

TRƯỜNG ĐẠI HỌC CẦN THƠ

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Phát Triển Ứng Dụng Di Động
App bán giày NIKE

Mã số học phần: CT484 – Nhóm HP: 02

Giảng viên hướng dẫn: TS. Bùi Võ Quốc Bảo

HK2 2023-2024

Sinh viên thực hiện

Lê Phước Vinh

B1910176

DI19V7A6

(vinhb1910176@student.ctu.edu.vn)

MỤC LỤC

I. Tổng quan	2
1. Miêu tả dự án:	2
2. Link github dự án: https://github.com/PhuocVinhLee/project_app_NIKE	2
II. Chi tiết các chức năng.....	2
1. Chức năng/giao diện: trang tổng quan các sản phẩm	2
2. Chức năng/giao diện: trang hiển thị thông tin chi tiết sản phẩm.....	3
3. Chức năng/giao diện: trang giỏ hàng	5
4. Chức năng/giao diện: trang quản lý sản phẩm	6
5. Chức năng/giao diện: trang đặt hàng.....	7
6. Chức năng/giao diện: trang đăng nhập.....	8

I. Tổng quan

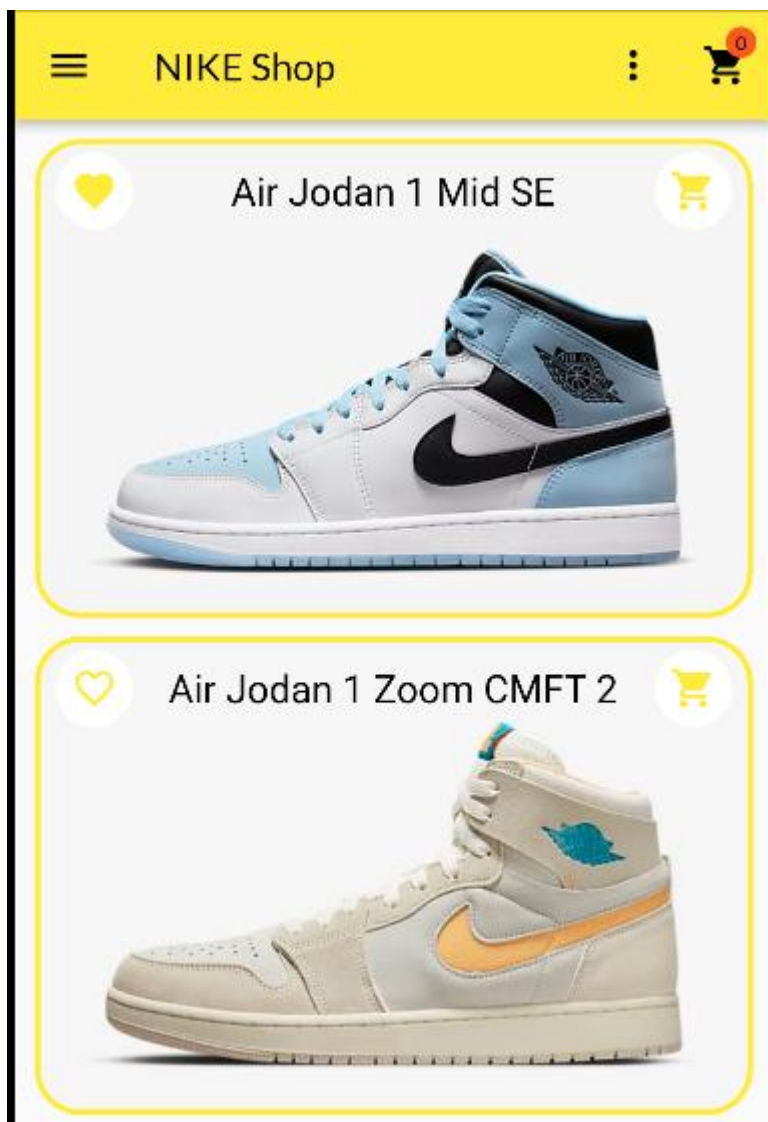
1. Miêu tả dự án:

2. Link github dự án: https://github.com/PhuocVinhLee/project_app_NIKE

II. Chi tiết các chức năng

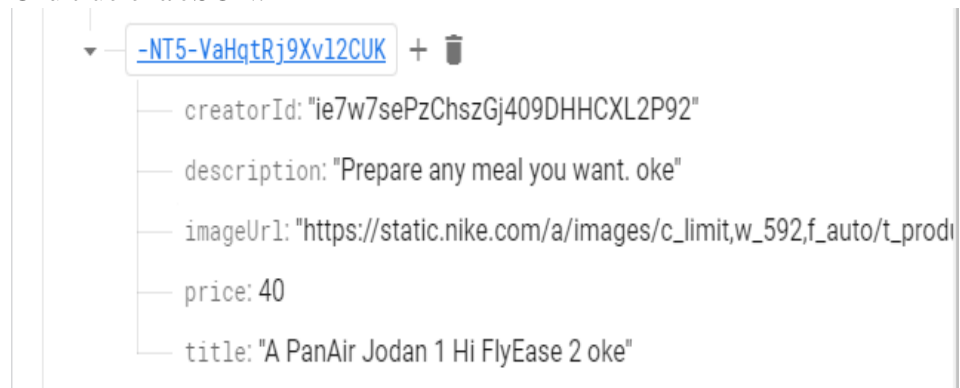
1. Chức năng/giao diện: trang tổng quan các sản phẩm

- **Miêu tả chức năng/giao diện:** Trang tổng quan các sản phẩm người dùng có thể xem tất những sản phẩm hiện có nếu người dùng cảm thấy thích sản phẩm nào có thể thêm sản phẩm đó vào danh sách sản phẩm yêu thích hoặc người dùng muốn mua sản phẩm đó thì có thể thêm vào giỏ hàng.
- **Ảnh chức năng/giao diện:**



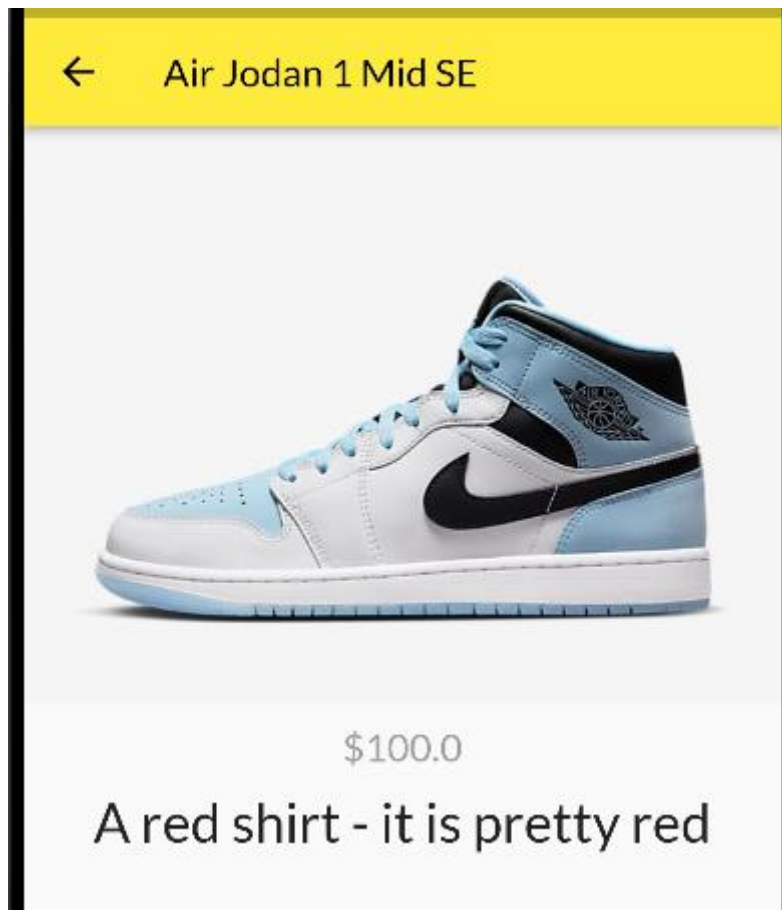
- **Chi tiết cài đặt:**

- **Các widget đã sử dụng:** buildProductFilterMenu, buildShoppingCartIcon, AppDrawer, FutureBuilder, ValueListenableBuilder, ProductsGrid, Center, CircularProgressIndicator, TopRightBadge,
- **Các thư viện hay plugin đã sử dụng:**
 - Gói **flutter/foundation.dart**: cho phép widget như **ValueListenableBuilder** lắng nghe sự thay đổi giá trị của biến.
- **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện **provider** để chia sẻ trạng thái
 - Kiến trúc code:
 - ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
 - ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
 - ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
 - ✓ Tái tạo lại UI sau khi trạng thái thay đổi.
- **Thực hiện đọc hoặc lưu trữ dữ liệu:**
 - Có đọc dữ liệu và lưu trữ
 - Từ các dịch vụ lưu trữ
 - Cấu trúc của JSON:

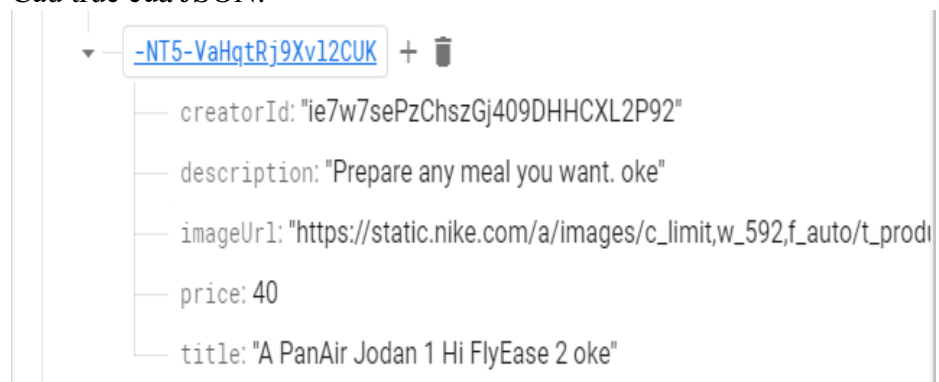


2. Chức năng/giao diện: trang hiển thị thông tin chi tiết sản phẩm

- **Miêu tả chức năng/giao diện:** Trang hiển thị thông tin chi tiết sản phẩm người dùng có thể xem chi tiết sản như là: tên, giá, miêu tả sản phẩm, hình ảnh.
- **Ảnh chức năng/giao:**

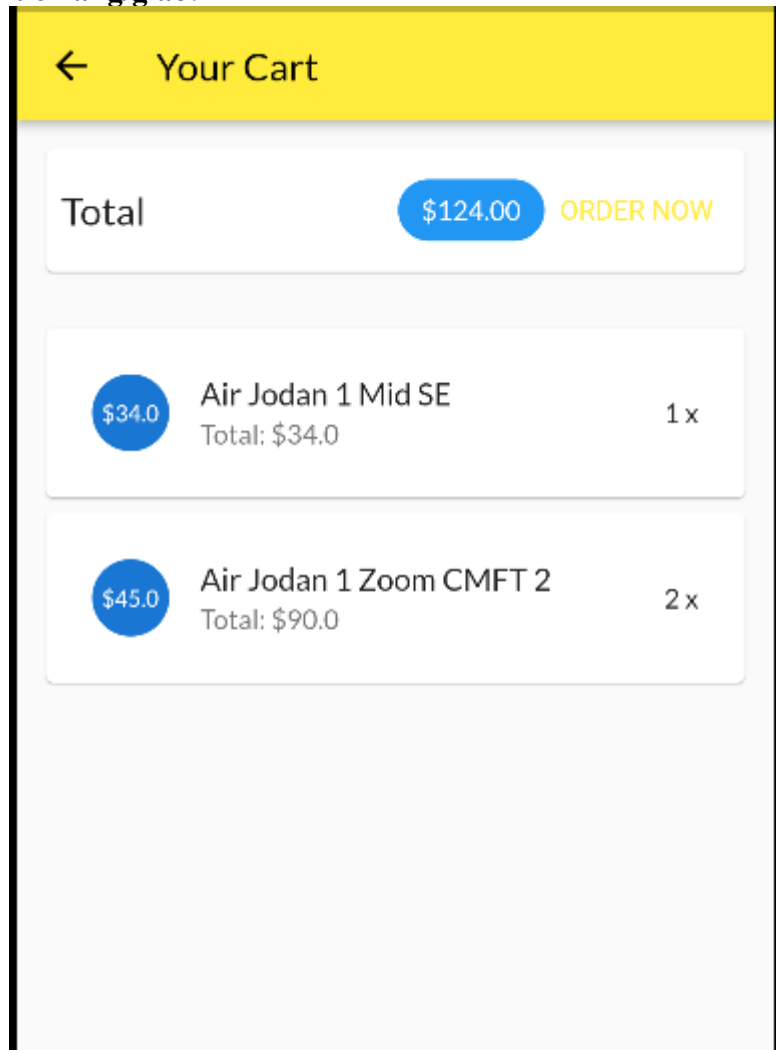


- **Chi tiết cài đặt:**
 - **Các widget đã sử dụng:** SingleChildScrollView, Column, SizedBox
 - **Các thư viện hay plugin đã sử dụng:**
 - **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện provider để chia sẻ trạng thái
 - Kiến trúc code:
 - ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
 - ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
 - ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
 - ✓ Tái tạo lại UI sau khi trạng thái thay đổi.
 - **Thực hiện đọc hoặc lưu trữ dữ liệu:**
 - Có đọc dữ liệu và lưu trữ
 - Từ các dịch vụ lưu trữ
 - Cấu trúc của JSON:



3. Chức năng/giao diện: trang giỏ hàng

- **Miêu tả chức năng/giao diện:** Trang giỏ hàng người dùng có thể xem chi tiết các sản phẩm đã thêm vào trong giỏ hàng trước đó và đồng thời người dùng có thể nhấn nút ORDER NOW để mua hàng
- **Ảnh chức năng/giao:**



- **Chi tiết cài đặt:**
 - **Các widget đã sử dụng:** buildCartSummary, SizedBox, Expanded, buildCartDetails, ListView, Padding, Row, Spacer, Dismissible, FittedBox
 - **Các thư viện hay plugin đã sử dụng:**
 - Gói **flutter/foundation.dart**: cho phép widget như **ValueListenableBuilder** lắng nghe sự thay đổi giá trị của biến.
 - **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện provider để chia sẻ trạng thái

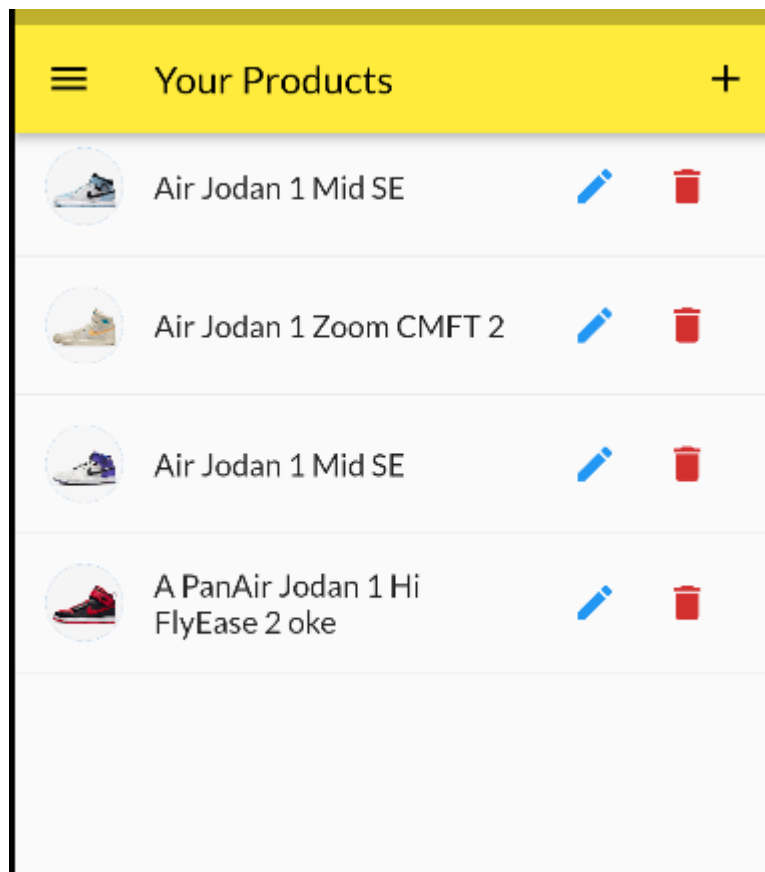
- Kiến trúc code:
 - ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
 - ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
 - ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
 - ✓ Tái tạo lại UI sau khi trạng thái thay đổi.

- **Thực hiện đọc hoặc lưu trữ dữ liệu:**

- Có đọc dữ liệu và lưu trữ
- Lưu trữ cục bộ

4. Chức năng/giao diện: trang quản lý sản phẩm

- **Miêu tả chức năng/giao diện:** Trang quản lý sản phẩm người dùng có thể thêm, sửa, xóa các sản phẩm
- **Ảnh chức năng/giao:**



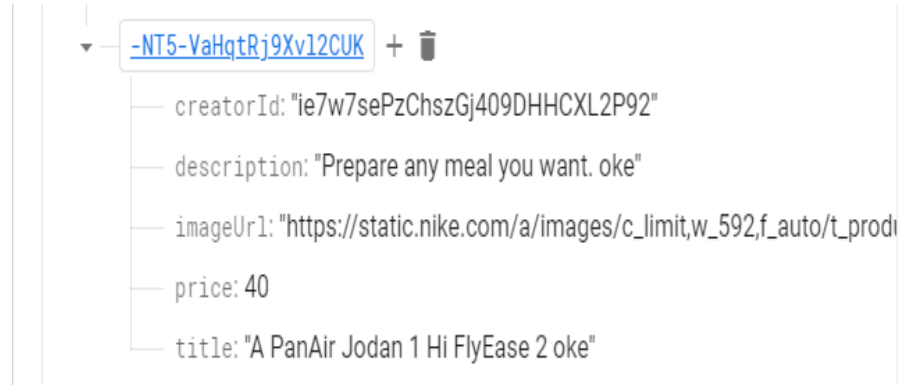
- **Chi tiết cài đặt:**

- **Các widget đã sử dụng:** buildTitleField, buildPriceField, buildDescriptionField, buildProductPreview
- **Các thư viện hay plugin đã sử dụng:**
 - Gói **flutter/foundation.dart**: cho phép widget như **ValueListenableBuilder** lắng nghe sự thay đổi giá trị của biến.
- **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện provider để chia sẻ trạng thái
 - Kiến trúc code:

- ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
- ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
- ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
- ✓ Tái tạo lại UI sau khi trạng thái thay đổi.

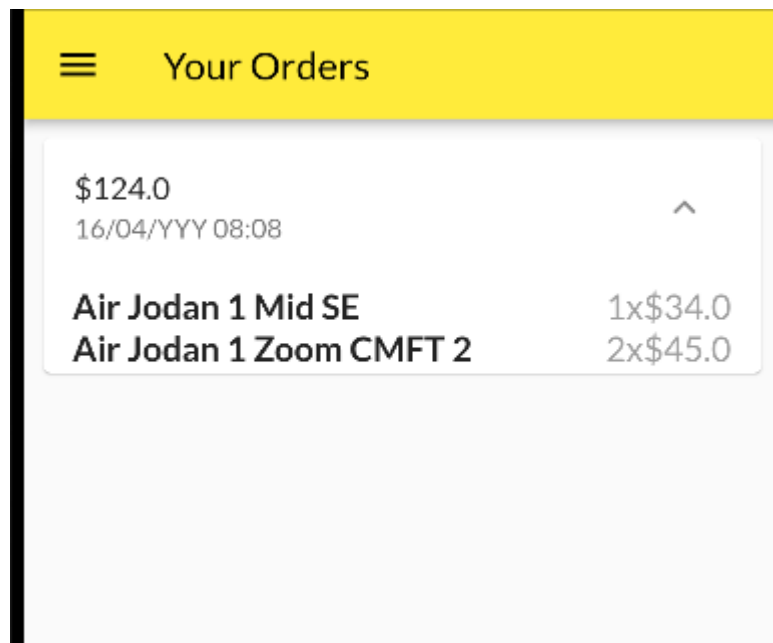
- **Thực hiện đọc hoặc lưu trữ dữ liệu:**

- Có đọc và lưu trữ dữ liệu
- Từ các dịch vụ lưu trữ
- Cấu trúc của JSON:



5. Chức năng/giao diện: trang đặt hàng

- **Miêu tả chức năng/giao diện:** Trang đặt hàng cho phép người dùng xem lại thông tin đơn hàng đã đặt
- **Ảnh chức năng/giao:**

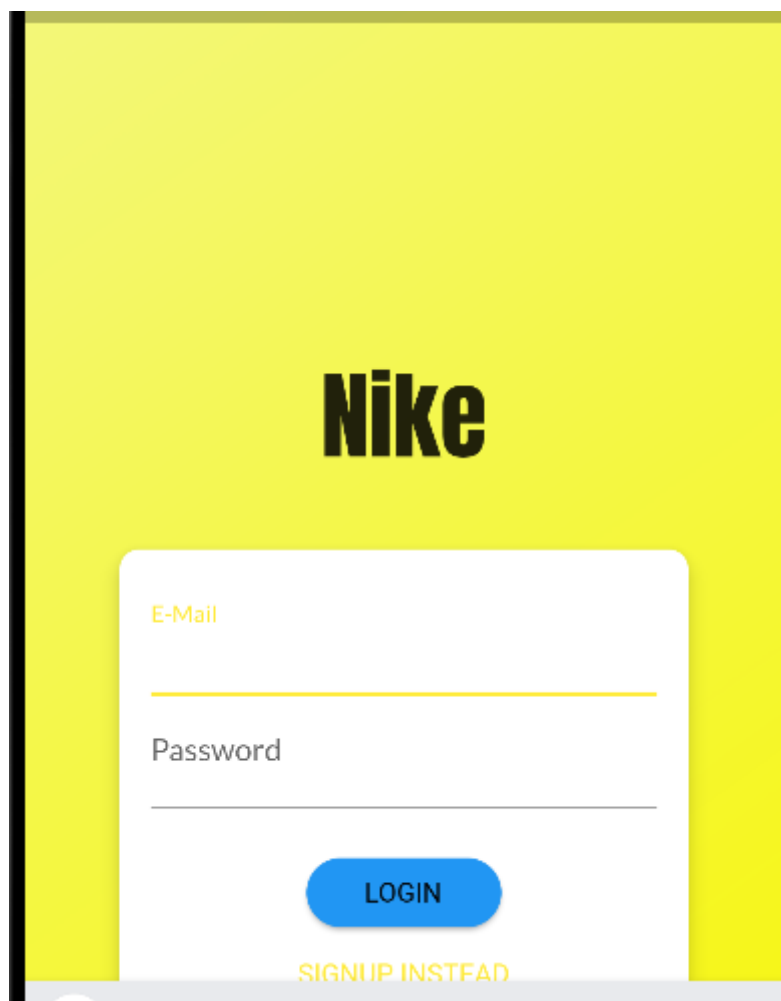


- **Chi tiết cài đặt:**
 - **Các widget đã sử dụng:** OrderItemCard, AppDrawer, buildOrderDetails, buildOrderSummary

- **Các thư viện hay plugin đã sử dụng:**
 - Gói **flutter/foundation.dart**: cho phép widget như **ValueListenableBuilder** lắng nghe sự thay đổi giá trị của biến.
 - Gói **intl**: sử dụng thư viện intl để định dạng ngày tháng
- **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện provider để chia sẻ trạng thái
 - Kiến trúc code:
 - ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
 - ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
 - ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
 - ✓ Tái tạo lại UI sau khi trạng thái thay đổi.
- **Thực hiện đọc hoặc lưu trữ dữ liệu:**
 - Có đọc dữ liệu và lưu trữ
 - Từ cục bộ

6. Chức năng/giao diện: trang đăng nhập

- **Miêu tả chức năng/giao diện:** Trang đăng nhập, người dùng có thể đăng nhập vào ứng dụng
- **Ảnh chức năng/giao:**



- **Chi tiết cài đặt:**

- **Các widget đã sử dụng:** Container, AppBanner, Flexible, AuthCard, _buildEmailField, _buildPasswordField, _buildSubmitButton, _buildAuthModeSwitchButton, _buildPasswordConfirmField
- **Các thư viện hay plugin đã sử dụng:**
 - Gói flutter/foundation.dart: cho phép widget như ValueListenableBuilder lắng nghe sự thay đổi giá trị của biến.
 - Thư viện flutter_dotenv:
 - Thư viện http:
 - Thư viện shared_preferences
- **Sử dụng giải pháp quản lý trạng thái chia sẻ:**
 - Gói thư viện provider để chia sẻ trạng thái
 - Kiến trúc code:
 - ✓ cung cấp cho lớp UI một tham chiếu đến lớp quản lý trạng thái
 - ✓ Báo hiệu cho lớp trạng thái biết về các sự kiện trên UI
 - ✓ Cung cấp cho lớp UI cách lắng nghe thay đổi về trạng thái
 - ✓ Tái tạo lại UI sau khi trạng thái thay đổi.
- **Thực hiện đọc hoặc lưu trữ dữ liệu:**
 - Có đọc và lưu trữ dữ liệu
 - Từ các dịch vụ lưu trữ
 - REST API:
 - Đăng ký người dùng mới:**
 - ✓ Phương thức yêu cầu: POST
 - ✓ Endpoint: [https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=\[API_KEY\]](https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=[API_KEY]) , với API_KEY là "Web API Key" lấy từ dự án firebase
 - ✓ Dữ liệu JSON gửi đi:

```

{
  "email": "[user@example.com]",
  "password": "[PASSWORD]",
  "returnSecureToken": true
}

```
 - ✓ Dữ liệu JSON trả lời (thành công):

```

{
  "idToken": "[ID_TOKEN]",
  "email": "[user@example.com]",
  "refreshToken": "[REFRESH_TOKEN]",
  "expiresIn": "3600",
  "localId": "tRcfmLH7..."
}

```
- **Đăng nhập người dùng**
 - ✓ Phương thức yêu cầu: POST
 - ✓ Endpoint: [https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=\[API_KEY\]](https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=[API_KEY]) , với API_KEY là "Web API Key" lấy từ dự án firebase
 - ✓ Dữ liệu JSON gửi đi:

```
{
  "email": "[user@example.com]",
  "password": "[PASSWORD]",
  "returnSecureToken": true
}
```

✓ Dữ liệu JSON trả lời (thành công):

```
{
  "localId": "ZY1rJK0eYLg...",
  "email": "[user@example.com]",
  "displayName": "",
  "idToken": "[ID_TOKEN]",
  "registered": true,
  "refreshToken": "[REFRESH_TOKEN]",
  "expiresIn": "3600"
}
```

Link github dự án: https://github.com/PhuocVinhLee/project_app_NIKE