BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

Ảnh có chứa biểu tượng

Mô tả được tạo tự động~~~~~~\*~~~~~~

**BÁO CÁO BÀI TẬP LỚN MÔN LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG**

ĐỀ TÀI: Thiết kế COUNTRY-APP

|  |  |  |
| --- | --- | --- |
| Giáo viên hướng dẫn | : | Trần Đức Minh |
| Thành viên nhóm | : | Ngô Thị Linh |
|  | : | Đặng Hoàng Anh |
|  | : | Nguyễn Việt Hưng |
|  | : | Nguyễn Xuân Hiếu |

Năm học: 2023-2024

Mục lục

PHẦN MỞ ĐẦU………………………………………………………………………………………………………………………………1

[**PHẦN MỞ ĐẦU** 1](#_Toc161010057)

[DANH MỤC HÌNH ẢNH 3](#_Toc161010058)

[DANH MỤC BẢNG 4](#_Toc161010059)

[**1. Tổng quan** 5](#_Toc161010060)

[**1.1 Khảo sát thực trạng** 5](#_Toc161010061)

[**1.2 Đánh giá** 5](#_Toc161010062)

[*1.2.1 Nhược điểm:* 5](#_Toc161010063)

[*1.2.2 Ưu điểm:* 6](#_Toc161010064)

[*1.2.3 Đối tượng:* 7](#_Toc161010065)

[*1.2.4 Phạm vi:* 7](#_Toc161010066)

[*1.2.4 Ràng buộc tổng quan hệ thống:* 8](#_Toc161010067)

[**2. CƠ SỞ LÝ THUYẾT** 10](#_Toc161010068)

[2.1 TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG 10](#_Toc161010069)

[*2.1.1. Môi trường lập trình* 10](#_Toc161010070)

[*2.1.2 Ngôn ngữ lập trình Dart* 11](#_Toc161010071)

[2.2 TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL 12](#_Toc161010072)

[**3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 14](#_Toc161010073)

[3.1 XÁC ĐỊNH YÊU CẦU 14](#_Toc161010074)

[*3.1.1 Yêu cầu chức năng* 14](#_Toc161010075)

[*3.1.2 Yêu cầu hệ thống* 14](#_Toc161010076)

[3.2 MÔ HÌNH HÓA 15](#_Toc161010077)

[3.3. FIREBASE SỬ DỤNG 15](#_Toc161010078)

[3.4. CÁC THÀNH PHẦN CHỨC NĂNG CỦA HỆ THỐNG 17](#_Toc161010079)

[*3.4.1. Chức năng đăng nhập, đăng ký hệ thống* 17](#_Toc161010080)

[*3.4.2. Chức năng tìm kiếm địa điểm* 17](#_Toc161010081)

[*3.4.3. Chức năng hiển thị thông tin* 17](#_Toc161010082)

[*3.4.4. Chức năng hiển thị quốc gia yêu thích* 18](#_Toc161010083)

[3.5 SOURCE CODE CỦA CHƯƠNG TRÌNH 18](#_Toc161010084)

[*3.5.1. Source code SignUP* 18](#_Toc161010085)

[*3.5.2 Source code SignIn* 18](#_Toc161010086)

[*3.5.3 Source code quốc gia yêu thích* 19](#_Toc161010087)

[*3.5.4. Source code danh sách quốc gia* 19](#_Toc161010088)

[*3.5.5. Source code danh sách thành phố* 20](#_Toc161010089)

[*3.5.6 Source code Main*. 20](#_Toc161010090)

[3.6 THIẾT KẾ GIAO DIỆN HỆ THỐNG 22](#_Toc161010091)

[*3.6.1. Form đăng kí- tạo tài khoản* 22](#_Toc161010092)

[*3.6.2 Form đăng nhập của phần mềm* 23](#_Toc161010093)

[*3.6.3 Form trang chủ ứng dụng người dùng* 24](#_Toc161010094)

[*3.6.4 Form giao diện lưu trữ quốc gia yêu thích* 25](#_Toc161010095)

[*3.6.5 Form giao diện danh sách thành phố* 26](#_Toc161010096)

[*3.6.6 Form giao diện thông tin thành phố* 27](#_Toc161010097)

[**KẾT LUẬN** 28](#_Toc161010098)

[4.1 KẾT LUẬN ĐỀ TÀI 28](#_Toc161010099)

[**4.1.1 Đánh giá chung** 28](#_Toc161010100)

[**4.1.2 Hướng phát triển và mở rộng đề tài** 28](#_Toc161010101)

[4.2 LỜI KẾT 29](#_Toc161010102)

**PHÂN CHIA CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Công việc |
| Ngô Thị Linh | 21011613 | Làm báo cáo chương 3 + code page quốc gia yêu thích, chức năng tìm kiếm cho page danh sách quốc gia, thành phố. |
| Đặng Hoàng Anh | 21012034 | Làm báo cao chương 4 + code page danh sách thành phố |
| Nguyễn Xuân Hiếu | 21011119 | Làm báo cáo chương 2 + firebase Sign In / Sign Up |
| Nguyễn VIệt Hưng | 21011608 | Làm báo cáo chương 1 + code page Trang chủ - danh sách thành phố |

# **PHẦN MỞ ĐẦU**

* **Lý do chọn đề tài**
* **Nhu cầu thực tiễn**: Thông tin về các quốc gia là một yếu tố quan trọng và được sử dụng hàng ngày bởi nhiều người. Một ứng dụng cung cấp thông tin về các quốc gia sẽ mang lại giá trị thực tiễn cho người dùng, giúp họ hiểu rõ hơn về các quốc gia trên thế giới.
* **Tiềm năng thị trường**: Có nhiều người sử dụng ứng dụng trên điện thoại di động để tra cứu thông tin về các quốc gia. Thị trường này luôn có nhu cầu và tiềm năng phát triển, đặc biệt là với sự phổ biến của thiết bị di động.
* **Thực hành kiến thức lập trình di động**: Xây dựng một ứng dụng về thông tin quốc gia sẽ yêu cầu bạn áp dụng kiến thức lập trình di động, xử lý dữ liệu và thiết kế giao diện người dùng. Đây là cơ hội tốt để nắm bắt và thực hành các khái niệm đã học trong lĩnh vực lập trình.
* **Tính linh hoạt và đa dạng**: Một ứng dụng về thông tin quốc gia có thể được phát triển với nhiều tính năng và phong cách khác nhau, từ việc hiển thị thông tin cơ bản đến tích hợp các tính năng nâng cao như bản đồ, thông tin văn hóa, du lịch và địa lý.
* **Thực hành nguyên tắc phát triển phần mềm**: Xây dựng một ứng dụng thực tế như Country App sẽ giúp bạn áp dụng các nguyên tắc phát triển phần mềm, kiểm thử, gỡ lỗi và tối ưu hóa ứng dụng của bạn. Điều này sẽ giúp bạn cải thiện kỹ năng lập trình và trở thành một lập trình viên chuyên nghiệp hơn.
* **Kiểm tra hiệu quả kiến thức học**: Qua việc thực hành xây dựng ứng dụng Country App, bạn có thể kiểm tra hiệu quả của kiến thức đã học và tiếp cận với thực tế một cách hiệu quả nhất. Điều này giúp bạn học cách xây dựng một ứng dụng thực tế và áp dụng các kỹ thuật lập trình hướng đối tượng vào thực tế.

Với những lợi ích trên cùng với sự nghiên cứu của các thành viên trong nhóm, việc lựa chọn đề tài "Country App" là một quyết định hợp lý và mang lại nhiều cơ hội để thực hành và phát triển kỹ năng lập trình. Với đề tài đã chọn này, chúng em mong muốn áp dụng các kiến thức đã được học trong trường cùng với việc tìm hiểu nghiên cứu ngôn ngữ và môi trường lập trình hướng thiết bị di động để xây dựng một hệ thống dự báo thời tiết được thật hiệu quả.

Do thời gian hạn chế và chưa có kinh nghiệm trong nghiên cứu và thực hành nên báo cáo còn nhiều thiếu sót. Em mong nhận được đóng góp ý kiến của thầy cô để đề tại được hoàn thiện hơn. Em xin chân thành cảm ơn!

## DANH MỤC HÌNH ẢNH

Hình 3.1 Biểu đồ phân rã chức năng (BFD)

Hình 3.2 Source code SignUp

Hình 3.3 Source code SignIn

Hình 3.4 Source code quốc gia yêu thích

Hình 3.5 Source code danh sách quốc gia

Hình 3.6 Source code danh sách quốc gia

Hình 3.7 Source code danh sách thành phố

Hình 3.8 Source code main

Hình 3.9 Form đăng kí- tạo tài khoản

Hình 3.10 Form giao diện Đăng nhập

Hình 3.11 Form trang chủ ứng dụng

Hình 3.12 Form giao diện lưu trữ quốc gia yêu thích

Hình 3.13 Form giao diện danh sách thành phố

Hình 3.14 Form giao diện thông tin thành phố

## DANH MỤC BẢNG

Bảng 3.1 Chức năng Form đăng kí

Bảng 3.2 Chức năng đăng nhập của phần mềm

Bảng 3.3 Chức năng lưu trữ quốc gia yêu thích

**1. Tổng quan**

**1.1 Khảo sát thực trạng**

Hiện tại, ứng dụng chứa thông tin về các thành phố và quốc gia rất phổ biến và được sử dụng rộng rãi trên nhiều nền tảng, bao gồm cả web và di động. Các ứng dụng này cung cấp thông tin chi tiết về các thành phố và quốc gia trên toàn thế giới, bao gồm thông tin địa lý, dân số, ngôn ngữ chính, văn hóa, lịch sử, thời tiết, và nhiều thông tin khác.

Dưới đây là một số thực trạng và xu hướng liên quan đến ứng dụng chứa thông tin về các thành phố và quốc gia:

* **Đa dạng và phong phú**: Có rất nhiều ứng dụng chứa thông tin về các thành phố và quốc gia khác nhau. Một số ứng dụng tập trung vào cung cấp thông tin du lịch, trong khi các ứng dụng khác tập trung vào thông tin địa lý và lịch sử. Các ứng dụng này cung cấp một góc nhìn đa chiều về các thành phố và quốc gia, giúp người dùng hiểu rõ hơn về các địa điểm và văn hóa khác nhau trên thế giới.
* **Tích hợp dịch vụ và chức năng**: Nhiều ứng dụng chứa thông tin về các thành phố và quốc gia cung cấp các tính năng và dịch vụ bổ sung. Ví dụ, người dùng có thể tìm kiếm thông tin về khách sạn, nhà hàng, điểm du lịch, hoặc thậm chí đặt vé máy bay và khám phá các chuyến đi.
* **Liên kết với dữ liệu thời gian thực**: Một số ứng dụng kết hợp thông tin về thành phố và quốc gia với dữ liệu thời tiết và thông tin giao thông để cung cấp cho người dùng cái nhìn tổng quan về điều kiện hiện tại và dự báo trong khu vực đó.
* **Cập nhật thông tin thường xuyên**: Do thông tin về các thành phố và quốc gia có thể thay đổi theo thời gian, nhiều ứng dụng cung cấp cập nhật thông tin thường xuyên để đảm bảo rằng người dùng có thông tin mới nhất và chính xác nhất.

**1.2 Đánh giá**

*1.2.1 Nhược điểm:*

Mặc dù ứng dụng chứa thông tin về các thành phố và quốc gia mang lại nhiều lợi ích, nhưng cũng có một số nhược điểm sau:

* **Độ tin cậy của thông tin**: Ứng dụng có thể cung cấp thông tin không chính xác hoặc cũ về các thành phố và quốc gia. Điều này có thể dẫn đến nhầm lẫn hoặc thông tin không chính xác cho người dùng. Do đó, người dùng cần kiểm tra và xác nhận thông tin từ các nguồn đáng tin cậy khác để đảm bảo tính chính xác.
* **Thiếu thông tin chi tiết**: Ứng dụng có thể không cung cấp đủ thông tin chi tiết về các thành phố và quốc gia. Điều này có thể làm hạn chế khả năng hiểu biết và khám phá của người dùng về một địa điểm cụ thể
* **Hạn chế về ngôn ngữ**: Ứng dụng có thể hạn chế thông tin chỉ được cung cấp bằng một số ngôn ngữ cụ thể. Điều này có thể tạo ra rào cản ngôn ngữ và cản trở khả năng tiếp cận thông tin cho một số người dùng.
* **Không đầy đủ và không cập nhật**: Ứng dụng có thể không đầy đủ về danh sách các thành phố và quốc gia trên thế giới. Ngoài ra, việc cập nhật thông tin thường xuyên cũng có thể là một thách thức đối với một số ứng dụng, dẫn đến việc thông tin không còn chính xác hoặc cập nhật.
* **Vấn đề bảo mật**: Ứng dụng có thể không đảm bảo đủ về bảo mật thông tin cá nhân của người dùng. Điều này có thể gây lo ngại về việc rò rỉ thông tin và sự riêng tư.

*1.2.2 Ưu điểm:*

**Tích hợp thông tin đa dạng**: Ứng dụng chứa thông tin đa dạng về các thành phố và quốc gia, cung cấp thông tin chi tiết về địa lý, lịch sử, văn hóa, dân số, ngôn ngữ, điểm du lịch và nhiều khía cạnh khác. Điều này giúp người dùng có cái nhìn tổng quan và sâu sắc hơn về các địa điểm quan trọng trên thế giới.

**Tính chính xác và cập nhật**: Có thể tự cập nhật thông tin mới và loại bỏ thông tin cũ để đảm bảo rằng người dùng luôn nhận được thông tin mới nhất và đáng tin cậy.

**Tùy chỉnh và linh hoạt**: Tùy chỉnh và linh hoạt trong việc thiết kế giao diện, tính năng và trải nghiệm người dung, giao diện thân thiện và dễ sử dụng, đáp ứng các nhu cầu cụ thể của người dùng.

**Tương tác và chia sẻ thông tin**: Có thể tạo ra các tính năng cho phép người dùng tương tác và chia sẻ thông tin với nhau. Ví dụ, có thể cung cấp chức năng đánh giá, bình luận và chia sẻ trải nghiệm của người dùng với các thành phố và quốc gia.

**Tạo ra giá trị và thương hiệu**: Tự tạo ra một ứng dụng chứa thông tin về các thành phố và quốc gia giúp tạo ra giá trị và thương hiệu riêng của mình. Có thể xây dựng một sản phẩm độc đáo và nổi bật trên thị trường, thu hút sự quan tâm và lòng tin của người dùng.

*1.2.3 Đối tượng:*

Đối tượng của ứng dụng chứa thông tin về các thành phố và quốc gia có thể là rất đa dạng và phụ thuộc vào mục đích cụ thể của ứng dụng. Dưới đây là một số đối tượng tiềm năng mà ứng dụng hướng tới:

**Du khách**: Ứng dụng có thể hướng tới những người đang lên kế hoạch cho chuyến du lịch hoặc muốn tìm hiểu thông tin về các thành phố và quốc gia trước khi đi. Đối tượng này có thể muốn biết về danh lam thắng cảnh, điểm đến du lịch nổi tiếng, thông tin về văn hóa và lịch sử, hoặc các hoạt động giải trí và ẩm thực trong các địa điểm đó.

**Người học**: Ứng dụng có thể hướng tới những người đang nghiên cứu hoặc học về địa lý, văn hóa, lịch sử hoặc các khía cạnh khác của các thành phố và quốc gia. Đối tượng này có thể là sinh viên, học sinh, nhà nghiên cứu hoặc những người muốn mở rộng kiến thức cá nhân.

**Người kinh doanh**: Ứng dụng có thể cung cấp thông tin về các thành phố và quốc gia để hỗ trợ người kinh doanh trong việc tìm hiểu thông tin thị trường, khả năng đầu tư, văn hóa kinh doanh và các yếu tố khác liên quan. Điều này có thể giúp họ đưa ra quyết định kinh doanh thông minh và nắm bắt cơ hội trong các thị trường quốc tế.

**Người quan tâm đến địa lý và văn hóa**: Ứng dụng có thể hướng tới những người có sở thích về địa lý, văn hóa và lịch sử. Đối tượng này có thể là những người đam mê khám phá và muốn tìm hiểu về các địa điểm quốc tế, những người yêu thích đọc sách, xem phim hoặc tìm hiểu về các nền văn hóa khác nhau trên thế giới.

**Cộng đồng người dùng**: Ứng dụng có thể hướng tới việc tạo ra một cộng đồng người dùng nơi mọi người có thể chia sẻ và trao đổi thông tin, trải nghiệm và lời khuyên về các thành phố và quốc gia. Điều này giúp tạo ra một môi trường tương tác và hỗ trợ cho những người có cùng sở thích và quan tâm.

*1.2.4 Phạm vi:*

Phạm vi của ứng dụng chứa thông tin về các thành phố và quốc gia có thể được xác định bởi các yếu tố sau:

**Thành phố và quốc gia được bao gồm**: Bạn có thể quyết định phạm vi địa lý mà ứng dụng của bạn sẽ bao gồm. Ví dụ, bạn có thể tập trung vào một số thành phố và quốc gia nổi tiếng hoặc mở rộng phạm vi để bao gồm nhiều địa điểm trên toàn thế giới. Tùy thuộc vào mục đích và tầm nhìn của bạn, bạn có thể quyết định rất cụ thể hoặc rộng rãi về phạm vi địa lý.

**Thông tin chi tiết**: Có thể xác định mức độ chi tiết và đa dạng thông tin mà ứng dụng của bạn sẽ cung cấp. Điều này có thể bao gồm thông tin về địa lý, lịch sử, văn hóa, dân số, ngôn ngữ, điểm du lịch, ẩm thực và nhiều khía cạnh khác của các thành phố và quốc gia. Có thể quyết định cung cấp thông tin tổng quan hoặc chi tiết hơn tùy thuộc vào mục đích và phạm vi của ứng dụng.

**Tính năng và chức năng**: Xác định các tính năng và chức năng mà ứng dụng sẽ cung cấp. Ví dụ, ứng dụng có thể cho phép người dùng tìm kiếm, đọc đánh giá và chia sẻ trải nghiệm. Phạm vi tính năng và chức năng của ứng dụng sẽ phụ thuộc vào mục tiêu và khả năng.

**Đối tượng người dùng**: Phạm vi của ứng dụng cũng có thể được xác định bởi đối tượng người dùng muốn hướng tới. Ví dụ, nếu ứng dụng hướng tới du khách, phạm vi sẽ tập trung vào thông tin du lịch và các hoạt động giải trí. Nếu ứng dụng hướng tới người kinh doanh, phạm vi có thể tập trung vào thông tin thị trường và khả năng đầu tư.

*1.2.4 Ràng buộc tổng quan hệ thống:*

Dưới đây là những ràng buộc tổng quan về hệ thống của ứng dụng:

**Ràng buộc về nền tảng**: Ứng dụng có thể được phát triển cho các nền tảng khác nhau như di động (iOS, Android), máy tính để bàn (Windows, macOS), trang web, hoặc có thể là một ứng dụng đa nền tảng. Mỗi nền tảng có yêu cầu riêng về ngôn ngữ lập trình, framework và công nghệ, vì vậy ràng buộc này sẽ ảnh hưởng đến quyết định về kiến trúc và phạm vi của ứng dụng.

**Ràng buộc về dữ liệu**: Ứng dụng cần có một nguồn dữ liệu đáng tin cậy để cung cấp thông tin về các thành phố và quốc gia. Điều này có thể bao gồm việc thu thập dữ liệu từ các nguồn bên ngoài, như cơ quan chính phủ, tổ chức du lịch hoặc các nguồn dữ liệu công khai khác. Ràng buộc này đòi hỏi việc xác định và quản lý quy trình thu thập, xử lý và cập nhật dữ liệu.

**Ràng buộc về hiệu suất**: Ứng dụng cần đảm bảo hiệu suất tốt để cung cấp trải nghiệm người dùng mượt mà và nhanh chóng. Điều này đòi hỏi tối ưu hóa việc truy xuất và hiển thị dữ liệu, quản lý bộ nhớ và tài nguyên hệ thống, và xử lý các tác vụ nền.

**Ràng buộc về bảo mật**: Ứng dụng cần đảm bảo an toàn thông tin người dùng và dữ liệu. Điều này bao gồm việc xác thực và ủy quyền người dùng, mã hóa dữ liệu, bảo vệ khỏi các cuộc tấn công và lỗ hổng bảo mật, và tuân thủ các quy định và tiêu chuẩn liên quan đến bảo mật.

**Ràng buộc về giao diện người dùng**: Ứng dụng cần có giao diện người dùng thân thiện, dễ sử dụng và tương thích với các thiết bị và màn hình khác nhau. Giao diện người dùng cần được thiết kế một cách rõ ràng, dễ hiểu và hợp lý để cung cấp trải nghiệm tốt cho người dùng.

**Ràng buộc về quản lý và bảo trì**: Ứng dụng cần được quản lý và bảo trì để đảm bảo hoạt động liên tục và duy trì tính năng của nó. Ràng buộc này đòi hỏi xác định và triển khai quy trình kiểm tra, sửa lỗi và cập nhật ứng dụng, cũng như quản lý phiên bản và sao lưu dữ liệu.

# **2. CƠ SỞ LÝ THUYẾT**

## 2.1 TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Lập trình hướng đối tượng (OOP) là một phương pháp lập trình mà trong đó chương trình được tổ chức xung quanh các "đối tượng". Mỗi đối tượng là một thể hiện cụ thể của một "lớp", và chúng bao gồm dữ liệu (các thuộc tính) và các hành vi (các phương thức). OOP tập trung vào việc mô hình hóa thế giới thực bằng cách tạo ra các đối tượng, mỗi đối tượng đại diện cho một phần của thế giới thực, và mối quan hệ giữa chúng.

Trong Dart, một ngôn ngữ lập trình đa nền tảng được phát triển bởi Google, OOP được hỗ trợ mạnh mẽ.

Dưới đây là một số khái niệm cơ bản của OOP và cách chúng được thực hiện trong Dart:

- Lớp (Class): Là một bản thiết kế cho các đối tượng, định nghĩa cấu trúc dữ liệu và hành vi của chúng. Trong Dart, một lớp được định nghĩa bằng từ khóa class. Đối tượng (Object): Là một thể hiện cụ thể của một lớp. Đối tượng bao gồm dữ liệu và các phương thức được xác định trong lớp tương ứng.

- Encapsulation (Đóng gói): Là nguyên tắc giữ dữ liệu và mã xử lý liên quan cùng nhau trong một lớp, và ẩn chúng khỏi các lớp khác. Trong Dart, bạn có thể sử dụng các phạm vi truy cập (public, private) để điều chỉnh quyền truy cập vào các thành phần của lớp.

- Kế thừa (Inheritance): Là quá trình một lớp kế thừa các thuộc tính và phương thức từ một lớp khác. Dart hỗ trợ kế thừa đơn kế thừa (single inheritance), một lớp chỉ có thể kế thừa từ một lớp cha.

- Đa hình (Polymorphism): Là khả năng của một đối tượng để hiển thị các hành vi khác nhau tùy thuộc vào ngữ cảnh. Trong Dart, đa hình thường được thực hiện thông qua kỹ thuật ghi đè phương thức (method overriding).

Với Dart, bạn có thể sử dụng OOP để xây dựng các ứng dụng có cấu trúc, dễ bảo trì và mở rộng. Dart cung cấp cú pháp và công cụ mạnh mẽ để triển khai các khái niệm OOP một cách linh hoạt và hiệu quả.

### *2.1.1. Môi trường lập trình*

Android Studio là một môi trường phát triển tích hợp (IDE) được phát triển bởi Google, được tối ưu hóa để phát triển ứng dụng Android. Đây là một trong những công cụ phát triển phổ biến nhất cho lập trình Android, và nó cung cấp một loạt các tính năng và công cụ giúp lập trình viên phát triển ứng dụng di động một cách hiệu quả.

Dưới đây là một số điểm nổi bật của Android Studio:

- Tích hợp với Android SDK: Android Studio đi kèm với Android SDK, bao gồm một bộ công cụ phát triển đầy đủ để xây dựng, biên dịch và gỡ lỗi ứng dụng Android. Việc tích hợp này giúp lập trình viên tiết kiệm thời gian và công sức trong việc cài đặt và cấu hình môi trường phát triển.

- Hỗ trợ cho Kotlin và Java: Android Studio hỗ trợ lập trình ứng dụng Android bằng cả ngôn ngữ lập trình Kotlin và Java. Kotlin là một ngôn ngữ lập trình hiện đại được Google công nhận là chính thức cho việc phát triển ứng dụng Android.

- Layout Editor: Android Studio đi kèm với một trình biên tập giao diện đồ họa mạnh mẽ cho phép lập trình viên thiết kế giao diện người dùng của ứng dụng một cách trực quan. Trình biên tập này cho phép bạn kéo và thả các thành phần UI, xem trước giao diện trên nhiều thiết bị, và tùy chỉnh các thuộc tính.

- Hỗ trợ Git và Quản lý Phiên bản: Android Studio tích hợp với Git và các hệ thống quản lý phiên bản khác, giúp bạn theo dõi và quản lý mã nguồn của dự án một cách dễ dàng. Nó cung cấp các công cụ cho việc commit, push, pull, và xem lịch sử thay đổi.

- Gỡ lỗi và Kiểm thử: Android Studio cung cấp một loạt các công cụ để gỡ lỗi và kiểm thử ứng dụng. Bạn có thể chạy ứng dụng trên các thiết bị thực, máy ảo, hoặc trình giả lập Android cung cấp sẵn.

- Hỗ trợ cho Flutter và Dart: Bạn có thể sử dụng Android Studio để phát triển ứng dụng di động sử dụng Flutter framework và ngôn ngữ lập trình Dart.

Android Studio là một công cụ mạnh mẽ cho việc phát triển ứng dụng Android, với các tính năng và công cụ cần thiết để xây dựng ứng dụng chất lượng cao và hiệu quả.

### *2.1.2 Ngôn ngữ lập trình Dart*

Ngôn ngữ lập trình Dart là một ngôn ngữ lập trình được phát triển bởi Google. Ban đầu, Dart được thiết kế nhằm mục đích giải quyết các thách thức trong việc phát triển các ứng dụng web lớn và phức tạp. Tuy nhiên, sau này, Dart đã mở rộng phạm vi của mình để hỗ trợ phát triển ứng dụng di động và máy chủ.

Dưới đây là một số điểm nổi bật về Dart:

- Dart là ngôn ngữ hướng đối tượng: Dart hỗ trợ các tính năng cơ bản của lập trình hướng đối tượng như class, inheritance, encapsulation, và polymorphism. Điều này giúp cho việc tổ chức mã nguồn, tái sử dụng và bảo trì trở nên dễ dàng hơn.

- Dart là ngôn ngữ có cú pháp đơn giản: Dart có cú pháp gần gũi và dễ hiểu, giúp cho người mới học lập trình cũng như lập trình viên có kinh nghiệm có thể nhanh chóng làm quen và sử dụng.

- Dart hỗ trợ phát triển ứng dụng đa nền tảng: Dart không chỉ được sử dụng để phát triển ứng dụng web mà còn có thể được sử dụng để phát triển ứng dụng di động và máy chủ. Đặc biệt, sử dụng với Flutter framework, Dart là một trong những công nghệ hàng đầu cho phát triển ứng dụng di động đa nền tảng.

- Dart có hệ thống quản lý bộ nhớ tự động (Garbage Collection): Dart sử dụng garbage collection để quản lý bộ nhớ tự động, giúp giảm thiểu việc phải lo lắng về việc giải phóng bộ nhớ.

- Dart hỗ trợ tính năng async/await: Dart cung cấp hỗ trợ cho việc xử lý bất đồng bộ thông qua việc sử dụng async/await, giúp cho việc lập trình với các tác vụ không đồng bộ trở nên đơn giản hơn.

Dart đang ngày càng trở nên phổ biến và được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng, đặc biệt là trong việc phát triển ứng dụng di động sử dụng Flutter framework.

## 2.2 TỔNG QUAN VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Nó cung cấp một loạt các dịch vụ đám mây để xây dựng và quản lý ứng dụng, bao gồm cả hệ quản trị cơ sở dữ liệu (Database) Firebase. Hệ quản trị cơ sở dữ liệu Firebase có các đặc điểm và ưu điểm sau:

**Cơ sở dữ liệu thời gian thực**: Firebase Database là một cơ sở dữ liệu thời gian thực (real-time database), có nghĩa là dữ liệu được đồng bộ trực tiếp và tự động giữa client và server. Khi dữ liệu thay đổi, các thiết bị kết nối sẽ tự động nhận được cập nhật ngay lập tức, tạo ra trải nghiệm trực tiếp và tương tác cho người dùng.

**Cấu trúc dữ liệu linh hoạt**: Firebase Database sử dụng mô hình cấu trúc dữ liệu JSON, cho phép bạn lưu trữ và truy xuất dữ liệu dễ dàng theo cấu trúc cây. Bạn có thể lưu trữ dữ liệu dưới dạng các cặp khóa-giá trị, và dữ liệu có thể chứa các đối tượng lồng nhau. Đám mây và mở rộng linh hoạt: Firebase Database hoạt động trên nền tảng đám mây của Google, điều này cho phép dữ liệu được lưu trữ và quản lý một cách an toàn, tin cậy và mở rộng linh hoạt. Bạn không cần phải quản lý cơ sở hạ tầng cơ sở dữ liệu của riêng mình, mà có thể tập trung vào phát triển ứng dụng của mình.

**Tích hợp sâu với các dịch vụ khác của Firebase**: Firebase Database tích hợp tốt với các dịch vụ khác của Firebase như xác thực người dùng, thông báo đẩy, phân tích và lưu trữ tệp. Điều này giúp bạn xây dựng các ứng dụng phong phú và tương tác, kết hợp nhiều tính năng của Firebase.

**Hỗ trợ đa nền tảng**: Firebase Database không chỉ hỗ trợ các ứng dụng di động trên nền tảng iOS và Android, mà còn hỗ trợ các ứng dụng web. Điều này giúp bạn xây dựng ứng dụng đa nền tảng và chia sẻ

# **3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## 3.1 XÁC ĐỊNH YÊU CẦU

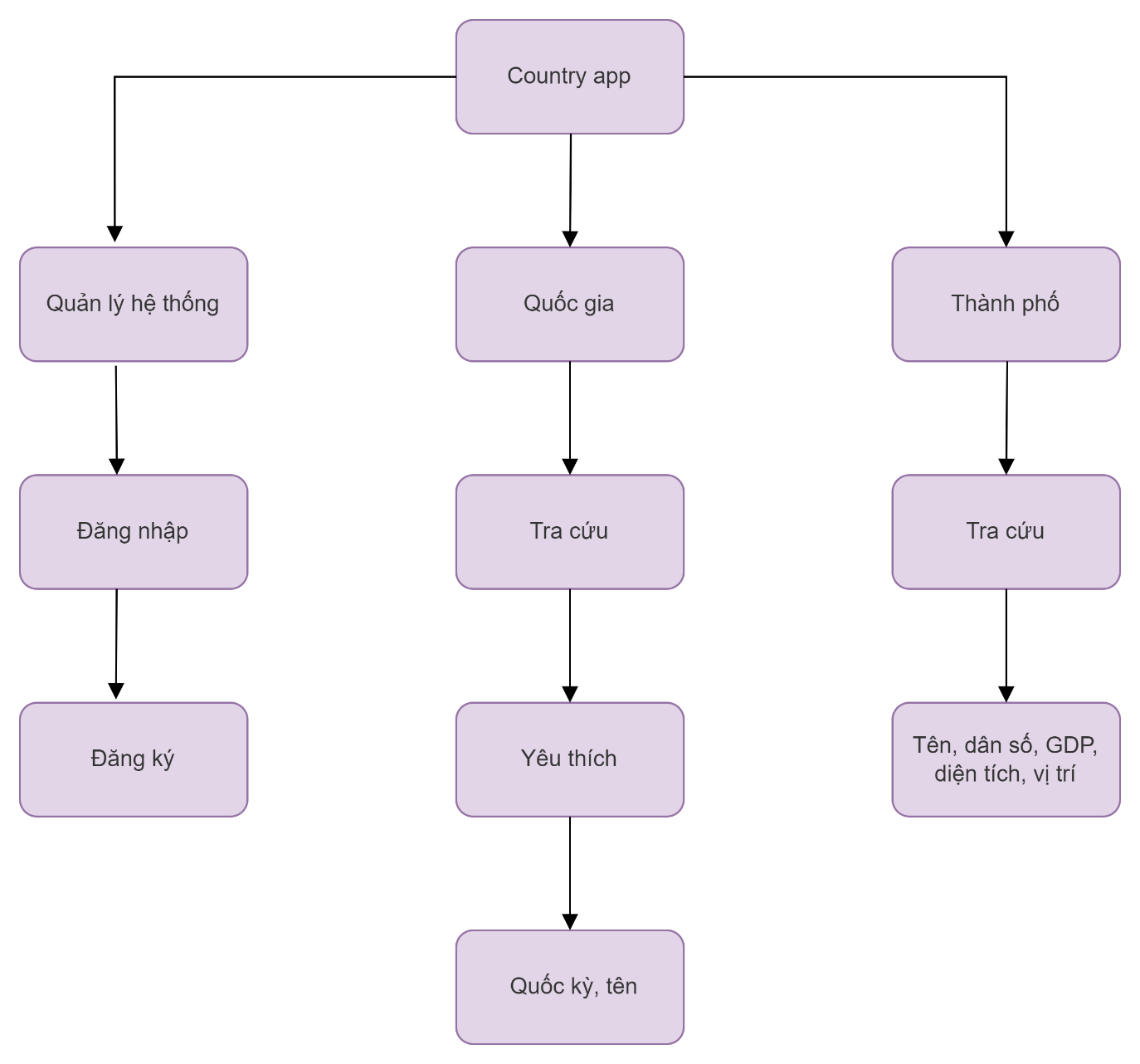
### *3.1.1 Yêu cầu chức năng*

* Ứng dụng phải cập nhập, hiển thị được tất cả các thông tin chi tiết về thành phố
* Cung cấp, tra cứu thông tin thành phố, quốc gia
* Hiển thị mục yêu thích của quốc gia

### *3.1.2 Yêu cầu hệ thống*

* Hệ thống sử dụng hệ quản trị cơ sở dữ liệu đủ lớn để đáp ứng số lượng người dùng, truy cập.
* Máy chủ có khả năng tính toán nhanh, chính xác, lâu dài, bảo mật.
* Hệ thống mạng đáp ứng khả năng truy cập lớn.
* Thông tin có tính đồng bộ, phân quyền quản lý chặt chẽ.
* Bảo mật tốt cho người quản trị hệ thống.

## 3.2 MÔ HÌNH HÓA



*Hình 3.1* Biểu đồ phân rã chức năng (BFD)

## 3.3. FIREBASE SỬ DỤNG

* Firebase Authentication

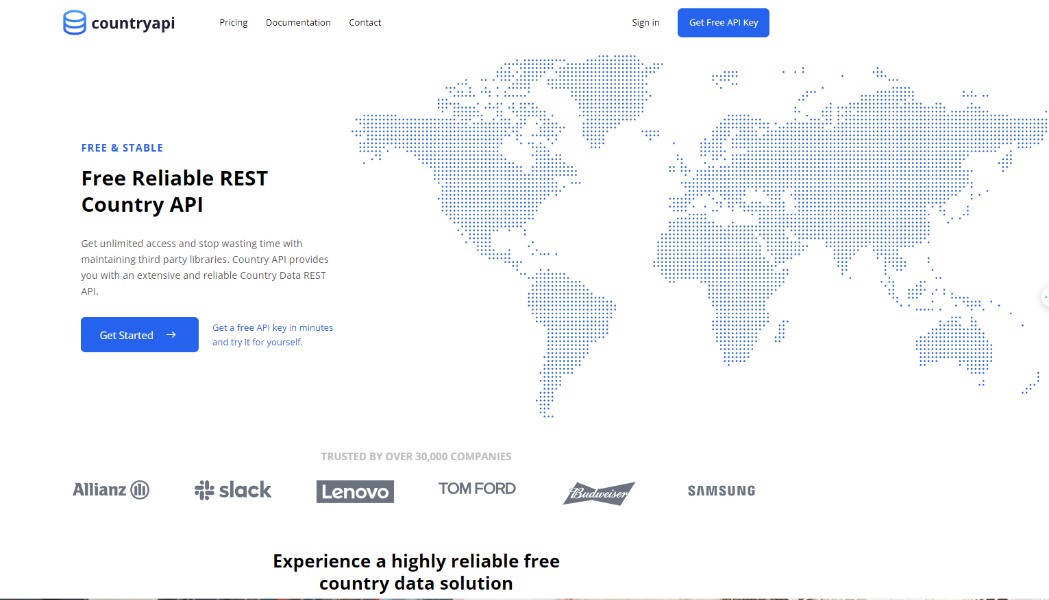
-Firebase Authentication nhằm giúp việc xây dựng hệ thống xác thực an toàn trở nên dễ dàng, đồng thời cải thiện trải nghiệm đăng nhập và quá trình tạo tài khoản cho người dùng cuối. Nó cung cấp một giải pháp xác thực toàn diện, hỗ trợ tài khoản email và mật khẩu, xác thực qua số điện thoại, đăng nhập bằng Google, Twitter, Facebook, GitHub, và nhiều hơn nữa.

A screen shot of a phone

Description automatically generated with low confidence

* Trang web cung cấp API: [Free Country API - https://countryapi.io/](Free%20Country%20API%20-%20https://countryapi.io/)

CountryAPI.com là một dịch vụ API cung cấp thông tin về quốc gia trên toàn thế giới. Nó cung cấp các dữ liệu như thông tin dân số, diện tích, thủ đô, ngôn ngữ chính thức, và nhiều thông tin khác liên quan đến các quốc gia trên thế giới. Với CountryAPI.com, nhà phát triển có thể sử dụng API để truy xuất dữ liệu về quốc gia từ hàng ngàn quốc gia khác nhau và tích hợp vào ứng dụng của mình. Dịch vụ này cung cấp một giao diện dễ sử dụng và hỗ trợ các yêu cầu API thông qua giao thức HTTP.



## 3.4. CÁC THÀNH PHẦN CHỨC NĂNG CỦA HỆ THỐNG

### *3.4.1. Chức năng đăng nhập, đăng ký hệ thống*

Chức năng đăng nhập và đăng ký trong hệ thống là một phần không thể thiếu trong mọi ứng dụng hoặc hệ thống, giúp người dùng xác thực và quản lý tài khoản cá nhân của họ. Điều này mang lại nhiều lợi ích cho người dùng, bao gồm tính bảo mật, quyền riêng tư và trải nghiệm cá nhân hóa.

Khi sử dụng chức năng này, người dùng cần cung cấp thông tin đăng nhập hoặc thông tin đăng ký mới như tên đăng nhập, mật khẩu, và thông tin cá nhân để xác thực danh tính hoặc tạo tài khoản mới. Sau khi hoàn tất quá trình đăng nhập hoặc đăng ký, người dùng có thể truy cập vào các tính năng và dịch vụ của ứng dụng.

Chức năng đăng nhập và đăng ký đảm bảo rằng chỉ có người dùng hợp lệ và được ủy quyền mới có thể truy cập và sử dụng hệ thống, đồng thời bảo vệ dữ liệu và thông tin quan trọng khỏi truy cập trái phép.

### *3.4.2. Chức năng tìm kiếm địa điểm*

Chức năng tra cứu địa điểm trong ứng dụng CountryApp cho phép người dùng tìm kiếm thông tin về một thành phố cụ thể trong một quốc gia bất kỳ. Người dùng có thể nhập tên quốc gia hoặc thành phố vào công cụ tra cứu và ứng dụng sẽ hiển thị thông tin liên quan về quốc gia, thành phố đó.

Thông tin này bao gồm thông tin về dân số, diện tích, GDP, vị trí và các thông tin khác liên quan đến thành phố đó. Chức năng này giúp người dùng dễ dàng tìm hiểu về một thành phố cụ thể trước khi lên kế hoạch cho chuyến đi hoặc các hoạt động liên quan đến địa điểm đó.

Chức năng tra cứu địa điểm giúp người dùng nắm bắt thông tin về một quốc gia một cách dễ dàng và thuận tiện, giúp họ có những quyết định và kế hoạch phù hợp với thông tin về quốc gia đó.

### *3.4.3. Chức năng hiển thị thông tin*

Chức năng hiển thị thông tin của các thành phố trong một quốc gia trong ứng dụng CountryApp cho phép người dùng xem thông tin chi tiết về các thành phố cụ thể. Người dùng có thể lựa chọn quốc gia từ danh sách có sẵn và sau đó xem danh sách các thành phố trong quốc gia đó.

Thông tin được hiển thị bao gồm tên của thành phố, dân số, diện tích, GPD và các thông tin địa lý khác. Chức năng này giúp người dùng hiểu rõ hơn về các thành phố trong quốc gia mình quan tâm, từ đó có thể đưa ra quyết định hoặc lên kế hoạch cho chuyến đi một cách thông minh.

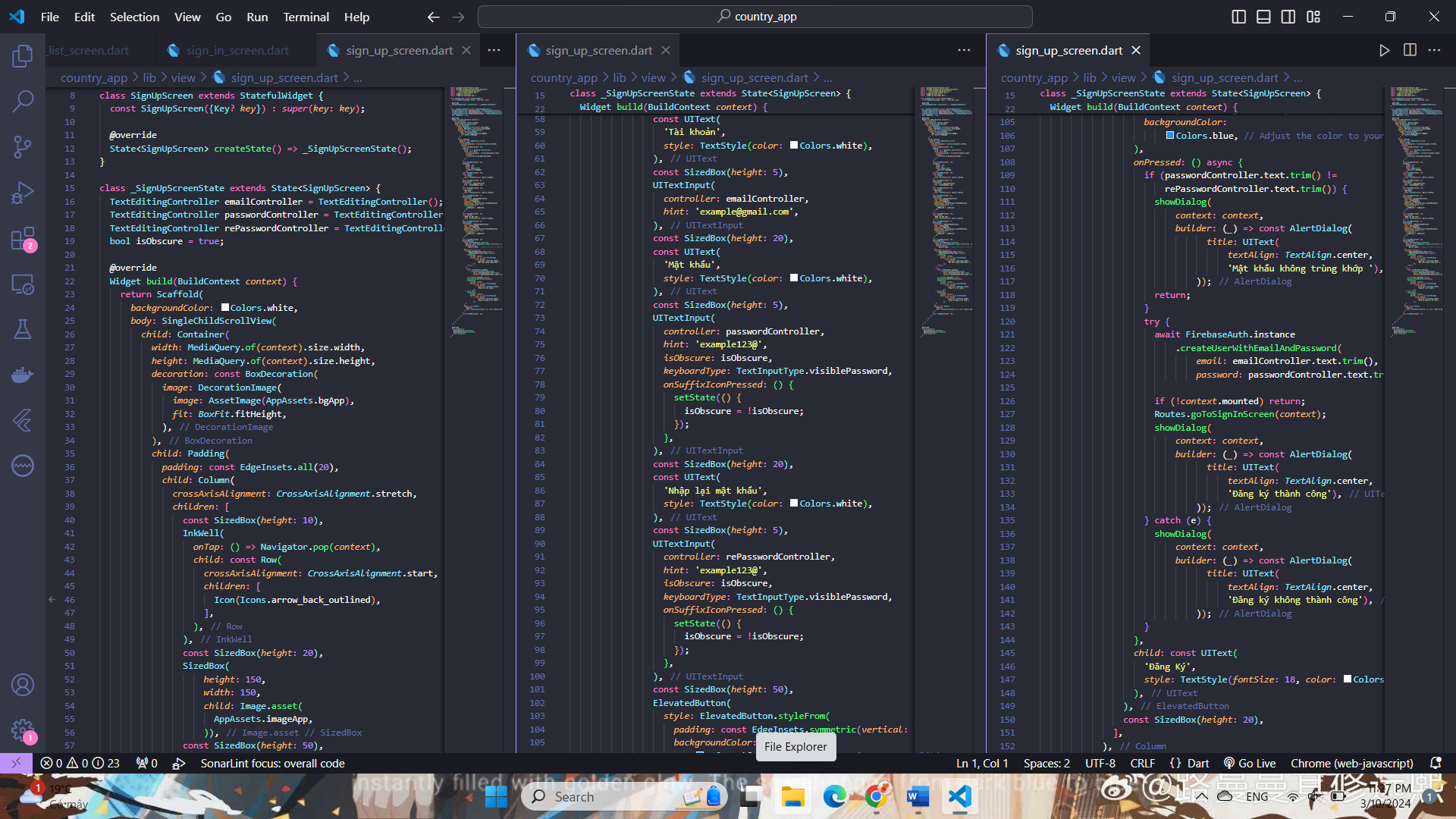
Bằng cách này, người dùng có thể dễ dàng so sánh các thành phố trong cùng một quốc gia và đưa ra quyết định dựa trên thông tin được cung cấp. Chức năng hiển thị thông tin của các thành phố trong một quốc gia giúp người dùng có cái nhìn toàn diện và chính xác về địa lý và dân số của quốc gia mình quan tâm.

### *3.4.4. Chức năng hiển thị quốc gia yêu thích*

## 3.5 SOURCE CODE CỦA CHƯƠNG TRÌNH

### *3.5.1. Source code SignUP*

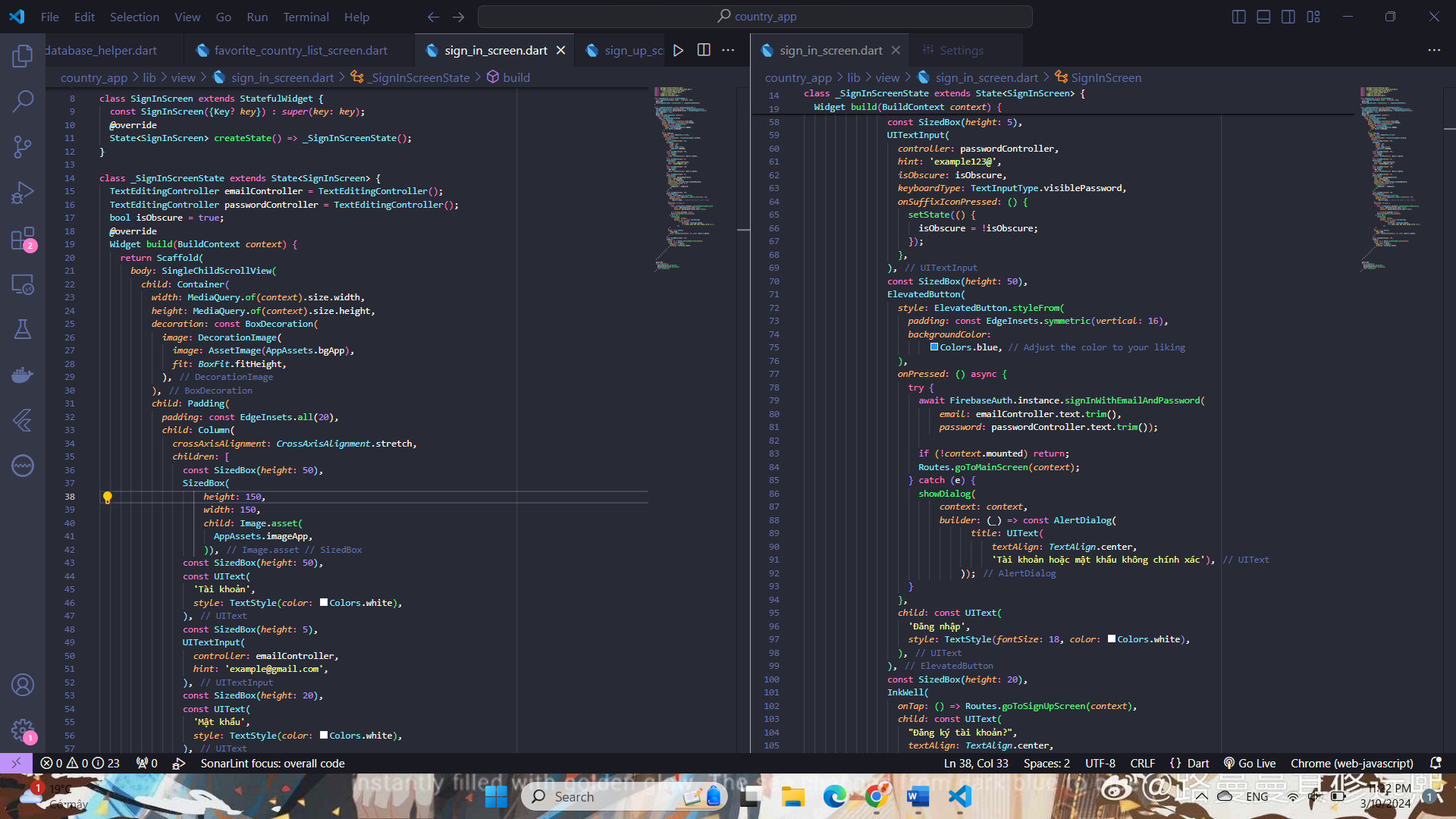
* Chức năng: Đăng kí người dùng mới



*Hình 3.2* Source code SignUP

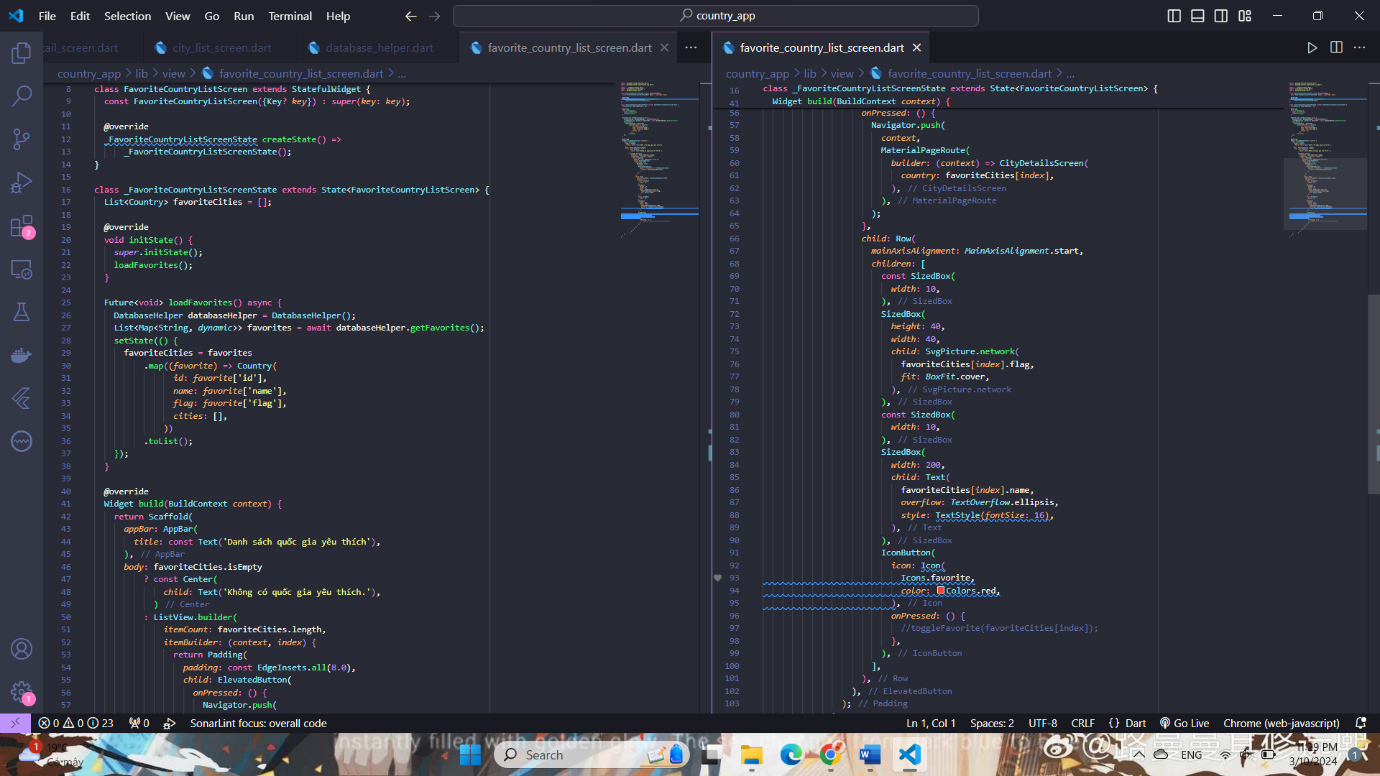
### *3.5.2 Source code SignIn*

* Chức năng: Đăng nhập tài khoản người dùng



*Hình 3.3* Source code SignIn

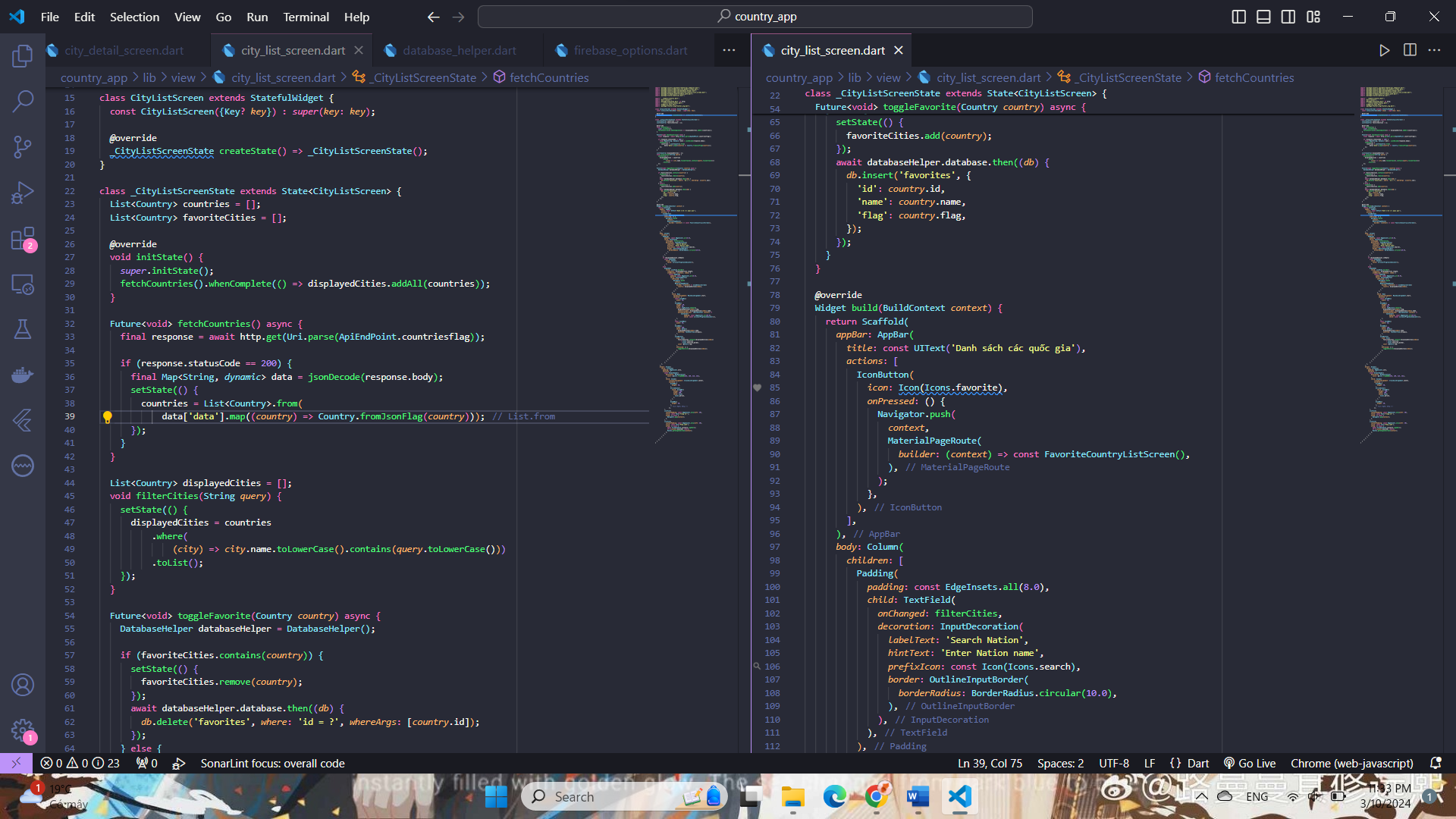
### *3.5.3 Source code quốc gia yêu thích*

* Chức năng: Hiển thị quốc gia yêu thích

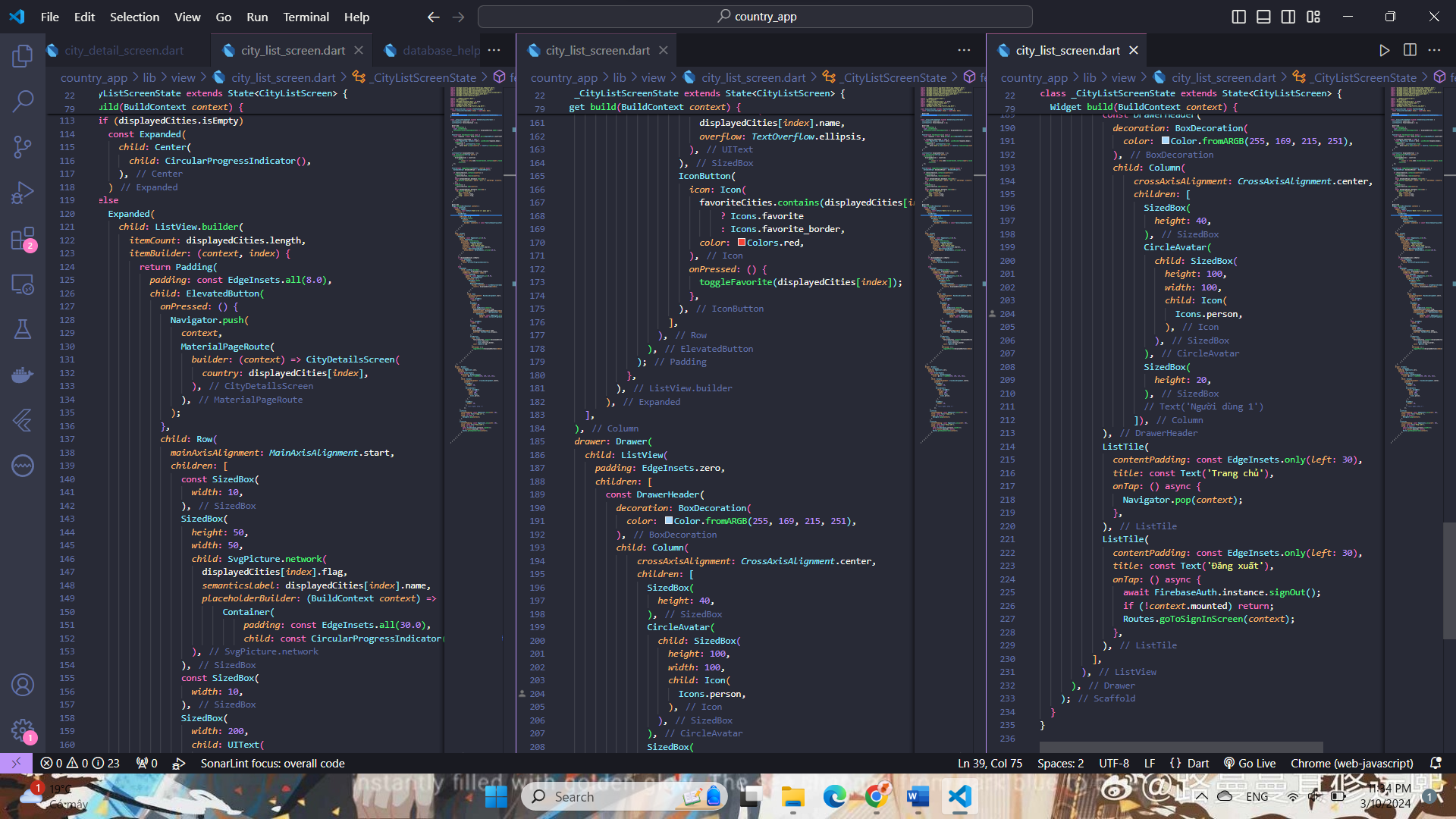
*Hình 3.4* Source code quốc gia yêu thích

### *3.5.4. Source code danh sách quốc gia*

* Chức năng: Hiển thị danh sách tất cả các qquoocs gia trên thế giới, tra cứu quốc gia

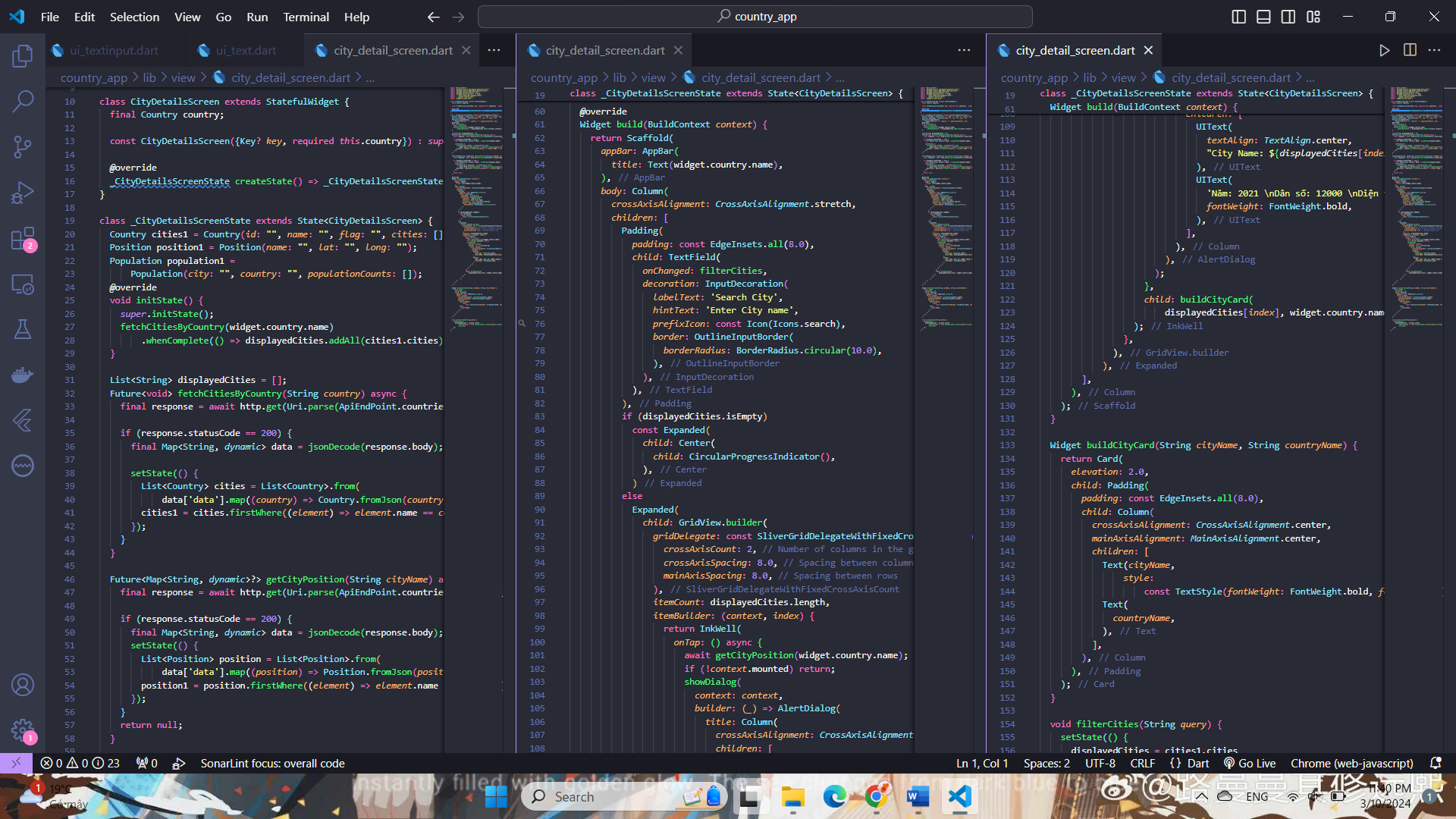


*Hình 3.5* Source code danh sách quốc gia



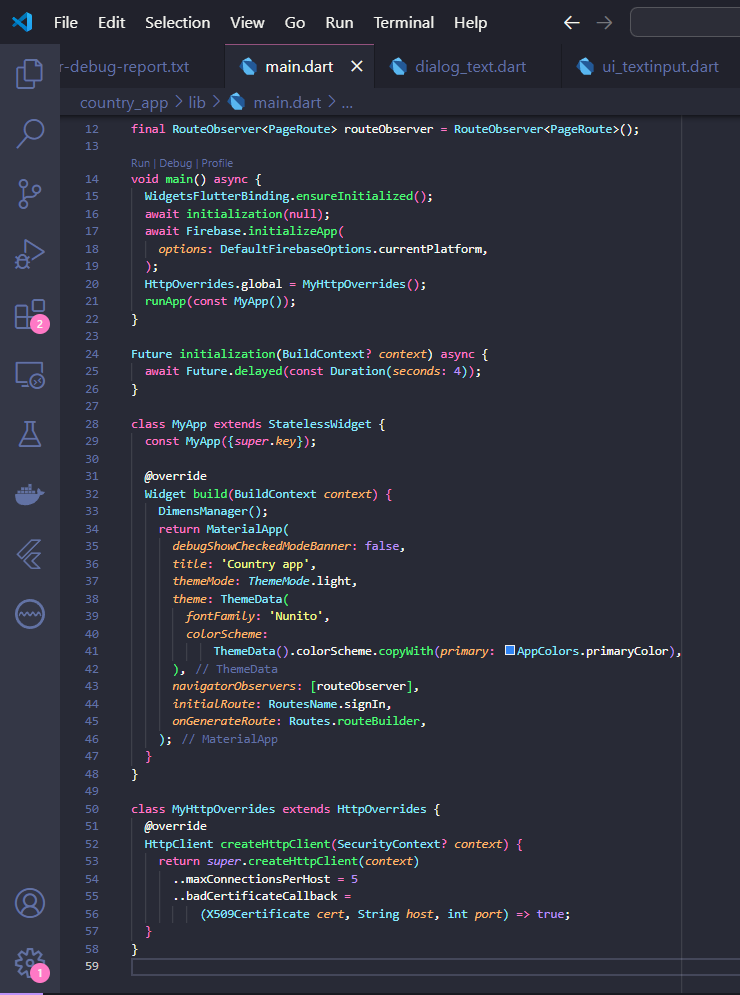
*Hình 3.6* Source code danh sách quốc gia

### *3.5.5. Source code danh sách thành phố*

* Chức năng: Hiển thị danh sách thành phố, tra cứu thành phố, hiển thị thông tin chi tiết của từng thành phố

*Hình 3.7* Source code danh sách thành phố

### *3.5.6 Source code Main*.

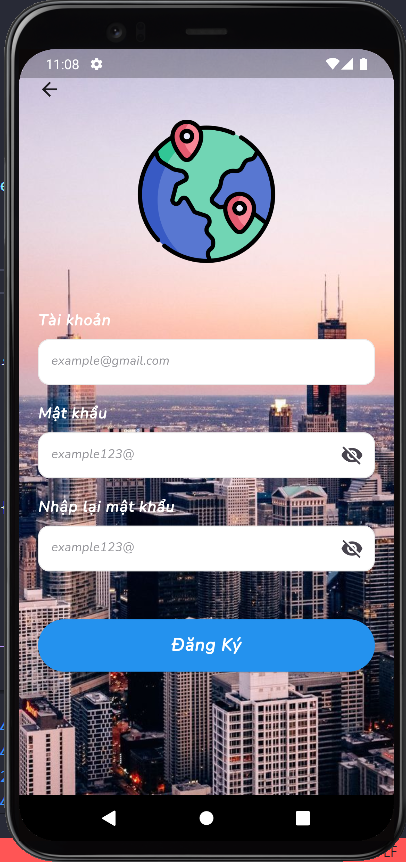


Hình 3.8 Source code main

## 3.6 THIẾT KẾ GIAO DIỆN HỆ THỐNG

Tiếp theo là phần thiết kế form cho chường trình sao cho hệ thống hoạt động có hiệu quả. Phần mềm phải hoạt động tốt, không trục trặc về thông tin, hạn chế sai sót trong các thao tác kỹ thuật

### *3.6.1. Form đăng kí- tạo tài khoản*



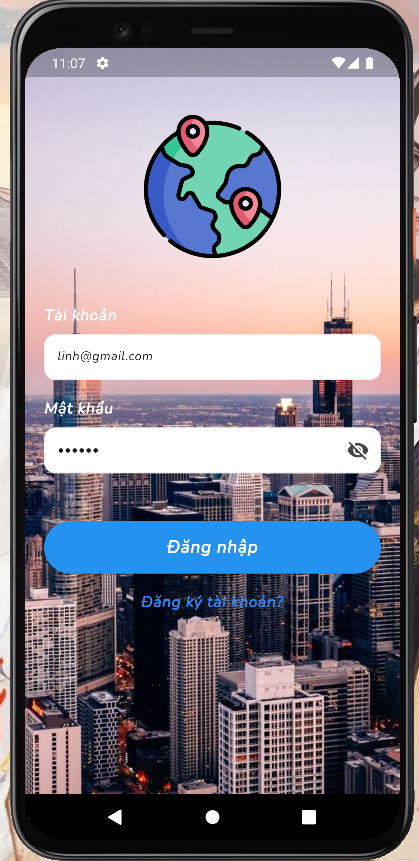
*Hình 3.9* Form đăng kí- tạo tài khoản

Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Đăng ký | Đăng kí tài khoản | Touch |
| 🡨 | Chuyển đến giao diên đăng nhập | Touch |

Bảng 3.1 Chức năng Form đăng kí

### *3.6.2 Form đăng nhập của phần mềm*



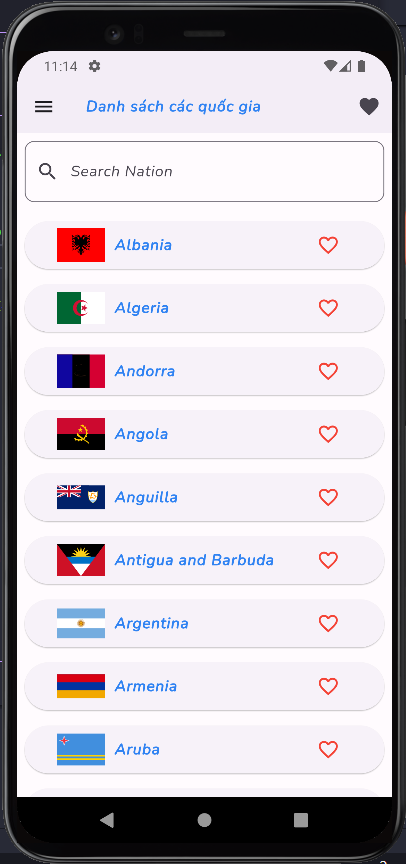
*Hình 3.10* Form giao diện Đăng nhập

Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Đăng nhập | Đăng nhập vào trang chủ | Touch |
| Đăng ký tài khoản | Chuyển đến giao diên Đăng ký | Touch |

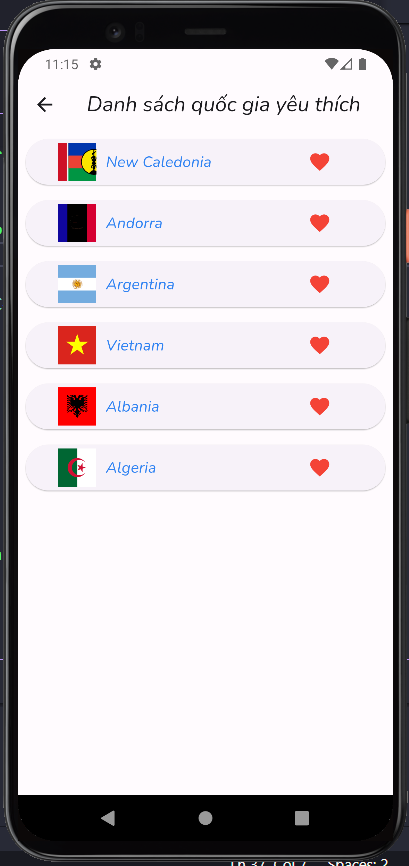
Bảng 3.2 Chức năng đăng nhập của phần mềm

### *3.6.3 Form trang chủ ứng dụng người dùng*



*Hình 3.11* Form trang chủ ứng dụng

### *3.6.4 Form giao diện lưu trữ quốc gia yêu thích*



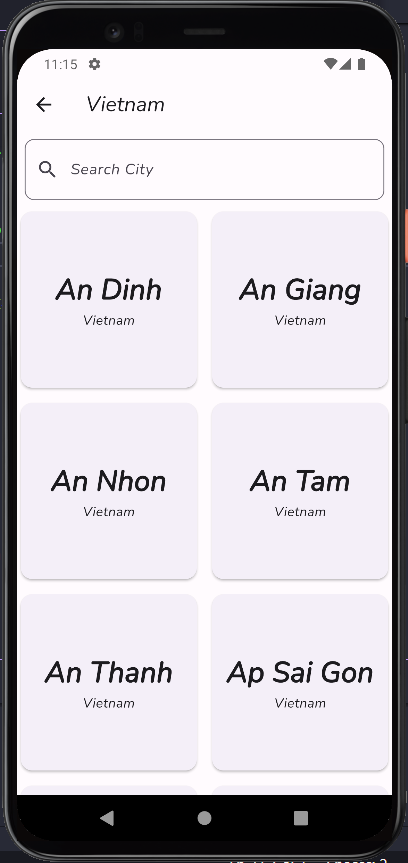
*Hình 3.12 From* giao diện lưu trữ quốc gia yêu thích

Chức năng nút lệnh:

|  |  |  |
| --- | --- | --- |
| Tên nút lệnh | Ý nghĩa | Sự kiện |
| Heart | Lưu quốc gia yêu thích | Touch |

Bảng 3.3 Chức năng lưu trữ quốc gia yêu thích

### *3.6.5 Form giao diện danh sách thành phố*



*Hình 3.13* Form giao diện danh sách thành phố

### *3.6.6 Form giao diện thông tin thành phố*

*Hình 3.14* Form giao diện thông tin thành phố

# **KẾT LUẬN**

## 4.1 KẾT LUẬN ĐỀ TÀI

### **4.1.1 Đánh giá chung ưu nhược điểm của hệ thống**

**Ưu điểm:**

* **Tiện ích cho người dùng**: Country-App cung cấp thông tin về các quốc gia và thành phố một cách nhanh chóng và thuận tiện cho người dùng.
* **Khả năng mở rộng**: Hệ thống được thiết kế để dễ dàng mở rộng và cập nhật thông tin mới về các quốc gia và thành phố.
* **Tính đa dạng**: Country-App cung cấp một loạt các thông tin đa dạng, từ thông tin địa lý đến văn hóa và lịch sử của các quốc gia và thành phố.

**Nhược điểm:**

* **Dữ liệu có thể không luôn được cập nhật**: Do tính chất biến động của thông tin địa lý và văn hóa, có thể xảy ra tình trạng thông tin không còn chính xác do dữ liệu không được cập nhật đầy đủ.
* **Yêu cầu kết nối internet**: Sử dụng Country-App yêu cầu kết nối internet liên tục, điều này có thể gây khó khăn cho người dùng trong các vùng có tín hiệu internet không ổn định.

### **4.1.2 Hướng phát triển và mở rộng đề tài**

* **Cải thiện dữ liệu**: Tiếp tục cập nhật và kiểm tra thông tin để đảm bảo tính chính xác và độ tin cậy của dữ liệu.
* **Phát triển tính năng**: Thêm các tính năng mới như bản đồ tương tác, hướng dẫn du lịch, thông tin về các sự kiện văn hóa hoặc sự kiện quan trọng của các quốc gia và thành phố.
* **Tích hợp công nghệ mới**: Áp dụng các công nghệ mới như trí tuệ nhân tạo để cải thiện trải nghiệm người dùng và tăng tính tương tác của ứng dụng.

## 4.2 KẾT LUẬN

Country-App đã đem lại một cách tiếp cận mới mẻ và tiện lợi trong việc truy cập thông tin về các quốc gia và thành phố trên thế giới. Qua quá trình phát triển và triển khai, chúng ta đã nhận thấy rõ rằng ứng dụng này có những ưu điểm đáng kể cùng với một số thách thức cần vượt qua.

Ưu điểm của Country-App nằm ở sự tiện lợi và linh hoạt mà nó mang lại cho người dùng. Khả năng cập nhật dữ liệu và tính năng mở rộng của ứng dụng cung cấp một cơ sở vững chắc cho việc nâng cao trải nghiệm người dùng trong tương lai. Việc tích hợp các thông tin về văn hóa, lịch sử và địa lý của các quốc gia và thành phố cũng giúp ứng dụng trở nên phong phú và đa chiều.

Tuy nhiên, cũng cần nhìn nhận các thách thức mà Country-App đang đối mặt. Việc duy trì và cập nhật dữ liệu là một vấn đề quan trọng, đặc biệt là trong bối cảnh thông tin thay đổi liên tục. Sự phụ thuộc vào kết nối internet cũng có thể tạo ra một rào cản cho một số người dùng, đặc biệt là ở các khu vực có tín hiệu internet không ổn định.

Để phát triển tiếp Country-App, chúng ta cần tiếp tục cải thiện dữ liệu, tối ưu hóa trải nghiệm người dùng và tích hợp các công nghệ mới nhất. Mục tiêu cuối cùng là tạo ra một ứng dụng hoàn thiện, hữu ích và phản ánh đầy đủ về sự đa dạng và vẻ đẹp của thế giới xung quanh chúng ta.