GridView, Container, GridTile, GridTileBar, Text, IconButton, Stack, Positioned, TextFormField, InputDecoration, SizedBox, ShoppingCartButton, BoxDecoration, FittedBox, Align, Expanded.

+ Chức năng này có sử dụng các thư viện/plugin như Provider, Http, Convert, Intl, Foundation

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.
- Http: giúp thực hiện các yêu cầu thông qua giao thức http, lấy chuỗi JSON từ Firebase Realtime Database.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Intl: dùng để format dữ liệu dạng Datetime.
- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (withChangeNotifier) để quản lý danh sách các loại mỹ phẩm và quản lý việc hiển thị số lượng sản phẩm trong giỏ hàng cũng như thêm sản phẩm vào giỏ hàng. Tại chức năng này sử dụng `context.select<ProductsManager, List<Product>>(){}>` và `context.watch<CartManager>()` để truy cập đối tượng ProductManager và CartManager được cung cấp.

+ Sử dụng StatefullWidget với biến trạng thái `_showProduct` để chọn cách hiển thị sản phẩm (các loại mỹ phẩm).

+ Chức năng này đọc dữ liệu các mã phẩm từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON

```
{  
  "products": [  
    "id" : {  
      "description": ,  
      "imageUrl": ,  
      "title": ,  
      "price": ,  
      "creatorId": ,  
    }  
  ]  
}
```

3. Chức năng/giao diện 3: Giỏ hàng

- **Miêu tả chức năng/giao diện:** Hiển thị các sản phẩm mà người dùng thêm vào giỏ hàng cũng như tổng số tiền các sản phẩm có trong giỏ và cho phép điều chỉnh số lượng cũng như xóa sản phẩm ra khỏi giỏ hàng.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, Column, ListView, Card, Padding, Row, Text, Spacer, TextButton, SizedBox, Dismissible, Container, Icon, showDialog, AlertDialog, ElevatedButton, ClipRRect.

+ Chức năng này có sử dụng các thư viện/plugin như Provider, Http, Convert, Shared_preferences, Intl, Cupertino.

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.

- Http: giúp thực hiện các yêu cầu thông qua giao thức http: thực hiện phương thức post để lưu thông tin đặt hàng lên Firebase Realtime Database.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Shared_preferences: giúp cung cấp một không gian lưu trữ cục bộ để lưu lại thông tin giỏ hàng của người dùng.
- Intl: dùng để format dữ liệu dạng Datetime.
- Cupertino: dùng để cung cấp các widget để thiết kế giao diện.

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng OrderManager và CartManager (with ChangeNotifier) để quản lý việc đặt hàng và quản lý việc hiển thị trong giỏ hàng. Tại chức năng này sử dụng context.read< OrderManager >() và context.read< CartManager >() để truy cập đối tượng OrderManager và CartManager được cung cấp.

+ Chức năng này đọc và lưu dữ liệu thông tin giỏ hàng của người dùng từ Shared_preferences. Với cấu trúc JSON

```
{
  "UserCart": {
    "item_id":
      {
        "id",
        "name",
        "title"
        "quantity",
        "price",
        "imageUrl",
      }
    }
}
```

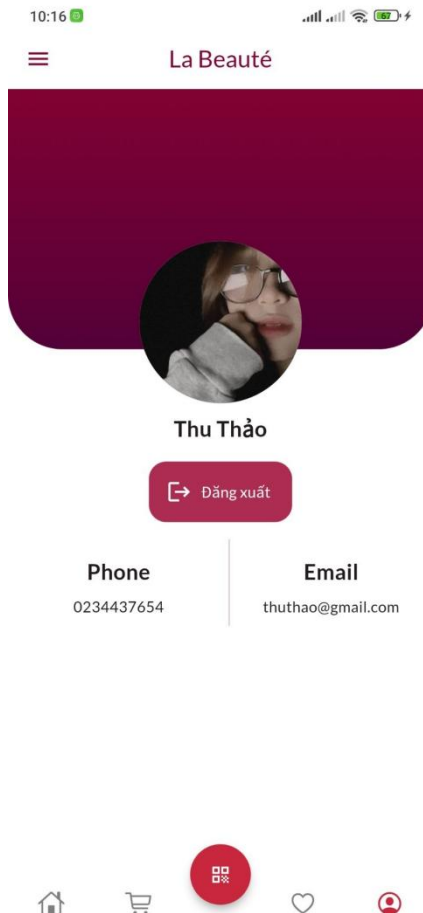
+ Chức năng này cũng thực hiện việc lưu thông tin đặt hàng của người dùng lên Firebase Realtime Database với cấu trúc JSON

```
{
  "orders": {
    "uid": {
      "order_id": {
        "amount":,
        "datetime",
        "id",
        "products": [
          {
            "id",
            "title",
            "quantity",
            "price",
            "imageUrl",
          }
        ]
      }
    }
  }
}
```

4. Chức năng/giao diện 4: Thông tin người dùng

- **Miêu tả chức năng/giao diện:** Hiển thị thông tin người dùng và cho phép đăng xuất.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, ButtonNavigationBar, Expanded, Padding, Column, Text, FloatingActionButton, Container, Row, Stack, Align, CircleAvatar.

+ Chức năng này có sử dụng các thư viện/plugin: Provider, Shared_preferences

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.

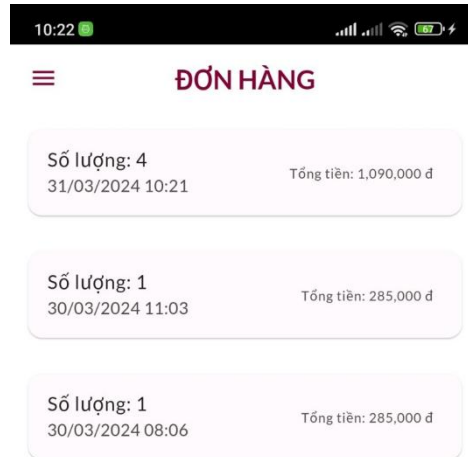
- Shared_preferences: giúp cung cấp một không gian lưu trữ cục bộ để lưu lại token của người dùng.

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng AuthManager (with ChangeNotifier) để xóa thông tin token ở Shared_preferences thực hiện chức năng đăng xuất. Tại chức năng này sử dụng context.read<AuthManager>() để truy cập đối tượng AuthManager được cung cấp.

5. Chức năng/giao diện 5: Quản lý đơn đặt hàng

- **Miêu tả chức năng/giao diện:** Hiển thị tất cả các đơn đặt hàng mà người dùng đã đặt.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, ButtonNavigationBar, ListView, ListTile, Text.

+ Chức năng này có sử dụng các thư viện/plugin như Provider, Http, Convert, Shared_preferences, Intl, Foundation.

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.
- Http: giúp thực hiện các yêu cầu thông qua giao thức http để lấy thông tin các đơn đặt hàng của người dùng từ Firebase Realtime Database.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Shared_preferences: giúp cung cấp một không gian lưu trữ cục bộ để lưu lại thông tin giỏ hàng của người dùng.
- Intl: dùng để format dữ liệu dạng Datetime.

- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng OrderManager (with ChangeNotifier) để quản lý các đơn đặt hàng. Tại chức năng này sử dụng context.read< OrderManager >() để truy cập đối tượng OrderManager và được cung cấp.

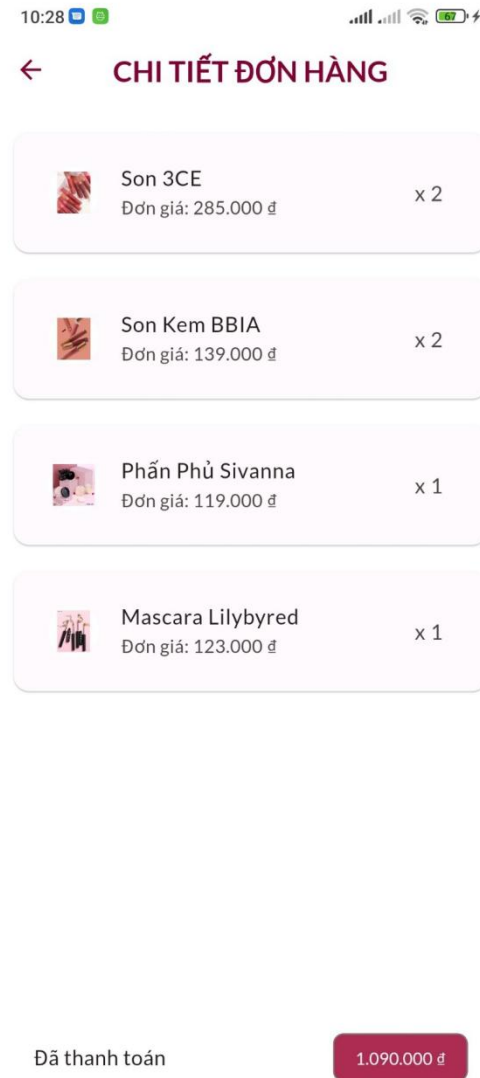
+ Chức năng này đọc dữ liệu thông tin tất cả đơn đặt hàng của người dùng từ Firebase Realtime Database. Với cấu trúc JSON

```
{
  "orders": {
    "uid": {
      "order_id": {
        "amount":,
        "datetime",
        "id",
        "products": [
          {
            "id",
            "name",
            "quantity",
            "price",
            "imageUrl",
          }
        ]
      }
    }
  }
}
```


6. Chức năng/giao diện 6: Hiển thị chi tiết đơn đặt hàng

- **Miêu tả chức năng/giao diện:** Hiển thị chi tiết một đơn đặt hàng của người dùng.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, ListView, ListTile, Text, Card, Padding, Row, Spacer, CircleAvatar, Container, Padding. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

+ Chức năng này có sử dụng các thư viện/plugin: Provider, Http, Convert, Shared_preferences, Intl, Foundation.

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.
- Http: giúp thực hiện các yêu cầu thông qua giao thức http để lấy thông tin các đơn đặt hàng của người dùng từ Firebase Realtime Database.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Shared_preferences: giúp cung cấp một không gian lưu trữ cục bộ để lưu lại thông tin đơn hàng của người dùng.
- Intl: dùng để format dữ liệu dạng Datetime.
- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng.

+ Chức năng này đọc dữ liệu thông tin chi tiết đơn hàng của người dùng từ Firebase Realtime Database. Với cấu trúc JSON

```
{
  "orders": {
    "uid": {
      "order_id": {
        "amount":,
        "datetime",
        "id",
        "products": [
          {
            "id",
            "name",
            "quantity",
            "price",
            "imageUrl",
          }
        ]
      }
    }
  }
}
```

]

}

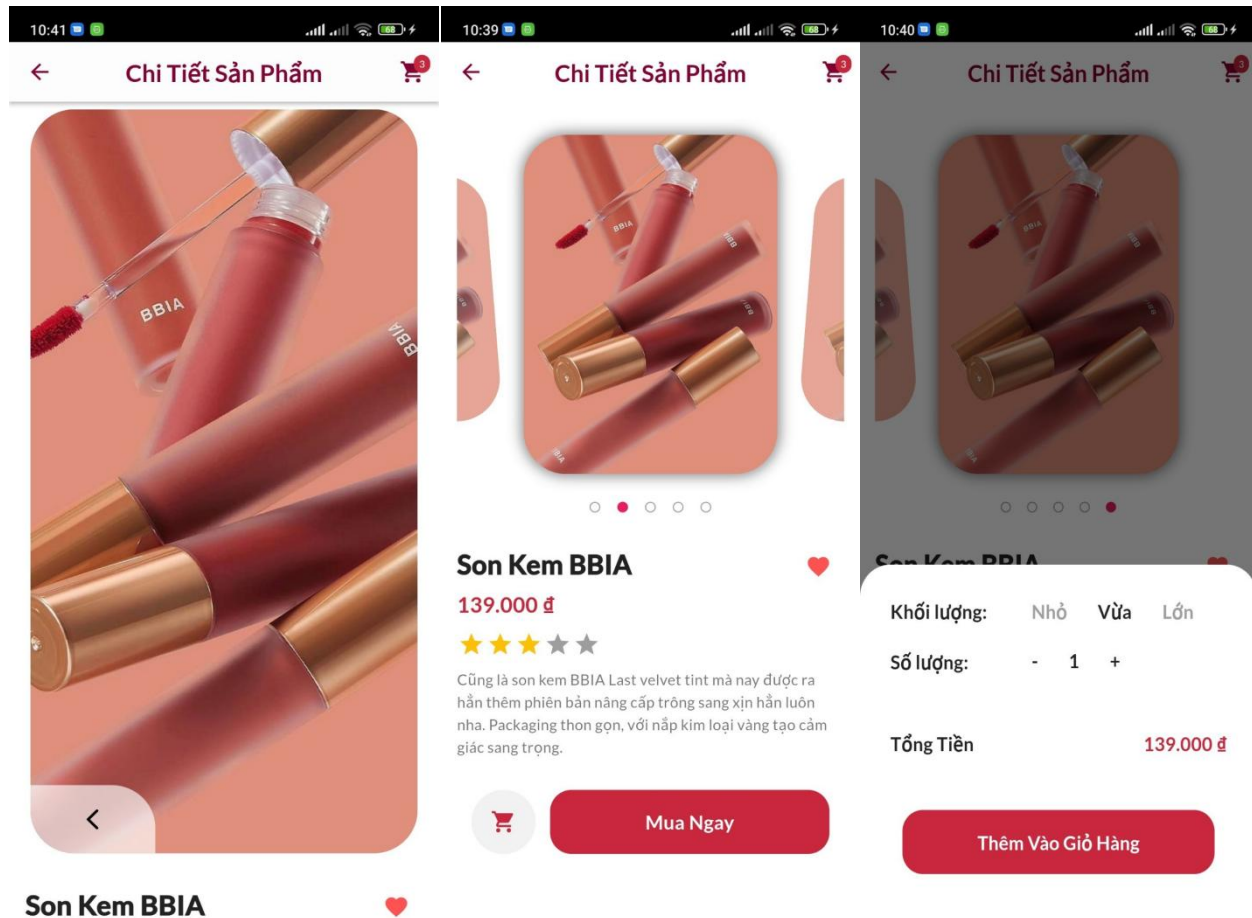
}

}

}

7. Chức năng/giao diện 7: Hiển thị chi tiết sản phẩm

- **Miêu tả chức năng/giao diện:** Hiển thị chi tiết một đơn đặt hàng của người dùng.
- **Ảnh chức năng/giao diện:**



- Chi tiết cài đặt:

- + Các widget sử dụng trong giao diện này: Scaffold, AppBar, ClipRRect, BottomNavigationBar, PopupMenuButton, PopupMenuItem, Icon, GridView, Container, GridTile, GridTileBar, Text, IconButton, Stack, Positioned, TextFormField, InputDecoration, SizedBox, ShoppingCartButton, BoxDecoration, FittedBox, Align, Expanded, Padding, Row, GestureDetector, ScaffoldMessenger
- + Chức năng này có sử dụng các thư viện/plugin: Provider, Http, Convert, Intl, Foundation, Fan_carousel, Flutter_rating_bar.

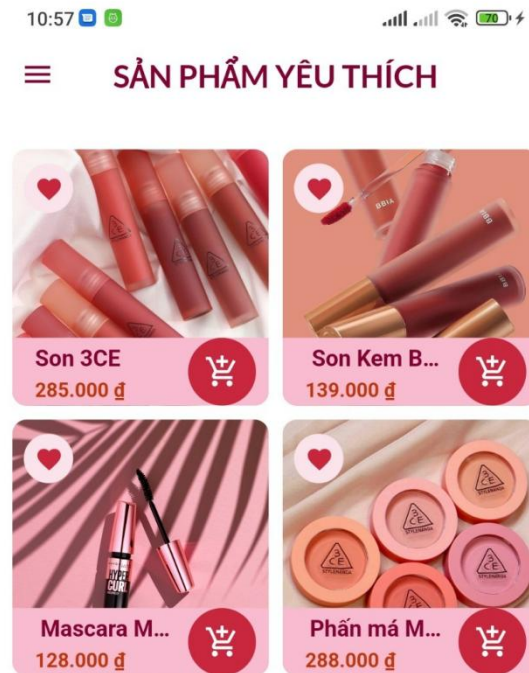
- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget
 - Http: giúp thực hiện các yêu cầu thông qua giao thức http, lấy chuỗi JSON từ Firebase Realtime Database.
 - Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
 - Intl: dùng để format dữ liệu dạng Datetime.
 - Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng.
 - Fan_carousel: tạo ra các carousel hoặc slider, cho phép người dùng chuyển đổi giữa các phần tử như ảnh, video.
 - Flutter_rating_bar: là một gói thư viện Flutter cho phép bạn tạo ra các thanh đánh giá. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...
- + Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng StatefullWidget với biến trạng thái `_showProduct` để chọn cách hiển thị sản phẩm (các loại mỹ phẩm).
- + Chức năng này đọc dữ liệu các mỹ phẩm từ dịch vụ lưu trữ Firebase Realtime.
- + Trong widget `ShoppingCartButton1`, sử dụng `context.watch<CartManager>()` để lắng nghe sự thay đổi trong `CartManager`, đặc biệt là số lượng sản phẩm trong giỏ hàng, và cập nhật biểu tượng badge trên nút giỏ hàng.
- + Trong widget `ProductDetailScreen`, sử dụng `context.read<CartManager>()` để truy cập đối tượng `CartManager` và thêm sản phẩm vào giỏ hàng khi người dùng nhấn vào nút "Thêm vào giỏ hàng". Bạn cũng sử dụng `context.select<ProductsManager, List<Product>>()` để lấy danh sách sản phẩm từ `ProductManager` nếu cần.
- + Trong widget `ProductDetailsPopUp`, sử dụng `context.read<CartManager>()` để thêm sản phẩm vào giỏ hàng khi người dùng nhấn vào nút "Thêm vào giỏ hàng".
- Database. Với cấu trúc JSON

```
{  
  "products": [  
    "id" :{  
      "description": ,  
      "imageUrl":,  
      "title": ,  
      "price": ,  
      "creatorId": ,  
    }  
  ]  
}
```

8. Chức năng/giao diện 8: Hiển thị sản phẩm yêu thích

- **Miêu tả chức năng/giao diện:** Hiển thị danh sách các sản phẩm yêu thích của khách hàng.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, ClipRRect, BottomNavigationBar, PopupMenuButton, PopupMenuItem, Icon, GridView, Container, GridTile, GridTileBar, TextFormField, InputDecoration, SizedBox, ShoppingCartButton, BoxDecoration, FittedBox, Align, Expanded.

+ Chức năng này có sử dụng các thư viện/plugin: Provider, Http, Convert, Intl, Foundation.

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.
- Http: giúp thực hiện các yêu cầu thông qua giao thức http, lấy chuỗi JSON từ Firebase Realtime Database.
- Convert: giúp chuyển đổi thành kiểu Json và ngược lại.
- Intl: dùng để format dữ liệu dạng Datetime.
- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (with ChangeNotifier) để quản lý danh sách các loại mỹ phẩm yêu thích. Tại chức năng này sử dụng `context.select<ProductsManager, List<Product>> () {}` và `context.watch< CartManager >()` để truy cập đối tượng ProductManager và CartManager được cung cấp.

+ Sử dụng StatefullWidget với biến trạng thái `_showProduct` để chọn cách hiển thị sản phẩm yêu thích được người dùng chọn (các loại mỹ phẩm).

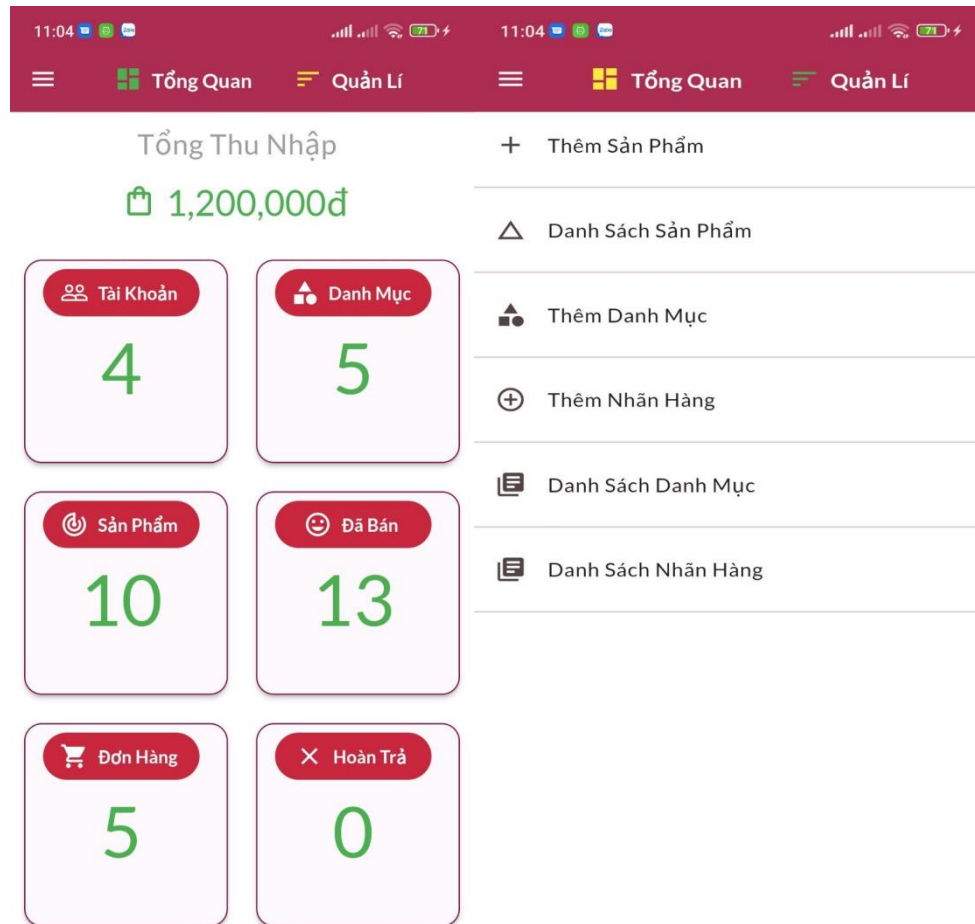
+ Chức năng này đọc dữ liệu các mã phẩm từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON

```
{  
  "products": [  
    "id": {  
      "description": ,  
      "imageUrl": ,  
      "title": ,  
      "price": ,  
      "creatorId": ,  
    }  
  ]  
}
```

9. Chức năng/giao diện 9: Giao diện quản trị viên

- **Miêu tả chức năng/giao diện:** giúp quản trị viên có thể quản lý sản phẩm của cửa hàng thống kê các loại sản phẩm hay cập nhật sản phẩm của cửa hàng.

- **Ảnh chức năng/giao diện:**



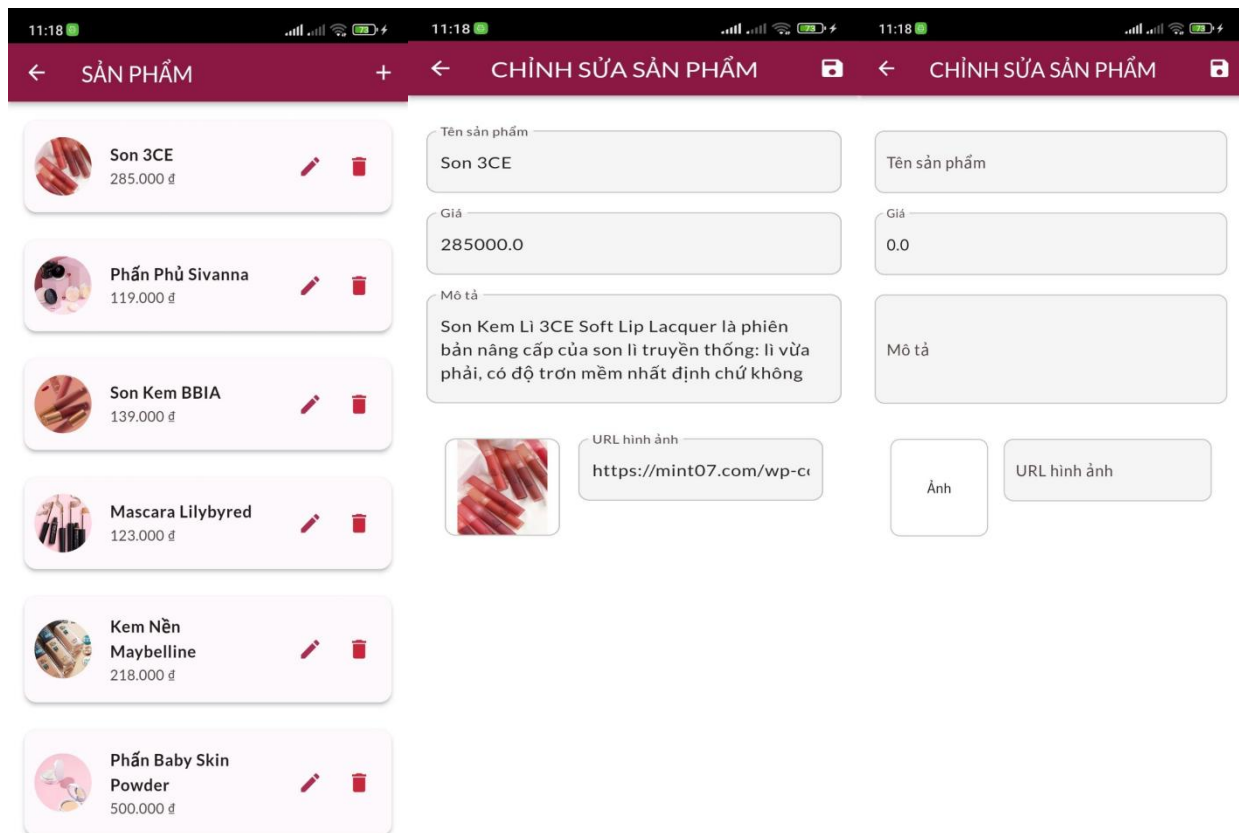
- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, Row, BottomNavigationBar, PopupMenuButton, PopupMenuItem, Icon, GridView, Container, GridTile, GridTileBar, TextFormField, InputDecoration, SizedBox, ShoppingCaMaterialButton, BoxDecoration, FittedBox, Align, Expanded.

10. Chức năng/giao diện 10: Quản lý sản phẩm

- **Miêu tả chức năng/giao diện:** giúp quản trị viên có thể xem sản phẩm của cửa hàng, thêm sản phẩm mới.

- **Ảnh chức năng/giao diện:**



- **Chi tiết cài đặt:**

+ Các widget sử dụng trong giao diện này: Scaffold, AppBar, Row, BottomNavigationBar, PopupMenuButton, PopupMenuItem, Icon, ListView, Container, GridTile, GridTileBar, TextFormField, InputDecoration, SizedBox, BoxDecoration, FittedBox, Align, Expanded, FutureBuilder.

+ Chức năng này có sử dụng các thư viện/plugin: Provider, Http, Intl, Foundation

- Provider: giúp quản lý trạng thái chia sẻ bằng cách cung cấp các đối tượng hoặc giá trị cho các widget hậu duệ thông qua Inherited Widget.

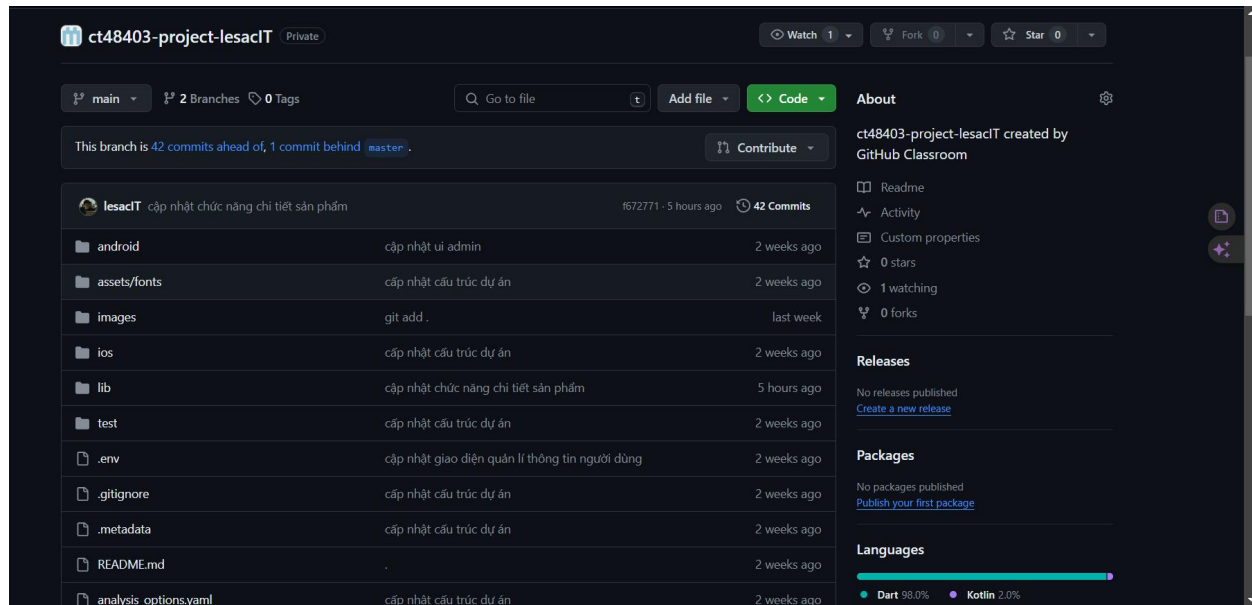
- Http: giúp thực hiện các yêu cầu thông qua giao thức http, lấy chuỗi JSON từ Firebase Realtime Database.
- Intl: dùng để format dữ liệu dạng Datetime.
- Foundation: cung cấp các khái niệm cơ bản và cấu trúc cho việc xây dựng ứng dụng. Gồm các Widgets cơ bản như Container, Row, Column, Text, Image,...

+ Chức năng này có sử dụng trạng thái chia sẻ: Sử dụng Widget MultiProvider ở đầu cây Widget để cung cấp đối tượng ProductManager và CartManager (with ChangeNotifier) để quản lý danh sách các loại mỹ phẩm. Tại chức năng này sử dụng `context.select<ProductsManager, List<Product>> () {}` và `context.watch<CartManager >()` để truy cập đối tượng ProductManager và CartManager hiện thị các sản phẩm.

+ Chức năng này đọc dữ liệu các mỹ phẩm từ dịch vụ lưu trữ Firebase Realtime Database. Với cấu trúc JSON

```
{
  "products": [
    {
      "id": {
        "description": ,
        "imageUrl":,
        "title": ,
        "price": ,
        "creatorId": ,
      }
    }
  ]
}
```

Lưu mã nguồn vào repo git và lên GitHub:



Link Github

<https://github.com/23-24Sem2-Courses/ct48403-project-lesacIT.git>