

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA

BÀI TẬP LỚN

TÊN HỌC PHẦN: LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG

LỚP: N03

HỌC KỲ II NĂM HỌC 2022 - 2023

ĐỀ TÀI: XÂY DỰNG CHATGPT APP SỬ DỤNG OPENAI API

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh
1	21010625	Nguyễn Đại Phát	24/04/2003
2	21013118	Phùng Hồng Phước	17/04/2003
3	21010662	Đỗ Xuân Thành	23/08/2003

Hà Nội 2023

LỜI NÓI ĐẦU

Trong thời đại số hóa phát triển nhanh chóng, sự tiến bộ của trí tuệ nhân tạo đã đem lại nhiều cơ hội và thách thức mới cho lĩnh vực phát triển ứng dụng di động. Và trong bối cảnh đó, ứng dụng chatbot đã trở thành một phần không thể thiếu trong việc tương tác với người dùng.

Trong báo cáo này, chúng tôi tập trung vào việc xây dựng một ứng dụng chatbot đột phá sử dụng công nghệ ChatGPT trên nền tảng di động bằng Flutter. Flutter, một framework phát triển ứng dụng di động đa nền tảng của Google, cung cấp một cách tiếp cận nhanh chóng và hiệu quả cho việc xây dựng giao diện người dùng hấp dẫn và tương thích trên nhiều thiết bị.

Ứng dụng chatbot mà chúng tôi đề xuất không chỉ đơn thuần là một công cụ giao tiếp, mà còn là một trợ lý ảo thông minh có khả năng hiểu và trả lời các câu hỏi, cung cấp thông tin hữu ích và tương tác với người dùng một cách tự nhiên và trực quan.

Chúng tôi sử dụng mô hình ChatGPT, một mô hình ngôn ngữ dựa trên trí tuệ nhân tạo của OpenAI, để xây dựng khả năng tương tác thông minh cho chatbot. Mô hình này được huấn luyện trên một lượng lớn dữ liệu và có khả năng tạo ra câu trả lời tự nhiên và logic dựa trên ngữ cảnh.

Báo cáo này sẽ trình bày chi tiết về quy trình xây dựng ứng dụng, bao gồm cấu trúc và thiết kế giao diện người dùng, tích hợp mô hình ChatGPT vào ứng dụng.

Chúng tôi hy vọng rằng báo cáo này sẽ mang lại cho bạn cái nhìn toàn diện về quá trình phát triển một ứng dụng chatbot thông minh trên nền tảng di động bằng Flutter, và sẽ cung cấp những kiến thức và kỹ năng cần thiết để bạn xây dựng các ứng dụng tương tự trong tương lai.

MỤC LỤC

LỜI NÓI ĐẦU	2
MỤC LỤC	3
MỤC LỤC HÌNH ẢNH	4
BẢNG CÁC TỪ VIẾT TẮT	6
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	7
1.1. Đặt vấn đề	7
1.2. Định hướng giải pháp	8
CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG	10
2.1. Dart	10
2.2. Flutter	10
2.3. Packages	11
2.5. ChatGPT API	12
CHƯƠNG 3. XÂY DỰNG GIAO DIỆN VÀ KẾT NỐI API	15
3.1. Triển khai kỹ thuật	15
3.1.1. Công cụ sử dụng	15
3.1.2. Kết quả đạt được	15
3.1.3. Kiểm thử	17
3.2. Xây dựng giao diện và kết nối API	19
KẾT LUẬN	26
DANH MỤC TÀI LIỆU THAM KHẢO	28

BẢNG PHÂN CÔNG NHIỆM VỤ

Họ tên	Các công việc/đóng góp chính
Nguyễn Đại Phát	Báo cáo: Chương 3 và phần còn lại. Code: api_service.dart, chats_provider.dart và models_provider.dart.
Phùng Hồng Phước	Báo cáo: Chương 2. Code: main.dart, constants.dart và chat_screen.dart.
Nguyễn Đại Phát	Báo cáo: Chương 1. Code: chat_widget.dart và model_widget.dart.

MỤC LỤC HÌNH ẢNH

Hình 1: Kiểm tra API keys	18
Hình 2: Giao diện ChatGPT sai key API.....	19
Hình 3: Thiết kế tổng quan.....	21
Hình 4: Giao diện app ChatGPT.....	22
Hình 5: Giao diện các model.....	23

BẢNG CÁC TỪ VIẾT TẮT

(Nếu có)

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	GPT	Generative Pre-trained Transformer
2	API	Application Programming Interface
3	AI	Artificial Intelligence
4	VM	Virtual Machine
5	IOS	iPhone Operating System
6	HTTP	Hypertext Transfer Protocol
7	JSON	JavaScript Object Notation
8	IDE	Integrated Development Environment

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Đặt vấn đề

Trong những năm gần đây, Trí tuệ nhân tạo (AI) đã trở thành một công nghệ ngày càng phổ biến và có mức ảnh hưởng đáng kể đối với con người và xã hội. AI được sử dụng trong nhiều lĩnh vực, từ công nghiệp, y tế, giao thông, tài chính, giáo dục đến giải trí và dịch vụ khách hàng.

Mức ảnh hưởng của AI đã thay đổi cách con người làm việc và sống. Nó đã cải thiện hiệu suất và độ chính xác trong nhiều tác vụ, từ xử lý dữ liệu, phân tích thông tin, tự động hóa quy trình sản xuất, đến phát hiện bệnh, chẩn đoán y tế và trợ giúp trong quyết định lâm sàng. AI cũng đã mang lại những tiện ích mới như trợ lý ảo, công nghệ nhận diện giọng nói và nhận diện hình ảnh, hệ thống tự động lái xe và nhiều ứng dụng khác.

Tuy nhiên, AI cũng đặt ra một số thách thức và vấn đề mà con người đang gặp phải. Một vấn đề quan trọng là vấn đề đạo đức và trách nhiệm trong việc sử dụng AI. Việc dựa vào quyết định của hệ thống AI có thể gây ra hậu quả không mong muốn hoặc gây ra sự phân biệt đối xử. Việc thiếu sự minh bạch và khả năng giải thích của hệ thống AI cũng gây ra lo ngại và khó khăn trong việc chấp nhận và sử dụng AI.

Nhu cầu giải đáp thắc mắc nhanh chóng cũng là một vấn đề quan trọng mà AI có thể giúp đỡ. Với khả năng xử lý dữ liệu nhanh chóng và tự động, AI có thể cung cấp thông tin và giải đáp câu hỏi một cách hiệu quả. Trong lĩnh vực dịch vụ khách hàng, chatbot và hệ thống trò chuyện thông minh có thể cung cấp hỗ trợ 24/7 và giải đáp các câu hỏi thường gặp một cách tự động.

Ngoài ra, AI cũng có thể giúp giải quyết một số vấn đề đang đối diện với con người. Ví dụ, trong lĩnh vực y tế, AI có thể hỗ trợ trong phát hiện sớm bệnh, tư vấn điều trị và phân tích dữ liệu lâm sàng. Trong lĩnh vực nông nghiệp, AI có thể giúp cải thiện quản lý và dự đoán sản lượng nông sản. Trong lĩnh vực môi trường, AI có thể được sử dụng để theo dõi và dự đoán biến đổi khí hậu.

Tuy nhiên, khi sử dụng AI để giải quyết các vấn đề, cần chú trọng đến việc đảm bảo rằng công nghệ này được áp dụng một cách công bằng, an toàn và minh bạch. Cần có sự đồng thuận và quản lý chặt chẽ để đảm bảo rằng AI không gây ra tác động tiêu cực hoặc gia tăng khoảng cách xã hội.

1.2. Định hướng giải pháp

1. Các phương hướng giải pháp

Để giải quyết các vấn đề liên quan đến sự phổ biến của AI, mức ảnh hưởng, tác động tới con người và nhu cầu giải đáp thắc mắc nhanh chóng, dưới đây là một số giải pháp cụ thể có thể được áp dụng:

- a. Nghiên cứu và phát triển đạo đức và trách nhiệm của AI: Cần đưa ra các chuẩn mực đạo đức và quy tắc pháp lý rõ ràng để đảm bảo rằng các hệ thống AI hoạt động một cách công bằng, không gây phân biệt đối xử và có khả năng giải thích quyết định của chúng. Cần tạo ra môi trường khuyến khích công nghệ AI có trách nhiệm và đảm bảo tuân thủ các quy tắc đạo đức và quy phạm pháp lý.
- b. Tăng cường minh bạch và giải thích AI: Cần nghiên cứu và phát triển các phương pháp để giải thích quyết định của hệ thống AI một cách rõ ràng và dễ hiểu cho người sử dụng. Việc này sẽ giúp tạo ra niềm tin và sự chấp nhận đối với công nghệ AI, đồng thời cho phép con người kiểm tra và đánh giá tính công bằng và đúng đắn của các quyết định AI.
- c. Đào tạo và giáo dục về AI: Cần đầu tư vào việc đào tạo và giáo dục người dùng, nhân viên và quản lý về công nghệ AI. Điều này giúp tăng cường nhận thức về tiềm năng và giới hạn của AI, đồng thời trang bị người dùng với kiến thức và kỹ năng cần thiết để tương tác và sử dụng AI một cách hiệu quả và đúng cách.
- d. Phát triển hệ thống AI công bằng: Cần đảm bảo rằng việc phát triển và triển khai AI không gây tăng khoảng cách xã hội. Cần kiểm soát và giám sát quy trình huấn luyện dữ liệu và xử lý dữ liệu để tránh thiên vị và phân biệt đối xử. Cần xem xét việc sử dụng các kỹ thuật và thuật toán khai thác dữ liệu để giảm bớt hiện tượng rào cản và đảm bảo rằng AI phục vụ mọi người một cách công bằng và đa dạng.

e. Nâng cao khả năng giải đáp nhanh chóng: Sử dụng công nghệ AI để xây dựng hệ thống giải đáp thắc mắc tự động và chatbot thông minh. Các hệ thống này có thể cung cấp câu trả lời tức thì và hỗ trợ người dùng trong việc tìm kiếm thông tin và giải quyết các vấn đề cơ bản.

2. Giải pháp bằng chatbot áp dụng công nghệ chatgpt

Hướng tới việc áp dụng công nghệ GPT vào các ứng dụng di động (mobile), có thể đem lại nhiều lợi ích đáng kể. Dưới đây là một số lợi ích chính của việc phát triển ứng dụng di động GPT:

a. Giao tiếp và trợ giúp người dùng: Ứng dụng di động GPT có thể được sử dụng làm trợ lý ảo hoặc chatbot thông minh để giao tiếp và giải đáp câu hỏi của người dùng. Người dùng có thể dễ dàng tương tác với ứng dụng thông qua chat hoặc giọng nói, nhận được các thông tin, hướng dẫn và giải đáp nhanh chóng cho các thắc mắc của mình.

b. Dịch thuật và thông tin đa ngôn ngữ: Ứng dụng di động GPT có thể cung cấp khả năng dịch thuật tự động và thông tin đa ngôn ngữ cho người dùng. Điều này rất hữu ích cho du khách, người học ngoại ngữ hoặc người làm việc trong môi trường đa ngôn ngữ.

c. Hỗ trợ học tập và giáo dục: Ứng dụng di động GPT có thể cung cấp tài liệu học tập, bài giảng và câu trả lời cho các câu hỏi liên quan đến học tập và giáo dục. Người dùng có thể tìm kiếm thông tin, tìm hiểu kiến thức mới và nhận được sự hỗ trợ trong việc giải quyết các bài tập và vấn đề học tập.

d. Tích hợp vào ứng dụng ngành công nghiệp: Ứng dụng di động GPT có thể được tích hợp vào các ngành công nghiệp như y tế, tài chính, du lịch, bất động sản và nhiều lĩnh vực khác. Chẳng hạn, trong lĩnh vực y tế, ứng dụng có thể cung cấp thông tin về triệu chứng bệnh, tư vấn về sức khỏe và hỗ trợ trong chẩn đoán sơ bộ.

e. Giao tiếp với người dùng một cách tự nhiên: GPT có khả năng tạo ra văn bản tự nhiên và gần giống con người. Điều này cho phép ứng dụng di động GPT tương tác với người dùng một cách tự nhiên, giúp cải thiện trải nghiệm người dùng và tạo ra một môi trường giao tiếp thuận tiện và hấp dẫn.

CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG

2.1. Dart

Dart là một ngôn ngữ lập trình phát triển bởi Google, được sử dụng để xây dựng ứng dụng di động, ứng dụng web và các hệ thống phía máy chủ. Nó được thiết kế để cung cấp một cú pháp đơn giản, hiệu suất cao và khả năng mở rộng cho các lập trình viên.

Một số đặc điểm của Dart:

- a. Cú pháp đơn giản: Dart có cú pháp rõ ràng và dễ đọc, giúp cho việc viết mã trở nên dễ dàng hơn cho cả những người mới học lập trình.
- b. Kiểu tĩnh và động: Dart hỗ trợ kiểu tĩnh và động. Bạn có thể khai báo kiểu tĩnh cho các biến nếu muốn, tuy nhiên nó cũng cho phép kiểu động nếu bạn không muốn khai báo kiểu.
- c. Hỗ trợ hướng đối tượng: Dart là một ngôn ngữ hướng đối tượng, có hỗ trợ đầy đủ các tính năng như kế thừa, đa hình, giao diện, và trừu tượng hóa.
- d. Bộ nhớ tự động thu gom rác: Dart sử dụng bộ nhớ tự động thu gom rác để quản lý việc cấp phát và thu hồi bộ nhớ, giúp đơn giản hóa quá trình lập trình và giảm rủi ro lỗi nhớ.
- e. Đa nền tảng: Mã nguồn Dart có thể chạy trên nhiều nền tảng, bao gồm web thông qua trình duyệt, ứng dụng di động thông qua Flutter, và ứng dụng máy tính để bàn thông qua Dart VM.
- f. Flutter framework: Dart là ngôn ngữ chính được sử dụng để phát triển ứng dụng di động và web bằng Flutter framework. Flutter là một framework giao diện người dùng gọn nhẹ và mạnh mẽ, cho phép bạn xây dựng các ứng dụng đẹp và tương tác trên nhiều nền tảng.

Dart là một ngôn ngữ linh hoạt và mạnh mẽ cho việc phát triển ứng dụng đa nền tảng. Với sự hỗ trợ từ Flutter framework, Dart đã trở thành một lựa chọn phổ biến cho việc xây dựng các ứng dụng di động...

2.2. Flutter

Flutter là một framework phát triển ứng dụng di động đa nền tảng được phát triển bởi Google. Nó cho phép bạn xây dựng các ứng dụng di động đẹp, nhanh chóng và tương thích trên cả iOS và Android từ một mã nguồn duy nhất.

Dưới đây là một số đặc điểm chính của Flutter:

- a. Tính đa nền tảng: Flutter cho phép bạn phát triển ứng dụng cho nhiều nền tảng khác nhau từ một mã nguồn duy nhất. Bạn có thể xây dựng ứng dụng di động cho iOS và Android, ứng dụng web và cả ứng dụng máy tính để bàn trên Windows, macOS và Linux.
- b. Giao diện người dùng linh hoạt: Flutter cung cấp một thư viện giao diện người dùng (UI) đáng tin cậy và mạnh mẽ. Với Flutter, bạn có thể xây dựng giao diện người dùng đẹp, tương tác và mượt mà với các thành phần được tạo sẵn như nút bấm, danh sách cuộn, hộp văn bản và hơn thế nữa.
- c. Tốc độ cao: Với Flutter, ứng dụng của bạn được biên dịch thành mã máy trực tiếp, điều này giúp cải thiện hiệu năng và tăng tốc độ khởi chạy. Flutter sử dụng động lực GPU để vẽ giao diện người dùng, giúp ứng dụng chạy mượt mà và đáp ứng nhanh chóng.
- d. Hot Reload: Đặc điểm nổi bật của Flutter là khả năng Hot Reload, cho phép bạn chỉnh sửa mã nguồn và nhìn thấy kết quả ngay lập tức trên thiết bị hoặc máy ảo. Điều này giúp tăng tốc quá trình phát triển và cho phép bạn thử nghiệm và tinh chỉnh giao diện người dùng một cách nhanh chóng.
- e. Cộng đồng mạnh mẽ: Flutter có một cộng đồng lớn và nhiệt tình. Bạn có thể tìm thấy nhiều tài liệu, ví dụ và thư viện hữu ích từ cộng đồng này. Flutter được hỗ trợ và phát triển liên tục bởi Google và các nhà phát triển trên toàn thế giới.

Flutter đã trở thành một công cụ phổ biến cho việc phát triển ứng dụng đa nền tảng. Với sự linh hoạt, tốc độ và khả năng tái sử dụng mã nguồn, Flutter giúp bạn xây dựng các ứng dụng di động... đẹp mắt và chất lượng.

2.3. Packages

1. Trong lĩnh vực phát triển ứng dụng di động, khái niệm "Packages" thường được sử dụng để chỉ các bộ thư viện, module hoặc gói phần mềm có sẵn để giúp lập trình viên xây dựng

ứng dụng di động nhanh chóng và hiệu quả. Packages cung cấp các chức năng và tính năng đã được triển khai trước đó, giúp giảm thời gian và công sức trong việc viết mã từ đầu.

Trong lĩnh vực phát triển ứng dụng di động, có hai Packages phổ biến nhất là:

a. Các Packages liên quan đến ứng dụng di động: Đây là các Packages cung cấp các thư viện và công cụ để phát triển ứng dụng di động trên các nền tảng cụ thể như iOS và Android. Ví dụ: Flutter ..

b. Các Packages liên quan đến chức năng và tính năng cụ thể: Đây là các Packages cung cấp các chức năng và tính năng cụ thể cho ứng dụng di động. Ví dụ: Firebase ...

2. Các Packages này cung cấp các khả năng, chức năng và tính năng sẵn có cho lập trình viên di động, giúp họ xây dựng ứng dụng một cách nhanh chóng, tiết kiệm thời gian và tối ưu hóa quy trình phát triển. Dưới đây là danh sách các package mà nhóm đã sử dụng:

a. Package “provider” được sử dụng để quản lý trạng thái và cung cấp dữ liệu cho các widget trong ứng dụng của bạn.

b. Package “http” được sử dụng để thực hiện các yêu cầu HTTP, như truy vấn dữ liệu từ máy chủ hoặc gửi dữ liệu lên máy chủ.

c. Package “flutter_spinkit” cung cấp các chủ đề quay vòng (spinners) đẹp mắt để sử dụng trong ứng dụng của bạn.

d. Package “animated_text_kit” cung cấp các hiệu ứng văn bản động và chuyển đổi cho các widget văn bản trong ứng dụng của bạn.

e. Package “cupertino” cung cấp các widget và chủ đề dựa trên thiết kế của hệ điều hành iOS, giúp bạn tạo ra giao diện người dùng giống iOS trên ứng dụng Flutter của mình.

Các package này đã giúp nhóm em xây dựng giao diện người dùng, quản lý trạng thái, thực hiện yêu cầu HTTP và tạo các hiệu ứng động trong ứng dụng của mình.

2.5. ChatGPT API

1. Tổng quan về API

API (Application Programming Interface) là một tập hợp các quy tắc và giao thức cho phép các phần mềm và ứng dụng khác tương tác và trao đổi dữ liệu với nhau. API cho phép các ứng dụng khác nhau hoạt động cùng nhau và sử dụng các chức năng của nhau mà không cần biết chi tiết cài đặt bên trong.

API cung cấp một giao diện cho các nhà phát triển để truy cập vào các tính năng, dịch vụ và dữ liệu của một ứng dụng, hệ thống hoặc nền tảng khác. Thông qua API, các ứng dụng có thể gửi yêu cầu (requests) và nhận phản hồi (responses) từ nhau.

API có thể có nhiều dạng khác nhau, bao gồm API web (web APIs) và API phần mềm (software APIs). API web thường được sử dụng để truy cập và tương tác với các dịch vụ và tài nguyên trên Internet thông qua giao thức HTTP. API phần mềm thường được sử dụng trong việc tích hợp các ứng dụng và hệ thống phần mềm khác nhau, cho phép chúng tương tác và trao đổi dữ liệu.

API định nghĩa các quy tắc và cú pháp mà các ứng dụng phải tuân theo để truy cập vào các chức năng và dữ liệu cung cấp bởi nhà cung cấp API. API cung cấp một cách tiêu chuẩn và chuẩn mực cho việc tương tác giữa các phần mềm và ứng dụng khác nhau, giúp tăng khả năng mở rộng và tái sử dụng của hệ thống phần mềm.

2. Tổng quan về ChatGPT

ChatGPT là một mô hình học sâu (deep learning) được phát triển bởi OpenAI. Nó là một biến thể của mô hình ngôn ngữ tự nhiên GPT (Generative Pre-trained Transformer).

GPT (Generative Pre-trained Transformer) là một mô hình học sâu dựa trên kiến trúc Transformer, được sử dụng để xử lý ngôn ngữ tự nhiên. Mô hình GPT được đào tạo trên một lượng lớn văn bản từ Internet để hiểu và tổng hợp kiến thức ngôn ngữ.

Mô hình ChatGPT, được xây dựng dựa trên kiến trúc GPT, được tăng cường để tạo ra một hệ thống tương tác ngôn ngữ tự nhiên. Nó được huấn luyện để có khả năng đọc và hiểu câu hỏi từ người dùng, và sau đó tạo ra câu trả lời phù hợp và có logic.

ChatGPT có khả năng tương tác thông minh với người dùng, giúp họ có thể gửi câu hỏi hoặc yêu cầu thông tin và nhận được câu trả lời tự động từ mô hình. Nó có thể được sử dụng trong

các ứng dụng chatbot, hệ thống trò chuyện tự động, hoặc các hệ thống tương tác ngôn ngữ tự nhiên khác.

Mô hình ChatGPT đã được phát triển và cải tiến liên tục để đạt được sự linh hoạt và tự nhiên trong tương tác ngôn ngữ, mang lại trải nghiệm trò chuyện gần như với con người.

3. ChatGPT API là gì

ChatGPT API là một API được cung cấp bởi OpenAI cho phép các nhà phát triển tương tác với mô hình ChatGPT thông qua mạng Internet. API này cho phép bạn gửi yêu cầu tới mô hình ChatGPT và nhận câu trả lời tự động từ nó.

Sử dụng ChatGPT API, bạn có thể tích hợp mô hình ChatGPT vào ứng dụng, trang web hoặc hệ thống của mình. Bằng cách gửi yêu cầu API với câu hỏi hoặc nội dung cần xử lý, bạn có thể nhận được câu trả lời tự động từ mô hình ChatGPT thông qua phản hồi API.

API này cung cấp khả năng tương tác thông qua giao thức HTTP và gửi các yêu cầu dạng JSON. Bạn cần gửi yêu cầu API chứa câu hỏi hoặc nội dung cần xử lý và API sẽ trả về câu trả lời từ mô hình ChatGPT dưới dạng JSON hoặc dữ liệu văn bản.

ChatGPT API giúp bạn sử dụng khả năng tương tác tự nhiên của mô hình ChatGPT mà không cần quan tâm đến việc triển khai và huấn luyện mô hình. Bạn chỉ cần sử dụng API để gửi yêu cầu và nhận câu trả lời, và OpenAI sẽ quản lý việc vận hành và duy trì mô hình ChatGPT.

Tuy nhiên cũng cần lưu ý rằng nó có thể tạo ra phản hồi không chính xác hoặc không đáng tin cậy đôi khi. Việc điều chỉnh và kiểm soát dữ liệu đầu vào và xử lý kết quả từ API là rất quan trọng để đảm bảo chất lượng và an toàn trong việc sử dụng nó

CHƯƠNG 3. XÂY DỰNG GIAO DIỆN VÀ KẾT NỐI API

3.1. Triển khai kỹ thuật

3.1.1. Công cụ sử dụng

Android Studio là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) được phát triển bởi Google, được sử dụng để phát triển ứng dụng di động chạy trên nền tảng Android. Nó cung cấp các công cụ và tài nguyên cho các nhà phát triển để tạo, kiểm tra và triển khai ứng dụng Android.

Android Studio cung cấp một giao diện đồ họa cho phép bạn tạo, chỉnh sửa và quản lý mã nguồn của ứng dụng. Nó tích hợp các công cụ phát triển Android cần thiết như trình biên dịch, trình gỡ lỗi, trình quản lý phiên bản và trình tạo giao diện người dùng.

Một trong những tính năng quan trọng của Android Studio là Emulator, cho phép bạn kiểm tra ứng dụng trên nhiều thiết bị Android ảo để đảm bảo tính tương thích và hiệu suất của ứng dụng trên các phiên bản Android khác nhau.

Android Studio cũng cung cấp các công cụ hỗ trợ phát triển ứng dụng Android như trình quản lý tài nguyên, trình tạo giao diện người dùng kéo và thả (drag-and-drop), và trình soạn thảo mã nguồn mạnh mẽ.

Tóm lại, Android Studio là một IDE chính thức và mạnh mẽ cho phát triển ứng dụng Android, được xây dựng trên nền tảng IntelliJ IDEA và cung cấp nhiều công cụ và tài nguyên hữu ích để giúp bạn tạo ra ứng dụng Android chất lượng cao.

3.1.2. Kết quả đạt được

1. Ứng dụng ChatGPT hoạt động:

a. Hộp thoại chat: Ứng dụng cung cấp một giao diện hộp thoại chat để người dùng tương tác với mô hình ChatGPT. Người dùng có thể gửi các câu hỏi, truy vấn hoặc tin nhắn và nhận được câu trả lời tự động từ mô hình.

b. Tích hợp mô hình ChatGPT: Ứng dụng đã tích hợp mô hình ChatGPT để xử lý các truy vấn và trả lời tự động. Khi người dùng gửi tin nhắn, chúng tôi gửi yêu cầu đến mô hình ChatGPT và hiển thị câu trả lời trên giao diện.

c. Quản lý trạng thái: Chúng tôi theo dõi lịch sử tin nhắn và giữ cho người dùng nhìn thấy cuộc trò chuyện trước đó.

d. Đánh giá hiệu suất: Bao gồm thời gian phản hồi từ mô hình ChatGPT, tốc độ tải và hiển thị tin nhắn trên giao diện của ứng dụng trong quá trình sử dụng.

2. Tương tác thông minh và tự nhiên

a. Hiểu ngôn ngữ tự nhiên: Mô hình ChatGPT đã được đào tạo để hiểu và xử lý ngôn ngữ tự nhiên. Nó có khả năng phân tích câu hỏi và truy vấn của người dùng và tạo ra câu trả lời phù hợp và chính xác.

b. Phản hồi tự động: Mô hình ChatGPT cung cấp khả năng tạo ra phản hồi tự động và có logic. Nó có thể đưa ra câu trả lời dựa trên thông tin trong câu hỏi và kiến thức có sẵn.

c. Tính linh hoạt: Mô hình ChatGPT có khả năng đối phó với nhiều loại câu hỏi và truy vấn. Nó có thể hiểu và đáp ứng với các câu hỏi đơn giản, câu hỏi phức tạp hoặc yêu cầu thông tin cụ thể.

d. Đánh giá tương tác: Chúng tôi đánh giá tương tác của người dùng với ứng dụng và độ hài lòng từ phản hồi của mô hình ChatGPT. Sự tương tác thông minh và tự nhiên là mục tiêu chính để đảm bảo trải nghiệm tốt cho người dùng.

3. Giao diện người dùng thân thiện và trực quan

a. Thiết kế giao diện hấp dẫn: Chúng tôi đã tạo ra một giao diện gọn gàng và hấp dẫn cho ứng dụng. Sử dụng các yếu tố thiết kế hợp lý, màu sắc hài hòa và phông chữ dễ đọc để mang đến trải nghiệm thẩm mỹ tốt cho người dùng.

b. Dễ sử dụng: Giao diện người dùng được thiết kế sao cho dễ sử dụng và thân thiện với người dùng. Chúng tôi đã đặt các phần tử giao diện và nút điều hướng một cách logic và trực quan để người dùng có thể tương tác dễ dàng.

c. Tương thích đa nền tảng: Ứng dụng được phát triển trên nền tảng Flutter, mang lại khả năng tương thích đa nền tảng. Điều này cho phép người dùng trải nghiệm ứng dụng trên nhiều thiết bị khác nhau như android hay ios.

3.1.3. Kiểm thử

1. Kiểm tra API keys

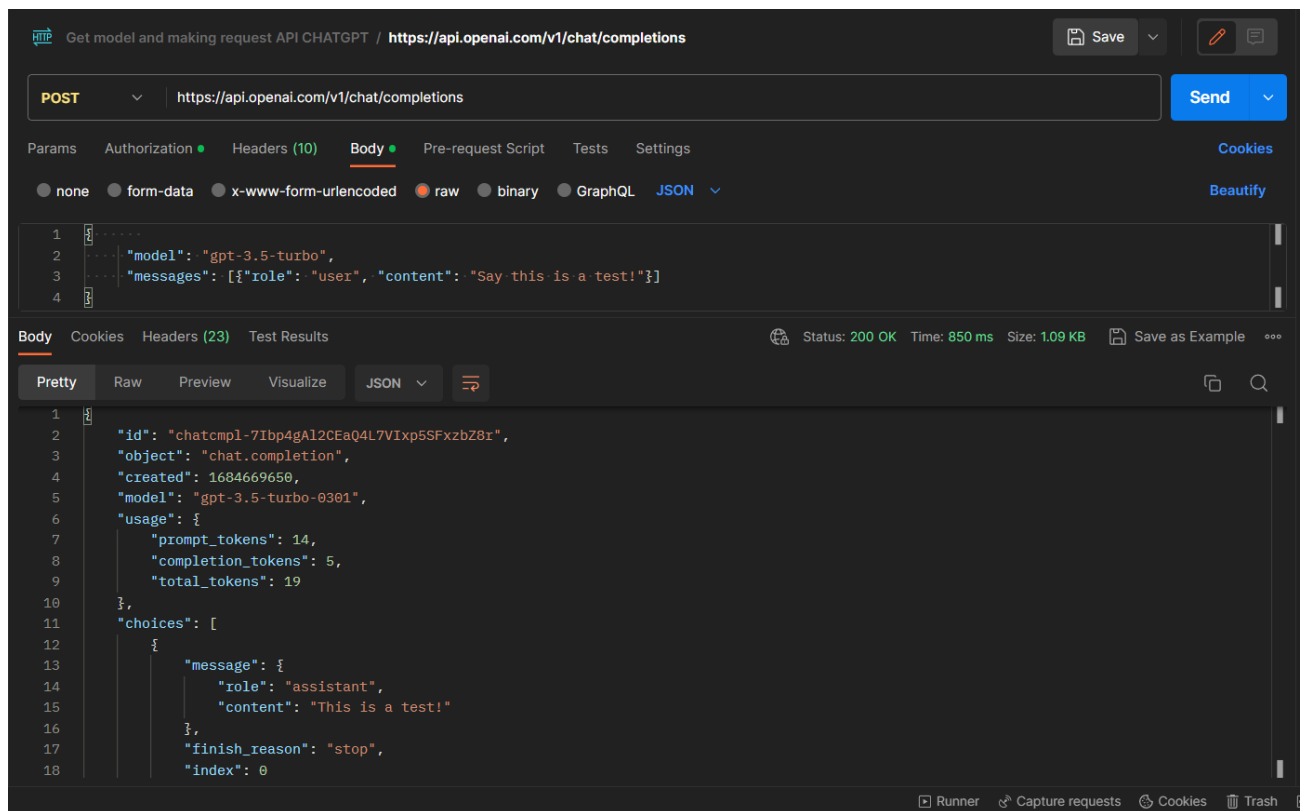
API keys (Application Programming Interface keys) là chuỗi ký tự đặc biệt được sử dụng để xác thực và quản lý quyền truy cập vào một API (Application Programming Interface). API keys được sử dụng để định danh và xác thực người dùng hoặc ứng dụng khi truy cập vào một API cụ thể. Khi một yêu cầu truy cập API được gửi đi, API key sẽ được gắn kèm vào yêu cầu đó để xác nhận danh tính và quyền hạn của người dùng hoặc ứng dụng. API keys giúp các nhà phát triển và nhà cung cấp dịch vụ API có thể kiểm soát quyền truy cập vào dữ liệu và chức năng của họ. Bằng cách sử dụng API keys, họ có thể hạn chế quyền truy cập cho từng người dùng hoặc ứng dụng, giới hạn số lượng yêu cầu mỗi ngày, hoặc theo dõi việc sử dụng API. API keys thường được tạo và quản lý bởi nhà cung cấp API. Khi sử dụng một API, người dùng hoặc nhà phát triển sẽ được cung cấp một API key riêng để sử dụng khi gửi các yêu cầu truy cập API.

Để kiểm tra xem một API key có hoạt động hay không, bạn cần thực hiện một số bước sau:

- a. Xác định API Endpoint: Tìm hiểu API mà bạn muốn kiểm tra và xác định endpoint (đường dẫn) của API đó. Đây là địa chỉ mà bạn gửi yêu cầu API đến.
- b. Sử dụng công cụ kiểm tra API: Sử dụng công cụ như Postman, cURL hoặc trình duyệt web để gửi yêu cầu tới endpoint API.
- c. Gắn kèm API Key: Gắn API key vào yêu cầu API. Phương thức và cách gắn kèm API key có thể khác nhau tùy thuộc vào API cụ thể mà bạn đang sử dụng. Thông thường, API key được gắn kèm trong phần Header (ví dụ: Authorization header) hoặc query parameter của yêu cầu.
- d. Gửi yêu cầu: Gửi yêu cầu API đến endpoint đã xác định bằng công cụ kiểm tra API. Công cụ sẽ gửi yêu cầu với API key được gắn kèm.

e. Xem kết quả: Kiểm tra kết quả trả về từ API. Nếu API key hợp lệ và có quyền truy cập, bạn sẽ nhận được kết quả mong đợi từ API. Ngược lại, nếu API key không hợp lệ hoặc không có quyền truy cập, bạn sẽ nhận được thông báo lỗi hoặc kết quả không mong đợi.

Lưu ý rằng cách kiểm tra API key có thể khác nhau tùy thuộc vào cách API được cấu hình. Một số API có các bước xác thực phức tạp hơn, chẳng hạn như yêu cầu OAuth token. Trong trường hợp đó, bạn cần tuân theo hướng dẫn cụ thể từ nhà cung cấp API để kiểm tra API key. Dưới đây là hình ảnh kiểm thử việc tạo một yêu cầu cho openai bằng postman

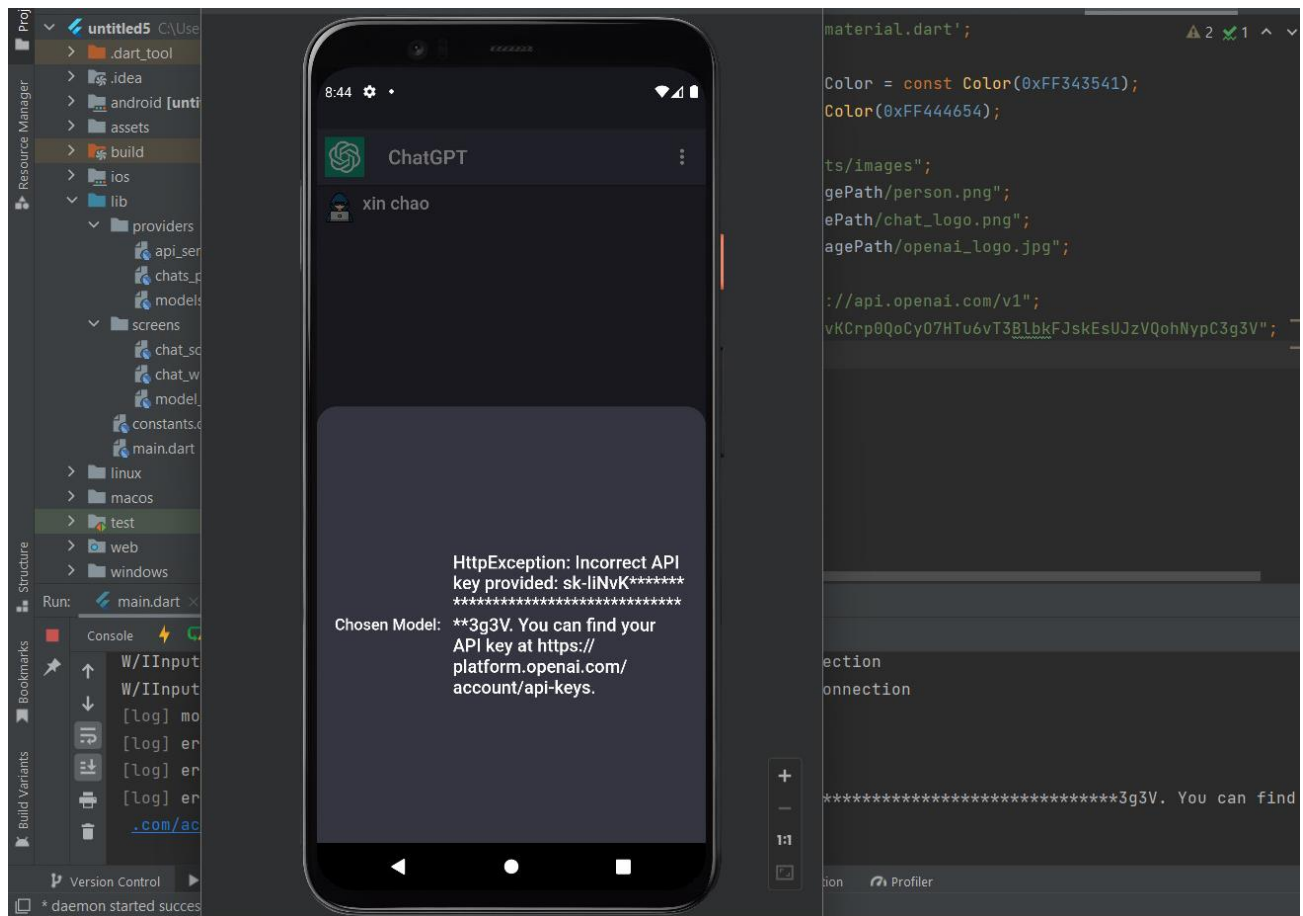


Hình 1: Kiểm tra API keys

2. Kiểm tra ứng dụng đã kết nối tới api chatgpt chưa

Để kiểm tra xem ứng dụng Flutter ChatGPT có kết nối được với API hay không, bạn có thể thực hiện các bước sau

- Kiểm tra key api trong source code viết đúng hay chưa, còn sử dụng được không.
- Kiểm tra các hàm trong tạo yêu cầu api trong source code, viết thêm một số ràng buộc hoặc thông báo lỗi để biết được tình trạng ứng dụng.



Hình 2: Giao diện ChatGPT sai key API

3.2. Xây dựng giao diện và kết nối API

1. Thiết kế tổng quan bằng widget tree

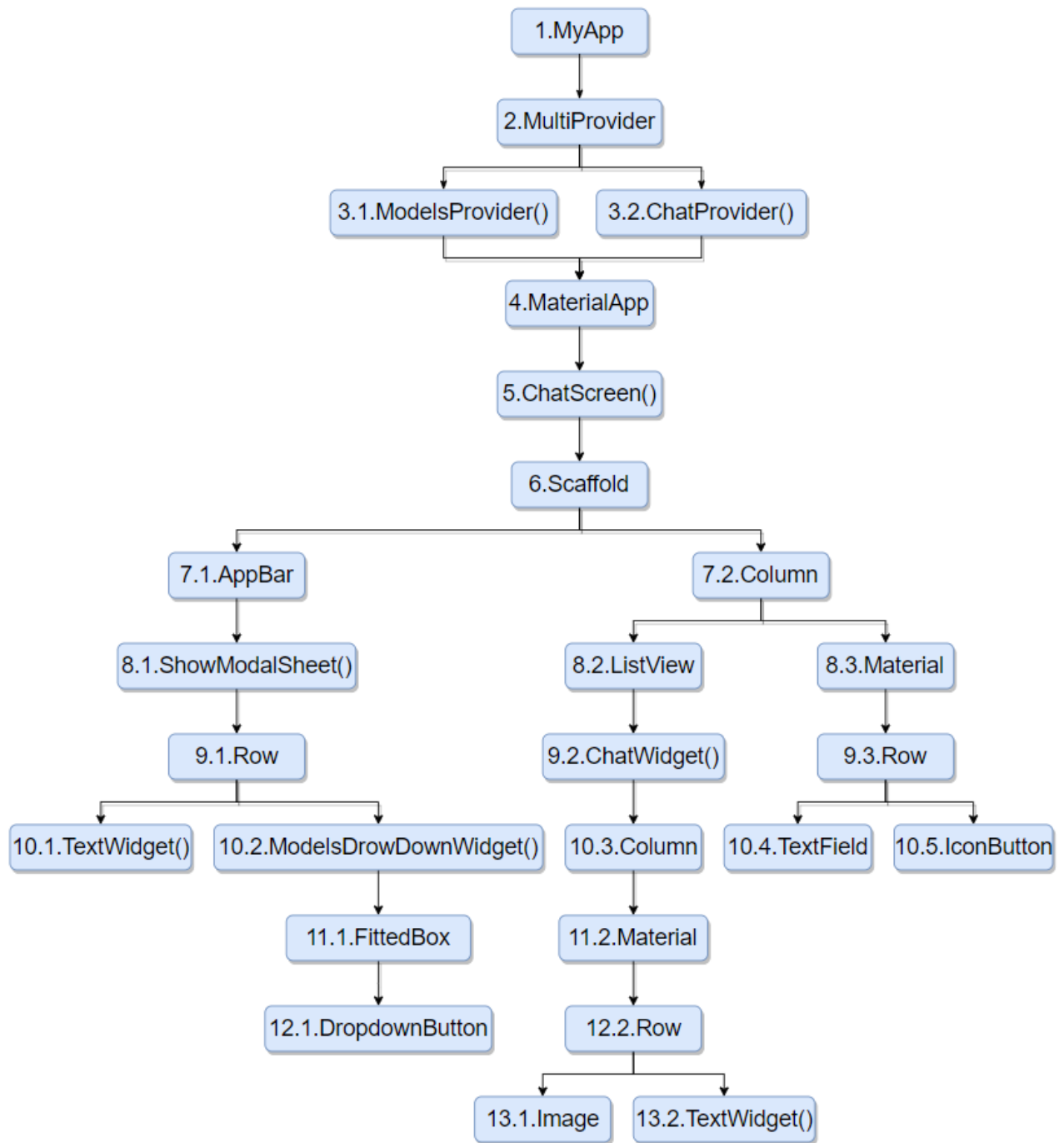
Widget tree (cây widget) là cách thể hiện cấu trúc và quan hệ giữa các widget trong một ứng dụng Flutter. Trong Flutter, các widget được xây dựng và sắp xếp theo cấu trúc cây, giống như cây genealogical tree (cây gia phả) với widget gốc ở đỉnh và các widget con phụ thuộc vào widget gốc và các widget cha trước đó.

Cây widget cho phép xác định và hiển thị sự phụ thuộc giữa các widget. Mỗi widget trong cây có thể chứa một hoặc nhiều widget con và có thể là widget cha của các widget khác. Sự phụ thuộc này định nghĩa cách các widget được hiển thị và tương tác với nhau trong giao diện người dùng.

Khi một widget thay đổi trạng thái, Flutter sẽ tìm ra cây widget và cập nhật lại chỉ những phần thay đổi. Quá trình này được gọi là quá trình xây dựng lại (rebuilding) cây widget, trong đó chỉ có những widget bị ảnh hưởng bởi sự thay đổi mới được cập nhật.

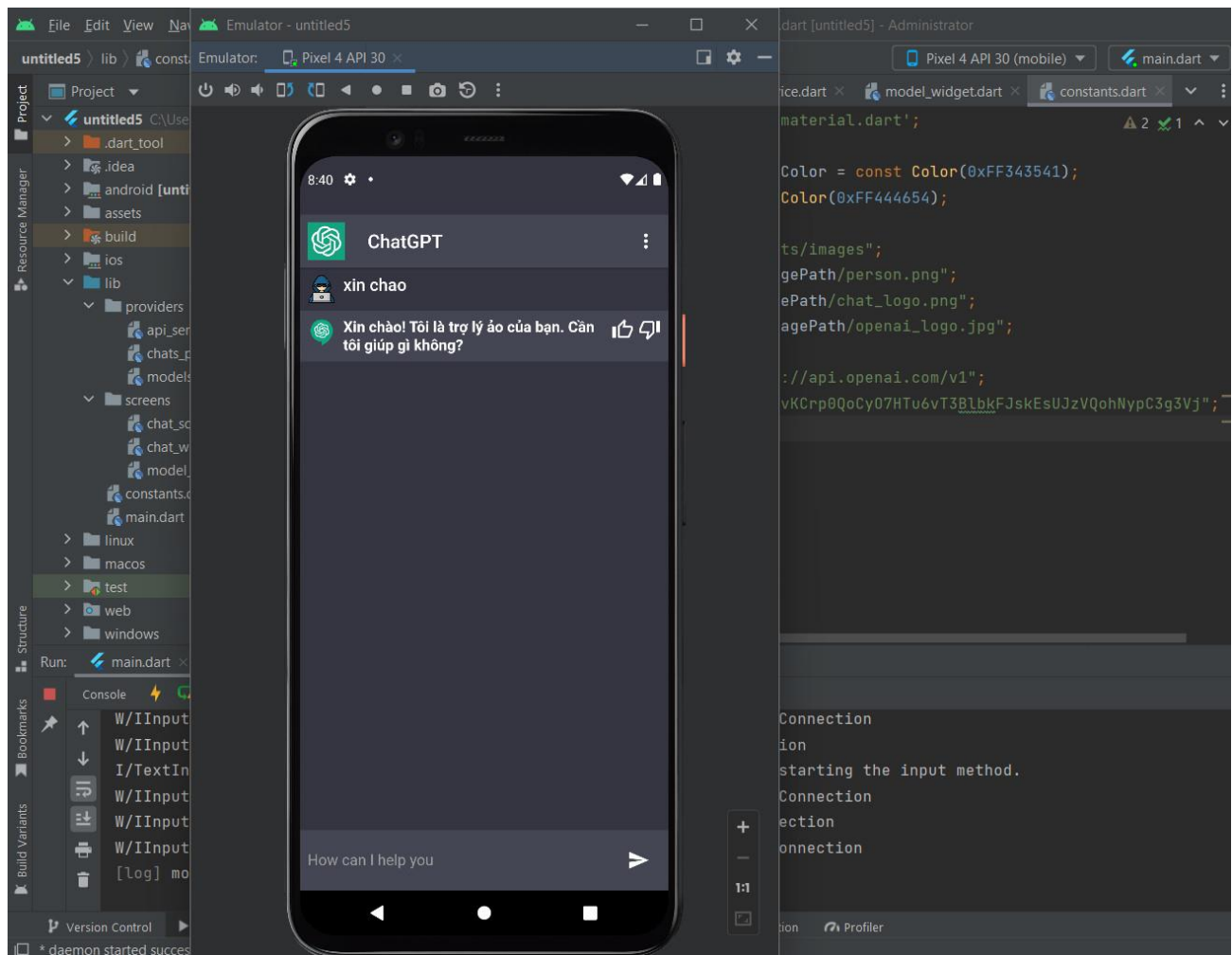
Cây widget giúp quản lý và tổ chức giao diện người dùng trong Flutter một cách hiệu quả. Nó cho phép xây dựng giao diện linh hoạt và có khả năng tái sử dụng các widget, đồng thời cho phép xử lý sự kiện và tương tác giữa các widget trong ứng dụng.

Dưới đây là mô hình widget tree sau khi hoàn thành app. Cây bao gồm nhiều widget con như `AppBar`, `Column`, `Material`, ... và nhiều hàm xử lý các tác vụ khác nhau.



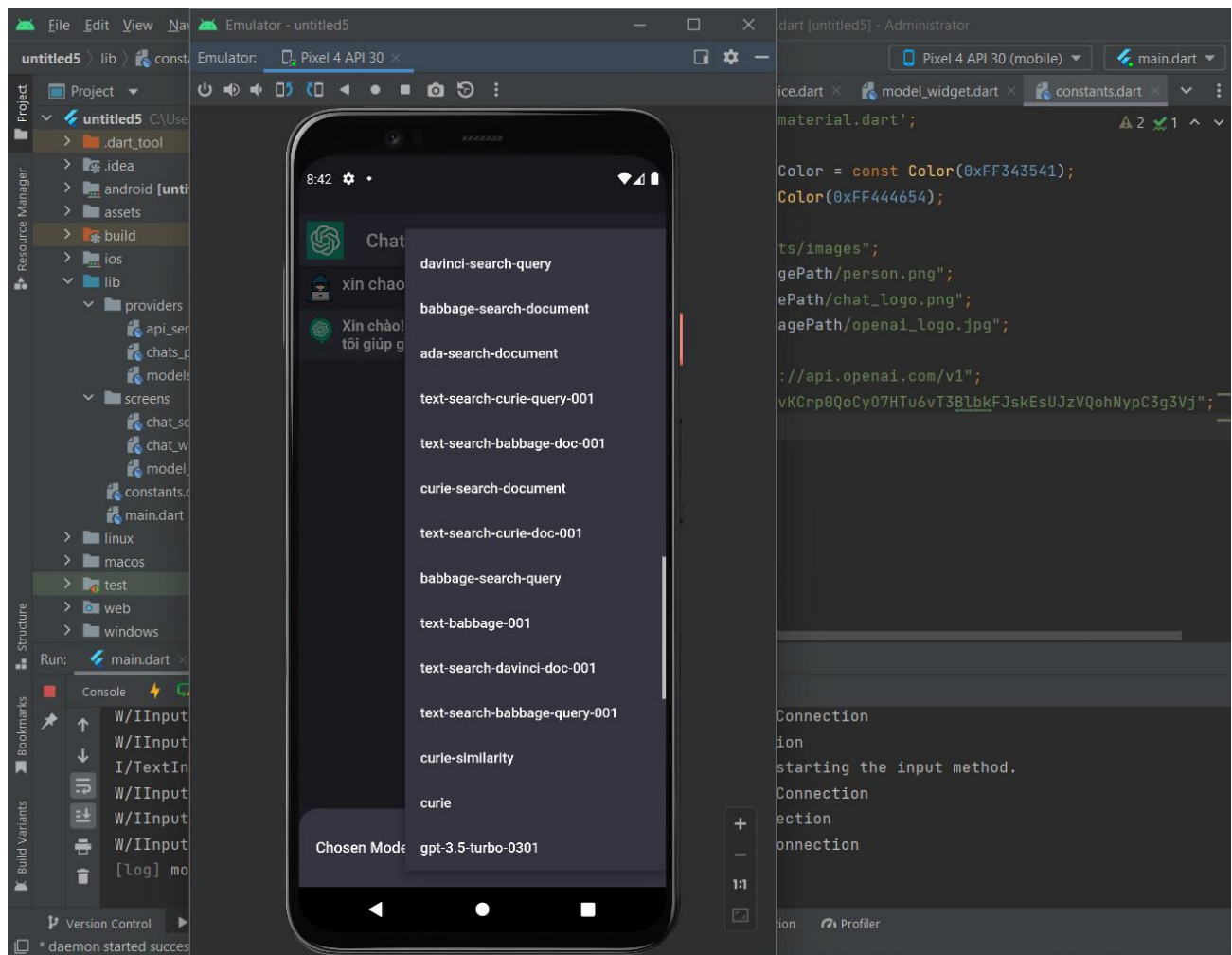
Hình 3: Thiết kế tổng quan

Dưới đây là một số hình ảnh giao diện của app.



Hình 4: Giao diện app ChatGPT

Trong ảnh 3.3.1. Giao diện app ChatGPT bao gồm một thanh AppBar có logo, tên và nút tùy chọn model. Sau khi nhập tin nhắn từ phần thanh text field và nhấn gửi tin nhắn “xin chao”, app sẽ tạo một yêu cầu tới trang chủ chatgpt thông qua keys api đã được viết trong code. Máy chủ openai sẽ trả về dòng chat “Xin chào! Tôi là trợ lý ảo của bạn. Cần tôi giúp gì không?”. Và tất cả các thao tác được thực hiện của người dùng sẽ được hiển thị trên một ListView như trong ảnh.



Hình 5: Giao diện các model

Trong ảnh 3.3.2. Giao diện và kết nối, người dùng chọn button tùy chọn để chọn các model chat gpt mà openai cung cấp, mặc định ở đây sẽ là “gpt-3-5-turbo-0301”.

2. Xây dựng giao diện và kết nối API

Để xây dựng giao diện và kết nối API cho ứng dụng Flutter mô phỏng ChatGPT, ta có những bước sau:

a. Đăng ký và nhận API key: Truy cập vào trang web của nhà cung cấp API ChatGPT (ví dụ: OpenAI) và đăng ký tài khoản. Sau đó, theo các hướng dẫn để nhận API key của bạn. API key này sẽ được sử dụng để xác thực và gửi yêu cầu tới API ChatGPT.

b. Thêm package HTTP vào ứng dụng: Sử dụng package http để gửi yêu cầu HTTP tới API ChatGPT. Thêm http package vào file pubspec.yaml của ứng dụng Flutter và chạy lệnh flutter pub get để cài đặt package.

c. Thiết kế giao diện người dùng: Xác định các thành phần giao diện cần thiết cho ứng dụng chat, ví dụ: vùng hiển thị tin nhắn, trường nhập liệu, nút gửi, và danh sách tin nhắn. Sử dụng các widget như Container, Column, Row, ListView, TextField, Button, và Text để xây dựng giao diện người dùng theo ý muốn.

d. Hiệu ứng: Sử dụng package flutter_spinkit để thêm một spinner ứng dụng đang tải dữ liệu từ máy chủ hoặc đang thực hiện một tác vụ lâu dài. Các hiệu ứng spinners có thể được tùy chỉnh với các thuộc tính như màu sắc, kích thước và tốc độ quay. Còn với animated_text_kit, bạn có thể thêm các hiệu ứng văn bản như đánh chữ, hiệu ứng chớp tắt, chuyển đổi màu sắc và nhiều hiệu ứng khác vào các phần tử văn bản trong ứng dụng của bạn. Điều này có thể làm cho giao diện của bạn trở nên sống động và thú vị hơn.

e. Tạo lớp ChatProvider: Đây là một lớp Provider (thông qua ChangeNotifierProvider) để quản lý trạng thái của ứng dụng chat. Lớp này sẽ lưu trữ danh sách tin nhắn và các phương thức để thêm tin nhắn, gửi yêu cầu API và nhận câu trả lời.

f. Kết nối với API ChatGPT: Sử dụng các thư viện HTTP để gửi yêu cầu đến API ChatGPT và nhận câu trả lời. Bạn có thể sử dụng http package hoặc các package HTTP khác để thực hiện việc này. Đảm bảo bạn đã có API key để sử dụng API ChatGPT.

g. Xử lý gửi tin nhắn: Khi người dùng gửi tin nhắn, bạn cần lấy nội dung tin nhắn từ trường nhập liệu và gửi yêu cầu API ChatGPT thông qua lớp ChatProvider. Sau đó, bạn cần xử lý câu trả lời từ API và thêm tin nhắn vào danh sách tin nhắn trong ChatProvider.

h. Hiển thị tin nhắn: Sử dụng danh sách tin nhắn trong ChatProvider để hiển thị các tin nhắn trong giao diện. Bạn có thể sử dụng ListView.builder để hiển thị danh sách tin nhắn và cập nhật giao diện khi có tin nhắn mới được thêm vào.

i. Cập nhật giao diện: Khi danh sách tin nhắn trong ChatProvider thay đổi, giao diện cần được cập nhật để hiển thị tin nhắn mới. Sử dụng Consumer widget để lắng nghe và cập nhật giao diện khi có sự thay đổi trong ChatProvider.

k. Xử lý lỗi và trạng thái: Xử lý các trạng thái như lỗi kết nối, tin nhắn đang gửi, hoặc trạng thái tải dữ liệu. Hiển thị thông báo lỗi hoặc các biểu tượng trạng thái tương ứng trong giao diện người dùng.

l. Kiểm tra và xem trước: Kiểm tra ứng dụng và xem trước giao diện để đảm bảo các chức năng và hiển thị hoạt động chính xác. Hiển thị các thông báo lỗi có thể xảy ra trong quá trình sử dụng.

KẾT LUẬN

Trong quá trình nghiên cứu và phát triển ứng dụng chatbot sử dụng công nghệ ChatGPT trên nền tảng di động bằng Flutter, chúng tôi đã thu được một số ưu điểm và nhược điểm của phương pháp này. Dưới đây là tổng kết của chúng tôi về ưu điểm, nhược điểm và hướng phát triển tiếp theo cho chủ đề này.

1. Ưu điểm

- a. Tương tác tự nhiên: Ứng dụng chatbot sử dụng mô hình ChatGPT cho phép tương tác tự nhiên và linh hoạt với người dùng. Khả năng hiểu và trả lời các câu hỏi, cung cấp thông tin hữu ích và tương tác một cách tự nhiên giúp cải thiện trải nghiệm người dùng.
- b. Khả năng học tập: Mô hình ChatGPT được huấn luyện trên lượng lớn dữ liệu và có khả năng học tập liên tục. Điều này cho phép chatbot ngày càng cải thiện và nâng cao khả năng trả lời thông qua việc tiếp tục thu thập phản hồi từ người dùng.
- c. Giao diện đa nền tảng: Flutter cung cấp một framework phát triển ứng dụng di động đa nền tảng, giúp chúng tôi xây dựng giao diện người dùng hấp dẫn và tương thích trên nhiều thiết bị và hệ điều hành khác nhau.

2. Nhược điểm

- a. Hạn chế về hiệu suất: Mô hình ChatGPT có thể gặp khó khăn trong việc xử lý các tác vụ phức tạp và đòi hỏi nhiều thời gian tính toán. Điều này có thể làm giảm hiệu suất của ứng dụng và gây trễ trong việc trả lời của chatbot.
- b. Vấn đề đào tạo và dữ liệu: Huấn luyện mô hình ChatGPT đòi hỏi một lượng lớn dữ liệu và tài nguyên tính toán. Việc thu thập và chuẩn bị dữ liệu huấn luyện có thể là một quá trình phức tạp và tốn nhiều thời gian.
- c. App còn khá đơn giản, chưa có nhiều tính năng và đa dạng giao diện.

Hướng phát triển:

- a) Tối ưu hiệu suất: Để cải thiện hiệu suất của ứng dụng chatbot, nhóm sẽ tìm hiểu và áp dụng các kỹ thuật tối ưu hóa mô hình, bao gồm việc thu gọn mô hình, sử dụng các kỹ thuật tăng tốc tính toán, hoặc sử dụng mô hình nhỏ hơn phù hợp với yêu cầu ứng dụng.
- b) Mở rộng tính năng: Có thể phát triển và mở rộng tính năng của chatbot, bao gồm việc lưu trữ dữ liệu chat người dùng, tích hợp với các dịch vụ và nguồn thông tin bên ngoài để cung cấp thông tin đa dạng và hữu ích hơn như sử dụng voice chat ...
- c) Nâng cao độ tin cậy: Để đảm bảo chatbot hoạt động một cách đáng tin cậy, nhóm sẽ xem xét việc triển khai các biện pháp kiểm tra và xác thực dữ liệu đầu vào để giảm thiểu lỗi và tăng khả năng đáp ứng chính xác của chatbot.

Tổng kết lại, ứng dụng chatbot sử dụng công nghệ ChatGPT trên nền tảng di động bằng Flutter đem lại những lợi ích lớn cho việc tương tác với người dùng. Mặc dù còn một số nhược điểm cần được giải quyết, nhưng với sự tiếp tục phát triển và nghiên cứu, chúng tôi có thể đẩy mạnh khả năng tương tác và trải nghiệm người dùng của chatbot trong tương lai.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Trang chủ Flutter, <https://flutter.dev/> , last visited May 2023.
- [2] Flutter Cookbook, <https://flutter.dev/docs/cookbook/>, last visited May 2023.
- [3] Flutter Package Pub, <https://pub.dev/>, last visited May 2023.
- [4] Tài liệu tham khảo cho API của OpenAI, <https://platform.openai.com/docs/api-reference/>, last visited May 2023.
- [5] Link code: <https://github.com/NguyenDaiPhat/ChatGPT.git> .