

**BÁO CÁO THỰC TẬP DOANH NGHIỆP**

NGÀNH CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH CÔNG NGHỆ PHẦN MỀM

**ĐỊA ĐIỂM THỰC TẬP: CÔNG TY CỔ PHẦN NETNAM**

**XÂY DỰNG ỨNG DỤNG DỰ BÁO THỜI TIẾT THỜI GIAN  
THỰC VỚI WEATHER API**

Giảng viên hướng dẫn  
(Ký, ghi rõ họ tên)

Người hướng dẫn tại doanh nghiệp  
(Ký, ghi rõ họ tên)

Sinh viên  
(Ký, ghi rõ họ tên)

HÀ NỘI 05/2025



## MỤC LỤC

MỤC LỤC .....	I
DANH MỤC CÁC TỪ VIẾT TẮT .....	III
DANH MỤC CÁC HÌNH VẼ .....	IV
DANH MỤC CÁC BẢNG BIỂU .....	V
MỞ ĐẦU .....	2
1.1 Lý do chọn đề tài .....	2
1.2 Mục tiêu thực tập .....	2
1.3 Đối tượng và phạm vi thực tập .....	3
1.4 Ý nghĩa khoa học và thực tiễn của đề tài .....	3
CHƯƠNG 1 Giới thiệu về công ty và vị trí thực tập .....	5
1.1 Mở đầu chương .....	5
1.2 Giới thiệu công ty .....	5
1.2.1 Tổ chức nhân sự .....	6
1.2.2 Các dịch vụ của công ty .....	6
1.3 Vị trí thực tập .....	7
1.4 Đề tài thực tập .....	8
1.4.1 Tên đề tài .....	8
1.4.2 Giới thiệu đề tài .....	8
1.4.3 Mục tiêu đề tài .....	8
1.4.4 Công cụ hỗ trợ .....	8
1.5 Lịch làm việc .....	9
1.6 Kết chương .....	10
CHƯƠNG 2 Nội dung thực tập .....	11
2.1 Cơ sở lý thuyết .....	11
2.1.1 Tổng quan về lập trình ứng dụng di động .....	11
2.1.2 Giới thiệu về Flutter và ngôn ngữ Dart .....	12
2.1.3 Giao tiếp với API .....	14
2.1.4 Xử lý dữ liệu thời tiết .....	14
2.2 Kết quả làm được .....	15
2.2.1 Tích hợp WeatherAPI để lấy dữ liệu thời tiết thời gian thực .....	15
2.2.2 Hiện thị chi tiết thời gian hiện tại .....	16
2.2.3 Dự báo thời tiết 7 ngày .....	20
2.2.4 Tìm kiếm thành phố .....	23
2.2.5 Tính năng chuyển đổi giữa °C và °F .....	25

2.2.6 Tích hợp GPS để lấy thời tiết theo vị trí hiện tại .....	27
2.2.7 Giao diện hỗ trợ Dark Mode .....	29
2.2.8 Kiểm tra và xử lý lỗi .....	31
2.3 Kết chương .....	33
CHƯƠNG 3 Đánh giá kết quả và định hướng phát triển .....	34
3.1 Đánh giá kết quả thực tập .....	34
3.1.1 Về kiến thức .....	34
3.1.2 Về kỹ năng .....	34
3.1.3 Đánh giá mức độ áp dụng kiến thức đã học tại trường với thực tế công việc trong quá trình thực tập. ....	35
3.1.4 Tự đánh giá khả năng đáp ứng công việc thực tế .....	36
3.2 Định hướng phát triển của đề tài .....	36
3.3 Kết chương .....	36
Kết Luận .....	37
TÀI LIỆU THAM KHẢO .....	39

## **DANH MỤC CÁC TỪ VIẾT TẮT**

<b>STT</b>	<b>Từ viết tắt</b>	<b>Từ tiếng Anh</b>	<b>Từ và nghĩa tiếng Việt</b>
1	CNTT		Công nghệ thông tin
2	GPS		Định vị, vị trí
3	HTTP	Hyper Text Transfer Protocol	Giao thức truyền tải siêu văn bản
4	SEO	Search Engine Optimization	Tối ưu hóa cho công cụ tìm kiếm
5			
6			
7			
8			
9			
10			

## **DANH MỤC CÁC HÌNH VẼ**

Hình 1-1 Công ty cổ phần NETNAM.....	5
Hình 1-2 Các dịch vụ của công ty.....	7
Hình 2-1 Logo của Dart .....	13
Hình 2-2 Logo của Flutter.....	13
Hình 2-3 Giao diện chính của ứng dụng .....	17
Hình 2-4 Giao diện nhiệt độ ngày thông qua chi tiết bình minh.....	18
Hình 2-5 Giao diện nhiệt độ ngày thông qua chi tiết hoàng hôn. ....	18
Hình 2-6 Giao diện chi tiết độ ẩm trong ngày theo giờ. ....	19
Hình 2-7 Giao diện chi tiết tốc độ gió trong ngày. ....	20
Hình 2-8 Giao diện màn hình thời tiết 7 ngày tiếp theo.....	21
Hình 2-9 Giao diện màn hình thời tiết chi tiết của ngày được xem. ....	22
Hình 2-10 Giao diện tìm kiếm thành phố.....	24
Hình 2-11 Kết quả tìm kiếm .....	25
Hình 2-12 Nhiệt độ ở dạng °C .....	26
Hình 2-13 Giao diện nhiệt độ ở dạng °F .....	27
Hình 2-14 Giao diện khi chưa bật GPS.....	28
Hình 2-15 Giao diện thông báo GPS chưa bật.....	29
Hình 2-16 Giao diện khi bật GPS và đã cập nhật vị trí.....	29
Hình 2-17 Giao diện Light Mode.....	31
Hình 2-18 Giao diện Dark Mode .....	31
Hình 2-19 Lỗi không có kết nối mạng .....	32

## **DANH MỤC CÁC BẢNG BIỂU**

Bảng 1-1 Công cụ, công nghệ hỗ trợ .....	8
Bảng 1-2 Bảng lịch làm việc theo từng tuần .....	10

## MỞ ĐẦU

### 1.1 Lý do chọn đề tài

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, lĩnh vực Công nghệ Thông tin ngày càng đóng vai trò quan trọng trong đời sống hàng ngày, đặc biệt là trong việc cung cấp các thông tin thiết yếu như dự báo thời tiết. Việc phát triển một ứng dụng dự báo thời tiết theo thời gian thực không chỉ mang lại giá trị thiết thực cho người dùng, giúp họ nắm bắt điều kiện thời tiết một cách nhanh chóng và chính xác, mà còn là cơ hội để sinh viên ngành Công nghệ Thông tin được tiếp cận, thực hành và vận dụng những kiến thức đã học vào môi trường ứng dụng thực tế, đồng thời tìm hiểu và làm chủ các công nghệ hiện đại.

- Tính cấp thiết của đề tài
  - Thông tin thời tiết có vai trò quan trọng trong nhiều lĩnh vực như nông nghiệp, du lịch, giao thông, tổ chức sự kiện...
  - Nhu cầu xem thời tiết theo thời gian thực ngày càng cao, đặc biệt trên nền tảng di động.
  - Các ứng dụng thời tiết hiện nay đôi khi chưa cá nhân hóa trải nghiệm người dùng, thiếu tương tác hoặc không hỗ trợ tiếng Việt tốt.
- Tầm quan trọng của vấn đề nghiên cứu
  - Việc xây dựng một ứng dụng dự báo thời tiết giúp sinh viên rèn luyện kỹ năng lập trình thực tế, hiểu rõ quy trình phát triển phần mềm từ phân tích yêu cầu đến triển khai, thử nghiệm.
- Lý do chọn đề tài
  - Bản thân em có định hướng phát triển trong mảng lập trình di động.
  - Đề tài giúp em vận dụng kiến thức học về Flutter, Dart,...
  - Đây là cơ hội tốt để rèn luyện kỹ năng làm việc nhóm, viết báo cáo, và trải nghiệm môi trường doanh nghiệp thực tế.

### 1.2 Mục tiêu thực tập

- Mục tiêu tổng quát
  - Giúp bản thân vận dụng kiến thức và kỹ năng đã học vào môi trường làm việc thực tế tại doanh nghiệp. Qua đó, tôi có cơ hội tiếp cận quy trình phát triển phần mềm chuyên nghiệp, rèn luyện tư duy lập trình, khả năng giải quyết vấn đề và nâng cao kỹ năng mềm như làm việc nhóm, giao tiếp, và quản lý thời gian. Đồng thời, thực tập giúp tôi hiểu rõ hơn về định hướng nghề nghiệp tương lai, tích lũy kinh nghiệm thực tế phục vụ công việc sau tốt nghiệp.
- Mục tiêu cụ thể
  - Làm quen công nghệ Flutter để phát triển ứng dụng di động đa nền tảng.



- Tìm hiểu và tích hợp WeatherAPI để hiển thị thông tin thời tiết theo thời gian thực.
- Thiết kế và xây dựng giao diện người dùng hiện đại, dễ sử dụng với các hiệu ứng chuyển cảnh, animation.
- Tích hợp tính năng định vị GPS để xác định vị trí hiện tại của người dùng và hiển thị thời tiết tương ứng.
- Phát triển chức năng tìm kiếm thành phố và dự báo thời tiết theo ngày, tuần.
- Thêm chế độ Dark Mode nhằm nâng cao trải nghiệm người dùng.
- Tối ưu hiệu suất, kiểm tra và sửa lỗi ứng dụng để đảm bảo tính ổn định khi triển khai.
- Viết báo cáo tổng hợp toàn bộ quá trình thực tập và kết quả đạt được.

### **1.3 Đối tượng và phạm vi thực tập**

- Xác định rõ đối tượng
  - Đối tượng của đợt thực tập là ứng dụng di động dự báo thời tiết thời gian thực được phát triển bằng ngôn ngữ lập trình Dart, sử dụng Flutter làm framework chính và tích hợp WeatherAPI để lấy dữ liệu thời tiết. Ứng dụng hướng tới cung cấp thông tin thời tiết một cách nhanh chóng, trực quan cho người dùng, đồng thời tích hợp thêm các tính năng như định vị GPS, tìm kiếm địa điểm và chế độ hiển thị ngày – đêm (Dark Mode).
- Giới hạn phạm vi nghiên cứu trong đợt thực tập (về thời gian, không gian, nội dung).
  - Về thời gian thực tập: Thực tập diễn ra từ ngày 10/03/2025 đến 18/05/2025, kéo dài trong vòng 10 tuần.
  - Về không gian: Thực tập được thực hiện tại công ty NETNAM, địa chỉ: Tầng 2, tòa nhà HITC, số 239 Xuân Thủy, phường Dịch Vọng Hậu, quận Cầu Giấy, thành phố Hà Nội.
  - Về nội dung
    - Tìm hiểu công nghệ Flutter và kiến trúc dự án.
    - Tích hợp API thời tiết và hiển thị dữ liệu theo thời gian thực.
    - Thiết kế giao diện, tối ưu trải nghiệm người dùng.
    - Bổ sung các tính năng nâng cao như GPS, tìm kiếm thành phố, chế độ ban đêm.
    - Kiểm thử, khắc phục lỗi và viết báo cáo thực tập tổng kết.

### **1.4 Ý nghĩa khoa học và thực tiễn của đề tài**

- Giá trị ứng dụng thực tiễn

- Ứng dụng thời tiết có tính ứng dụng cao trong đời sống hàng ngày, giúp người dùng cập nhật thông tin thời tiết một cách nhanh chóng, tiện lợi và chính xác tại mọi thời điểm, mọi vị trí.
- Đề tài mang lại trải nghiệm thực tế cho sinh viên trong môi trường doanh nghiệp, giúp phát triển kỹ năng mềm như làm việc nhóm, tự nghiên cứu, giải quyết vấn đề và thích nghi với quy trình làm việc chuyên nghiệp.
- Sản phẩm sau khi hoàn thiện có thể được sử dụng trong thực tế hoặc làm nền tảng để mở rộng thêm các tính năng khác như cảnh báo thời tiết, hiển thị bản đồ thời tiết, kết nối với lịch biểu người dùng...

## **CHƯƠNG 1 GIỚI THIỆU VỀ CÔNG TY VÀ VỊ TRÍ THỰC TẬP**

### **1.1 Mở đầu chương**

Trong khuôn khổ thực tập tốt nghiệp, em vinh dự được phân công thực tập tại Công ty Cổ phần NETNAM – một đơn vị uy tín với nhiều năm hoạt động trong lĩnh vực công nghệ mạng, hạ tầng viễn thông và phát triển các giải pháp phần mềm. Đây là môi trường làm việc chuyên nghiệp, có đội ngũ kỹ sư trình độ cao và giàu kinh nghiệm thực tiễn.

Tại đây, em được giao thực hiện đề tài "Xây dựng ứng dụng dự báo thời tiết thời gian thực sử dụng WeatherAPI", một đề tài có tính ứng dụng cao và sát với nhu cầu thực tế của người dùng hiện nay. Trong suốt quá trình thực hiện, em nhận được sự hỗ trợ và hướng dẫn tận tình từ các anh/chị kỹ sư trong công ty – những người không chỉ giỏi chuyên môn mà còn sẵn sàng chia sẻ kinh nghiệm và truyền đạt kỹ năng làm việc chuyên nghiệp.

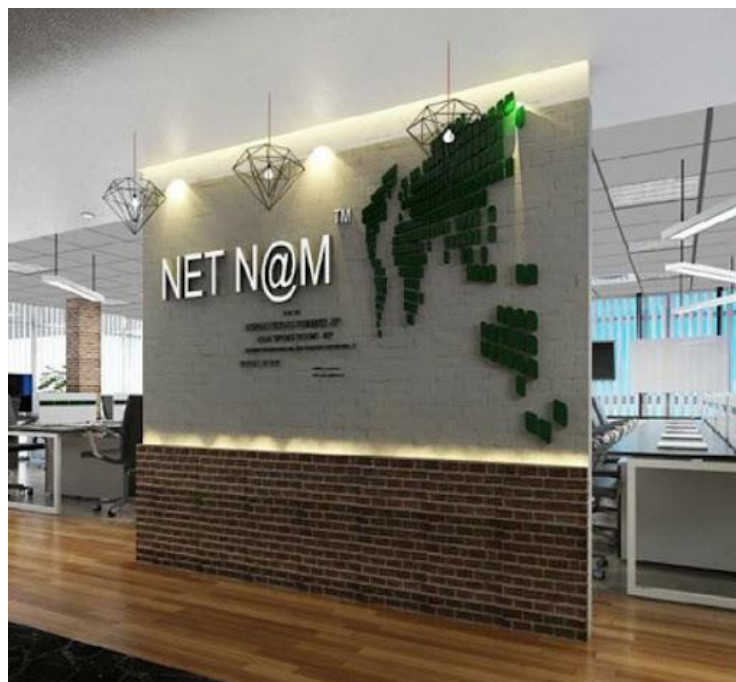
Đây thực sự là một cơ hội quý báu để em vận dụng những kiến thức đã học tại nhà trường vào môi trường làm việc thực tế, đồng thời học hỏi thêm nhiều công nghệ mới, quy trình phát triển phần mềm, kỹ năng làm việc nhóm và giải quyết vấn đề. Trải nghiệm này đã giúp em trưởng thành hơn cả về chuyên môn lẫn tư duy làm việc, tạo tiền đề vững chắc cho con đường phát triển sự nghiệp sau này.

### **1.2 Giới thiệu công ty**

Tên công ty: Công ty cổ phần NETNAM

Địa chỉ: HITC Building, Đ. Xuân Thủy, Dịch Vọng Hậu, Cầu Giấy, Hà Nội.

Website: <https://netnam.com/>



Hình 1-1 Công ty cổ phần NETNAM

### **1.2.1 Tổ chức nhân sự**

NetNam là đơn vị trực thuộc Viện Công nghệ thông tin, trong quá trình hoạt động luôn được Viện định hướng, hỗ trợ, hợp tác, chuyển giao trong nghiên cứu, phát triển công nghệ.

Bộ phận Nghiên cứu - Phát triển của NetNam được sự hỗ trợ đặc lực của cán bộ của Viện CNTT (gồm hơn 200 chuyên gia cao cấp, trong đó có 60 tiến sĩ và phó tiến sĩ, 16 giáo sư và phó giáo sư). Với chức năng kinh doanh, NetNam có khả năng sử dụng các chuyên gia cao cấp để triển khai các dự án công nghệ thông tin cỡ lớn ...

Theo NetNam [1], “Lĩnh vực kinh doanh: NetNam không chỉ cung cấp dịch vụ truy nhập Internet mà còn cung cấp giải pháp mạng chuyên nghiệp, đáp ứng nhu cầu phức tạp của nhiều tổ chức, doanh nghiệp.”

### **1.2.2 Các dịch vụ của công ty**

NETNAM là đơn vị tiên phong trong lĩnh vực cung cấp dịch vụ Internet và giải pháp công nghệ thông tin tại Việt Nam. Công ty cung cấp đa dạng các sản phẩm và dịch vụ, bao gồm:

#### **A. Dịch vụ internet tốc độ cao**

- Dịch vụ Internet chuyên biệt cho doanh nghiệp và tổ chức.
- Các giải pháp kết nối Internet cáp quang tốc độ cao, ổn định, phù hợp với nhu cầu của khách hàng lớn như khách sạn, khu nghỉ dưỡng, và sự kiện.

#### **B. Dịch vụ quản lý hệ thống CNTT**

- Giải pháp quản trị hệ thống mạng và bảo mật, giúp khách hàng duy trì hoạt động liên tục và an toàn.
- Cung cấp các công cụ giám sát và bảo trì hệ thống mạng hiệu quả.

#### **C. Dịch vụ điện toán đám mây (Cloud Services)**

- Cung cấp cơ sở hạ tầng điện toán đám mây (IaaS) với hiệu suất cao và khả năng tùy chỉnh.
- Giải pháp lưu trữ, sao lưu và phục hồi dữ liệu an toàn.

#### **D. Giải pháp hạ tầng mạng sự kiện**

- Cung cấp dịch vụ WiFi tốc độ cao cho các hội nghị, sự kiện lớn với yêu cầu khắt khe về chất lượng và độ ổn định.

#### **E. Dịch vụ bảo mật**

- Tư vấn và triển khai các giải pháp bảo mật như Firewall, DDoS Protection, và VPN doanh nghiệp.
- Cung cấp các công cụ và dịch vụ bảo mật cho tổ chức, đảm bảo tính an toàn dữ liệu.

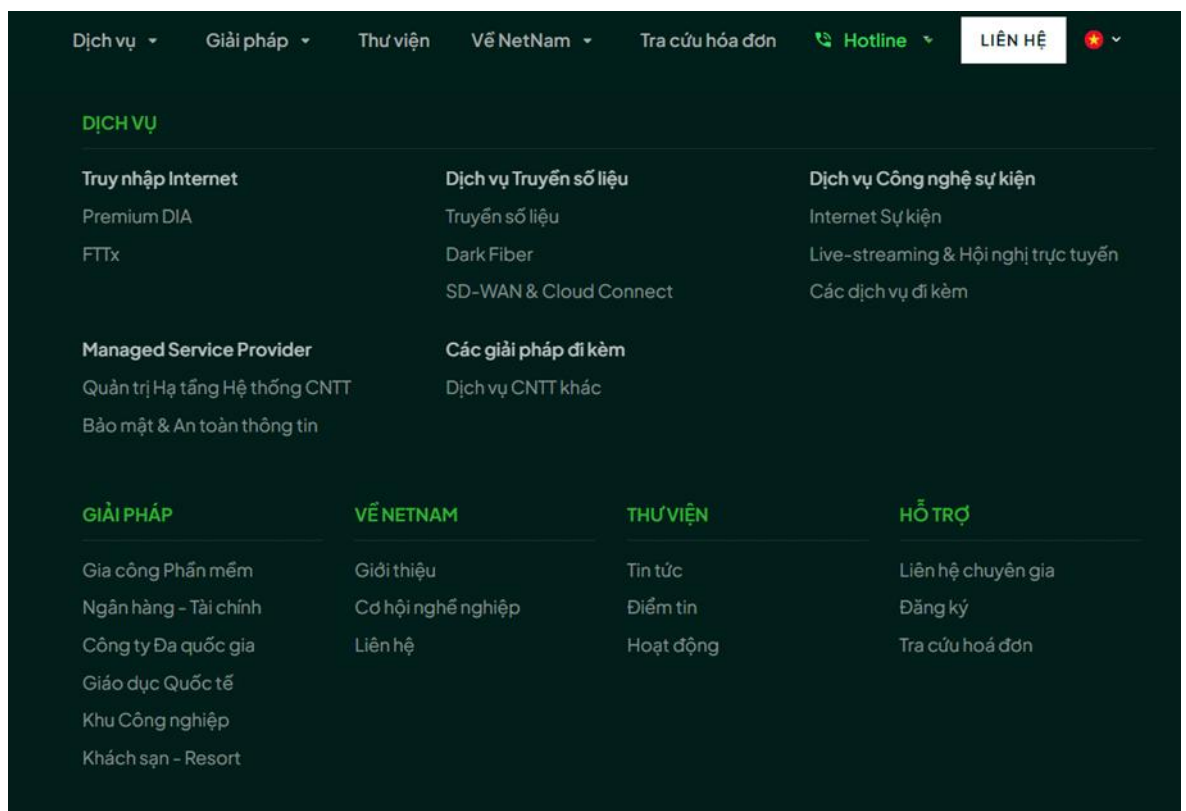
## F. Giải pháp truyền thông số (Digital solutions)

- Hỗ trợ các tổ chức trong việc triển khai chiến lược truyền thông số hiệu quả.
- Cung cấp dịch vụ quản lý nội dung và tối ưu hóa SEO.

## G. Hạ tầng mạng và giải pháp tích hợp CNTT

- Thiết kế và triển khai hệ thống mạng nội bộ (LAN/WAN).
- Tư vấn, lắp đặt và bảo trì các thiết bị mạng, hệ thống máy chủ, và trung tâm dữ liệu.

Theo NetNam [1], “NetNam đã triển khai thành công các dự án lớn về hệ thống quản lý thông tin và an toàn an ninh mạng (mạng diện rộng của Đảng Cộng sản, hệ thống thông tin UNDP, các bộ ngành, .v.v.).”



Hình 1-2 Các dịch vụ của công ty

### 1.3 Vị trí thực tập

Trong thời gian thực tập tại Công ty Cổ phần NetNam, em được bố trí thực tập tại Bộ phận phát triển giải pháp phần mềm dưới sự hướng dẫn trực tiếp của anh Đỗ Văn Ánh – TeamLead của nhóm. Vị trí thực tập của em thuộc nhóm kỹ thuật phát triển ứng dụng mobile, tập trung vào nghiên cứu và xây dựng ứng dụng thời tiết thời gian thực trên nền tảng Flutter.

Nhiệm vụ cụ thể bao gồm:

- Tìm hiểu và nắm vững công nghệ Flutter để phục vụ việc xây dựng giao diện ứng dụng.

- Nghiên cứu, khai thác và tích hợp API thời tiết WeatherAPI để lấy dữ liệu thời gian thực.
- Thiết kế và xây dựng giao diện người dùng (UI) thân thiện, trực quan, hỗ trợ chế độ sáng – tối (Dark Mode).
- Tích hợp chức năng định vị (GPS) để lấy thông tin thời tiết tại vị trí hiện tại của người dùng.
- Xây dựng tính năng tìm kiếm thành phố và hiển thị dự báo thời tiết theo ngày và tuần.
- Kiểm tra, khắc phục lỗi và tối ưu hiệu suất của ứng dụng.
- Viết báo cáo mô tả chi tiết quá trình nghiên cứu, phát triển và kết quả đạt được trong thời gian thực tập.

## 1.4 Đề tài thực tập

### 1.4.1 Tên đề tài

Xây dựng ứng dụng dự báo thời tiết thời gian thực với API WeatherAPI.

### 1.4.2 Giới thiệu đề tài

Đề tài “Xây dựng ứng dụng dự báo thời tiết thời gian thực với API WeatherAPI” nhằm phát triển một ứng dụng di động đa nền tảng bằng Flutter, hỗ trợ người dùng tra cứu thông tin thời tiết hiện tại và dự báo trong 7 ngày. Ứng dụng tích hợp WeatherAPI để lấy dữ liệu thời gian thực, định vị GPS để tự động hiển thị thời tiết theo vị trí người dùng, cùng các tính năng nâng cao như chuyển đổi nhiệt độ, Dark Mode và hiệu ứng giao diện thân thiện.

### 1.4.3 Mục tiêu đề tài

Đề tài “Xây dựng ứng dụng dự báo thời tiết thời gian thực với WeatherAPI” được thực hiện với các mục tiêu cụ thể như sau:

- Xây dựng một ứng dụng di động có khả năng hiển thị thông tin thời tiết hiện tại và dự báo trong 7 ngày, sử dụng dữ liệu từ WeatherAPI.
- Tích hợp các chức năng hiện đại như định vị GPS, tìm kiếm địa điểm, chuyển đổi đơn vị nhiệt độ (°C và °F), và chế độ Dark Mode.
- Thiết kế giao diện người dùng trực quan, thân thiện, kết hợp hiệu ứng chuyển cảnh để nâng cao trải nghiệm người dùng.

### 1.4.4 Công cụ hỗ trợ

STT	Công cụ / công nghệ	Mục đích sử dụng
1	Flutter	Framework để phát triển ứng dụng
2	Dart	Ngôn ngữ lập trình trong Flutter
3	WeatherAPI	Lấy dữ liệu thời tiết thời gian thực
4	Android Studio	Môi trường phát triển (IDE)

Bảng 1-1 Công cụ, công nghệ hỗ trợ

### 1.5 Lịch làm việc

Tuần	Thời gian	Công việc	Người hướng dẫn	Mức độ hoàn thành (SV tự đánh giá)
1	10/3 – 16/3	Đến gặp mặt chào hỏi công ty, làm quen với môi trường làm việc của công ty.  Nghe công ty hướng dẫn về kinh nghiệm làm việc cũng như văn hóa, tổ chức của công ty.	Đỗ Văn Ánh	Hoàn thành
2	17/3 – 23/3	Đọc tài liệu về sơ đồ cơ cấu tổ chức công ty và tìm hiểu các vị trí làm trong công ty.  Bước đầu nghiên cứu về đề tài Weather App.	Đỗ Văn Ánh	Hoàn thành
3	24/3 – 30/3	Tìm hiểu các kiến thức nền tảng về Flutter và những công nghệ xoay quanh để phục vụ cho đề tài.	Đỗ Văn Ánh	Hoàn thành
4	31/3 – 6/4	Thiết kế giao diện.	Đỗ Văn Ánh	Hoàn thành
5	7/4 – 13/4	Xây dựng giao diện cho đề tài với Flutter.	Đỗ Văn Ánh	Hoàn thành
6	14/4 – 20/4	Tích hợp API thời tiết (OpenWeather) để lấy dữ liệu thời tiết theo thời gian thực.	Đỗ Văn Ánh	Hoàn thành
7	21/4 – 27/4	Thêm tính năng chuyển đổi nhiệt độ ( $^{\circ}\text{C} \leftrightarrow ^{\circ}\text{F}$ ).  Hiển thị dự báo thời tiết theo ngày và tuần.	Đỗ Văn Ánh	Hoàn thành
8	28/4 – 4/5	Tích hợp GPS để lấy vị trí hiện tại của người	Đỗ Văn Ánh	Hoàn thành

		dùng và hiển thị thời tiết theo vị trí.  Thêm tính năng tìm kiếm thành phố theo tên.		
9	5/5 – 11/5	Tối ưu giao diện và hiệu ứng chuyển đổi.  Tích hợp Dark Mode cho ứng dụng.	Đỗ Văn Ánh	Hoàn thành
10	12/5 – 18/5	Kiểm tra và sửa lỗi.  Hoàn thiện báo cáo.	Đỗ Văn Ánh	Hoàn thành

Bảng 1-2 Bảng lịch làm việc theo từng tuần

## 1.6 Kết chương

Trong chương này, em đã trình bày tổng quan về đơn vị thực tập – Công ty Cổ phần NETNAM, cùng với vị trí, nhiệm vụ thực tập và đề tài được giao. Môi trường làm việc chuyên nghiệp tại công ty là điều kiện thuận lợi để em áp dụng kiến thức đã học vào thực tế.

Chương tiếp theo sẽ trình bày nội dung thực tập cụ thể, bao gồm các công nghệ được sử dụng và quá trình xây dựng ứng dụng thời tiết theo thời gian thực.



## **CHƯƠNG 2 NỘI DUNG THỰC TẬP**

### **2.1 Cơ sở lý thuyết**

#### **2.1.1 Tổng quan về lập trình ứng dụng di động**

##### **A. Khái niệm**

Theo MONA.Media [2], “Lập trình ứng dụng di động chính là việc sử dụng những ngôn ngữ lập trình để viết app, phát triển những ứng dụng để gia tăng tối đa các tiện ích cho các thiết bị di động mà người dùng đang dùng. Hiện nay có 2 hệ điều hành được sử dụng phổ biến là iOS và Android.”

Theo Teky [3], “Lập trình ứng dụng di động là viết ngôn ngữ lập trình (code) để xây dựng các tiện ích, chương trình, app, hệ điều hành trên mobile. Hiện nay, có 2 hệ điều hành trên Smartphone phổ biến nhất là Android và IOS.

Lập trình ứng dụng di động mang đến một không gian mua sắm, tin tức, giải trí thu nhỏ ngay trên Smartphone của mỗi cá nhân.”

Trong bối cảnh công nghệ thông tin không ngừng phát triển và ngày càng gắn bó mật thiết với đời sống con người, các thiết bị di động như điện thoại thông minh và máy tính bảng đã trở thành phương tiện phổ biến, không thể thiếu trong sinh hoạt hàng ngày. Chúng không chỉ đóng vai trò là công cụ liên lạc mà còn trở thành nền tảng chính để truy cập thông tin, giải trí, học tập và làm việc từ xa.

Chính vì vậy, nhu cầu phát triển các ứng dụng di động đang ngày càng gia tăng, đặc biệt là các ứng dụng mang tính tiện ích cao, phục vụ nhu cầu thiết thực của người dùng như: theo dõi sức khỏe, quản lý tài chính cá nhân, điều khiển thiết bị thông minh, tra cứu thông tin thời tiết,... Việc xây dựng và triển khai các ứng dụng di động không chỉ giúp tối ưu hóa trải nghiệm người dùng mà còn tạo điều kiện để các lập trình viên, đặc biệt là sinh viên ngành Công nghệ Thông tin, tiếp cận gần hơn với các công nghệ hiện đại như: lập trình đa nền tảng (cross-platform), tích hợp API, quản lý trạng thái (state management), và các tiêu chuẩn thiết kế giao diện người dùng (UI/UX) tiên tiến.

Sự kết hợp giữa lý thuyết được học trong nhà trường và thực tiễn phát triển ứng dụng di động không chỉ giúp sinh viên nâng cao kỹ năng chuyên môn mà còn mở rộng tư duy công nghệ, tăng khả năng thích ứng với xu hướng chuyển đổi số đang diễn ra mạnh mẽ hiện nay. Ưu điểm của lập trình ứng dụng di động:

- Dễ dàng tiếp cận nhanh chóng người dùng: Ứng dụng di động giúp doanh nghiệp, tổ chức và cá nhân đưa sản phẩm/dịch vụ đến gần người dùng chỉ với vài thao tác trên smartphone.
- Tính linh hoạt cao: Người dùng có thể sử dụng ứng dụng mọi lúc, mọi nơi mà không bị giới hạn bởi không gian hay thời gian.

- Tích hợp nhiều công nghệ hiện đại: Ứng dụng mobile có thể kết hợp với các công nghệ như GPS, cảm biến, AI, chatbot, thanh toán điện tử... để tạo ra trải nghiệm phong phú và thông minh.
- Tăng khả năng cá nhân hóa: Ứng dụng có thể ghi nhớ thói quen, hành vi người dùng để đưa ra gợi ý phù hợp, tăng mức độ hài lòng và gắn bó.
- Khả năng cập nhật nhanh: Các phiên bản ứng dụng mới có thể dễ dàng được cập nhật để sửa lỗi hoặc bổ sung tính năng mới.

Vai trò của lập trình ứng dụng di động:

- Thúc đẩy chuyển đổi số: Ứng dụng mobile giúp các doanh nghiệp số hóa quy trình kinh doanh, bán hàng, chăm sóc khách hàng và quản lý nội bộ.
- Tạo ra thị trường việc làm rộng mở: Nhu cầu nhân lực lập trình mobile tăng mạnh, mở ra cơ hội nghề nghiệp lớn cho sinh viên công nghệ thông tin.
- Góp phần nâng cao chất lượng cuộc sống: Các ứng dụng về y tế, giáo dục, giao thông, tài chính, thời tiết, giải trí... giúp người dùng tiết kiệm thời gian và tiếp cận thông tin dễ dàng hơn.
- Đóng vai trò trung tâm trong hệ sinh thái công nghệ: Ứng dụng di động là cầu nối giữa phần cứng (thiết bị) và dịch vụ số (dịch vụ online, đám mây), tạo nên hệ sinh thái công nghệ toàn diện.

### **2.1.2 Giới thiệu về Flutter và ngôn ngữ Dart**

#### **A. Khái niệm và đặc điểm của Dart**

##### **a. Khái niệm**

Theo ITViec [3] , “Dart là một ngôn ngữ lập trình hướng đối tượng do Google phát triển, được thiết kế để tối ưu cho việc xây dựng giao diện người dùng (UI) hiện đại trên nhiều nền tảng. Đây là ngôn ngữ chính được sử dụng trong Flutter – bộ công cụ phát triển ứng dụng đa nền tảng.”

##### **b. Đặc điểm nổi bật của Dart**

- Hướng đối tượng hoàn toàn: Dart hỗ trợ lập trình theo mô hình đối tượng với class, object, inheritance (kế thừa), abstract (trừu tượng), mixin...
- Cú pháp quen thuộc: Tương tự các ngôn ngữ như Java, JavaScript hoặc C#, nên dễ tiếp cận với sinh viên ngành công nghệ thông tin.
- Hiệu suất cao: Dart được biên dịch thành mã máy (native code) nên tốc độ xử lý nhanh, phù hợp cho các ứng dụng cần hiệu năng như mobile.
- Hỗ trợ lập trình bất đồng bộ: Dart có Future, async, await, và Stream, giúp xử lý các tác vụ như gọi API hoặc đọc file một cách hiệu quả.

- Có thể chạy trên nhiều môi trường: Dart hỗ trợ cả phía client (ứng dụng Flutter) lẫn phía server (như Node.js), và có thể biên dịch sang JavaScript để chạy trên trình duyệt.



Hình 2-1 Logo của Dart

## **B. Khái niệm và đặc điểm của Flutter**

### ***a. Khái niệm***

Theo aws [4], “Flutter là một framework mã nguồn mở do Google phát triển, hỗ trợ lập trình giao diện người dùng cho nhiều nền tảng khác nhau chỉ với một mã nguồn duy nhất, giúp rút ngắn thời gian và chi phí phát triển phần mềm.

Tại thời điểm ra mắt vào năm 2018, Flutter chủ yếu hỗ trợ phát triển ứng dụng di động. Hiện nay, Flutter hỗ trợ phát triển ứng dụng trên sáu nền tảng: iOS, Android, web, Windows, MacOS và Linux.”



Hình 2-2 Logo của Flutter

### ***b. Đặc điểm của Flutter***

- Phát triển đa nền tảng (cross-platform): Viết một lần, chạy được trên Android, iOS, web và desktop.
- Ngôn ngữ Dart: Sử dụng Dart – ngôn ngữ hiện đại, dễ tiếp cận với lập trình viên đã quen Java, JavaScript, hoặc C#.

- Hot Reload: Thay đổi mã và xem kết quả ngay lập tức mà không cần biên dịch lại toàn bộ ứng dụng.
- Hiệu suất cao: Ứng dụng Flutter chạy gần như native vì không cần bridge như các framework hybrid khác.
- Thư viện widget phong phú: Giao diện được xây dựng hoàn toàn bằng widget giúp tùy biến dễ dàng.

### **2.1.3 Giao tiếp với API**

#### **A. Khái niệm**

Theo aws [5], “API là một công cụ trung gian cho phép các phần mềm tương tác và truyền tải dữ liệu lẫn nhau. Trong ứng dụng thời tiết, API được dùng để lấy dữ liệu thời tiết từ máy chủ và hiển thị đến người dùng.”

Theo 200Lab [6], “API - Application Programming Interface hay Giao diện lập trình ứng dụng, là một tập hợp các định nghĩa và giao thức cho phép các ứng dụng phần mềm giao tiếp và tương tác với nhau. Chắc bạn đã từng nghe về định nghĩa của User Interface (Giao diện người dùng) là nơi mà User sẽ tương tác với ứng dụng của chúng ta, cũng tương tự như thế khi các phần mềm “nói chuyện” với nhau cũng cần thông qua giao diện đặc biệt có tên gọi là API.”

#### **B. Giao tiếp với HTTP và dữ liệu JSON**

Khi ứng dụng gửi yêu cầu đến API (thường là HTTP request), máy chủ sẽ phản hồi về một khối dữ liệu, thường là JSON (JavaScript Object Notation) – định dạng dữ liệu nhẹ, dễ phân tích.

Các thành phần chính trong một chuỗi giao tiếp:

- Endpoint: Đường dẫn đến tài nguyên dữ liệu.
- Phương thức HTTP: GET, POST, PUT, DELETE... Trong ứng dụng thời tiết thường dùng GET để lấy thông tin.
- Header và Parameters: Dùng để xác thực hoặc định nghĩa yêu cầu cụ thể (ví dụ: API key, đơn vị nhiệt độ, ngôn ngữ...).

### **2.1.4 Xử lý dữ liệu thời tiết**

Sau khi nhận được phản hồi từ hệ thống API thời tiết (WeatherAPI), một trong những bước quan trọng và không thể thiếu trong quá trình phát triển ứng dụng là phân tích, bóc tách và xử lý dữ liệu để hiển thị một cách chính xác, trực quan và dễ hiểu đối với người dùng cuối. Đây không chỉ đơn thuần là việc hiển thị các con số, mà còn đòi hỏi lập trình viên phải hiểu rõ cấu trúc dữ liệu, xác định chính xác những trường thông tin cần thiết, đồng thời tổ chức và trình bày thông tin sao cho phù hợp với giao diện và trải nghiệm người dùng.

WeatherAPI trả về dữ liệu dưới định dạng JSON (JavaScript Object Notation) – một chuẩn định dạng phổ biến dùng để trao đổi dữ liệu giữa client và server. Dữ liệu JSON từ WeatherAPI rất phong phú và chi tiết, bao gồm:

- Thông tin thời tiết hiện tại (nhiệt độ, độ ẩm, sức gió, mô tả trạng thái thời tiết, biểu tượng minh họa, cảm giác như,...)
- Dự báo thời tiết theo ngày và theo giờ (gồm cả nhiệt độ cao/thấp, trạng thái, gió, độ ẩm theo từng giờ)
- Thông tin địa lý (tên thành phố, quốc gia, múi giờ, vĩ độ – kinh độ,...)
- Thời gian mặt trời mọc – lặn, tình trạng mây, chỉ số UV, và nhiều trường thông tin bổ sung khác.

Việc bóc tách các trường dữ liệu từ JSON yêu cầu lập trình viên phải nắm rõ cấu trúc của phản hồi JSON, biết cách xử lý dữ liệu lồng nhau (nested data), và biến đổi chúng thành các kiểu dữ liệu phù hợp với ngôn ngữ lập trình đang sử dụng (trong trường hợp này là Dart – ngôn ngữ của Flutter).

Sau khi xử lý, dữ liệu sẽ được hiển thị trực quan trên giao diện người dùng. Ứng dụng cần đảm bảo:

- Các trường thông tin chính phải rõ ràng và dễ đọc (ví dụ: nhiệt độ hiển thị lớn, mô tả thời tiết kèm icon).
- Các dữ liệu phụ như độ ẩm, gió, thời gian bình minh/hoàng hôn... được trình bày gọn gàng, dễ theo dõi, thường kết hợp với biểu tượng trực quan để tăng tính thẩm mỹ và khả năng nhận diện nhanh.
- Đảm bảo đồng bộ dữ liệu giữa các thành phần giao diện khi người dùng chuyển đổi thành phố, đơn vị nhiệt độ hoặc bật chế độ định vị GPS.

Tóm lại, việc phân tích và hiển thị dữ liệu từ API không chỉ là một thao tác kỹ thuật đơn thuần mà còn đóng vai trò then chốt trong việc tạo nên trải nghiệm người dùng tốt, góp phần làm cho ứng dụng trở nên thực tiễn, hiệu quả và chuyên nghiệp hơn. Tích hợp GPS trong ứng dụng Flutter

Theo TOYOTA [7], “GPS là viết tắt của cụm từ Global Positioning System (hệ thống định vị toàn cầu), là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ phát triển, vận hành và quản lý.

Hệ thống định vị toàn cầu GPS được tạo thành từ một mạng lưới gồm 30 vệ tinh quay xung quanh Trái Đất. Ban đầu, định vị GPS được sử dụng cho các ứng dụng quân sự. Tuy nhiên vào những năm 1980, Chính phủ Hoa Kỳ đã cung cấp hệ thống này cho mục đích dân sự. GPS có thể cung cấp thông tin về vị trí và thời gian liên tục 24 giờ, trong mọi điều kiện thời tiết, ở bất kỳ đâu trên thế giới.”

## **2.2 Kết quả làm được**

### **2.2.1 Tích hợp WeatherAPI để lấy dữ liệu thời tiết thời gian thực**

- Ứng dụng sử dụng WeatherAPI từ trang weatherapi.com để lấy dữ liệu thời tiết.
- Tạo một lớp WeatherService đảm nhiệm các chức năng:
  - Gửi yêu cầu HTTP đến API.

- Lấy dữ liệu JSON và chuyển về dạng Map<String, dynamic> để xử lý.
- Gọi đúng endpoint để lấy:
  - Thời tiết hiện tại.
  - Dự báo thời tiết 7 ngày.
  - Gợi ý tên thành phố (cho chức năng tìm kiếm).
- Ưu điểm:
  - Tách biệt logic xử lý với giao diện
  - Tái sử dụng dễ dàng ở nhiều màn hình
  - Có thể mở rộng nếu muốn dùng thêm các API khác của WeatherAPI như: thời gian mặt trời mọc, chỉ số UV, chất lượng không khí,....

### **2.2.2 Hiện thị chi tiết thời gian hiện tại**

Màn hình chính là nơi người dùng theo dõi nhanh các thông tin thời tiết hiện tại của thành phố được chọn hoặc vị trí hiện tại (nếu dùng GPS). Giao diện được thiết kế trực quan, dễ sử dụng, tối ưu hóa trải nghiệm người dùng.

Nội dung hiển thị trên màn hình chính:

- Tên thành phố: nằm ở phía trên cùng, giúp người dùng biết thông tin thời tiết đang được hiển thị ở đâu, địa phương nào.
- Nhiệt độ hiện tại: được hiển thị cỡ chữ lớn, dễ nhìn. Người dùng có thể chuyển đổi giữa đơn vị °C và °F bằng một nút bấm đơn giản.
- Mô tả điều kiện thời tiết: Dữ liệu được lấy từ API và giúp người dùng hiểu ngay thời tiết thực tế.
- Biểu tượng thời tiết: Lấy hình ảnh minh họa (icon) tương ứng với mô tả thời tiết hiện tại.
- Các thông tin khác:
  - Độ ẩm hiện tại (%).
  - Tốc độ gió (km/h).
  - Giờ bình minh và hoàng hôn.

#### **A. Giao diện màn hình chính**

- Giao diện giúp người dùng có thể xem tổng quan thời tiết hiện tại như nhiệt độ cao nhất, thấp nhất, thời gian bình minh bắt đầu, thời gian hoàng hôn, độ ẩm và tốc độ gió thời điểm hiện tại của các thành phố lớn trên thế giới.
- Bên trên cùng là thanh ứng dụng bao gồm tiêu đề (tên thành phố) và các icon tìm kiếm, định vị và chuyển đổi nhiệt độ.
- Bên dưới là một nút bấm vào để xem thời tiết 7 ngày tiếp theo.



Hình 2-3 Giao diện chính của ứng dụng

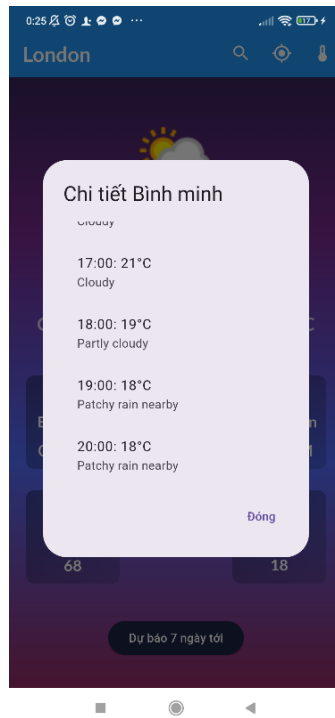
### B. Giao diện xem nhiệt độ trong ngày thông qua thời gian bình minh và hoàng hôn

Trên màn hình chính, bên cạnh các thông tin quen thuộc như nhiệt độ, độ ẩm, tốc độ gió,... ứng dụng còn hiển thị thời gian bình minh và hoàng hôn dưới dạng các khối thông tin trực quan, được thiết kế nổi bật với biểu tượng minh họa riêng biệt. Mỗi khối đều hiển thị rõ thời gian cụ thể (ví dụ: 05:35 AM, 18:12 PM), giúp người dùng dễ dàng nhận biết thời điểm chuyển giao trong ngày.

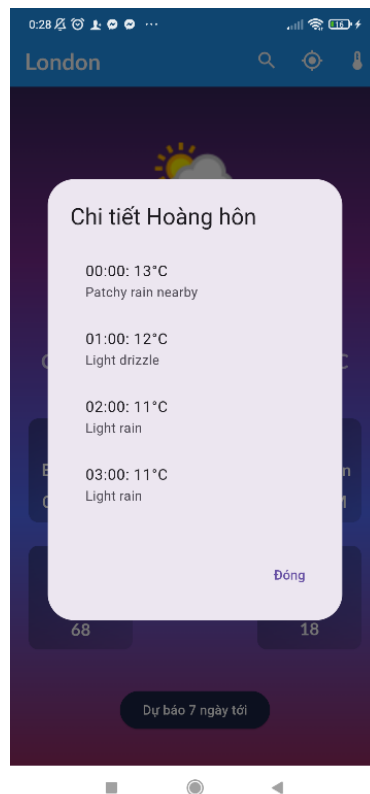
Các khối này không chỉ mang tính thông tin mà còn đóng vai trò tương tác: khi người dùng nhấn vào, ứng dụng sẽ mở ra giao diện hiển thị chi tiết thời tiết theo từng giờ trong ngày tương ứng. Việc tích hợp thông tin về bình minh và hoàng hôn giúp giao diện trở nên sinh động, đồng thời mang lại nhiều giá trị thực tiễn cho người dùng trong việc lập kế hoạch sinh hoạt, đi lại, làm việc ngoài trời,...

Khi người dùng nhấn vào một trong hai mục:

- Ứng dụng sẽ mở hộp thoại hoặc màn hình hiển thị thông tin chi tiết về thời tiết theo từng giờ trong ngày đó.
- Các dữ liệu do WeatherAPI cung cấp, bao gồm đầy đủ 24 khung giờ.



Hình 2-4 Giao diện nhiệt độ ngày thông qua chi tiết bình minh



Hình 2-5 Giao diện nhiệt độ ngày thông qua chi tiết hoàng hôn.

### C. Xem giao diện độ ẩm trong ngày

Một trong những thông tin quan trọng trong ứng dụng thời tiết là độ ẩm – yếu tố ảnh hưởng trực tiếp đến cảm giác nóng ẩm, khả năng mưa, và chất lượng không khí. Ứng dụng được xây dựng đã cung cấp đầy đủ thông tin về độ ẩm ở hai cấp độ:

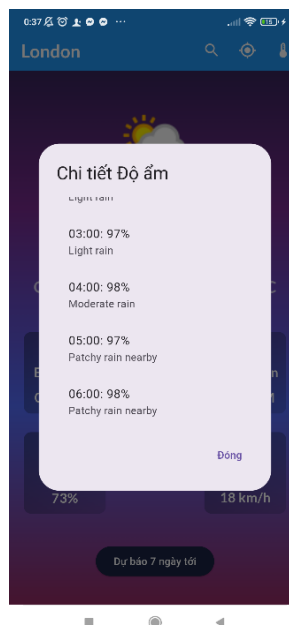
- Độ ẩm hiện tại



- Trên màn hình chính, người dùng có thể quan sát được độ ẩm hiện tại của địa phương đang được theo dõi.
- Dữ liệu được lấy từ trường `current.humidity` trong API WeatherAPI.
- Độ ẩm được hiển thị dưới dạng phần trăm, ví dụ: Độ ẩm: 72%.
- Giao diện trình bày rõ ràng, đặt gần các chỉ số thời tiết khác như nhiệt độ, gió, UV để người dùng dễ dàng theo dõi tổng quan.
- Độ ẩm theo từng giờ trong ngày
  - Khi người dùng nhấn vào một ngày bất kỳ trong màn hình dự báo 7 ngày, ứng dụng sẽ chuyển sang màn hình chi tiết của ngày đó.
  - Tại đây, danh sách thời tiết theo giờ được hiển thị.
  - Mỗi giờ bao gồm thông tin: thời gian, nhiệt độ, tốc độ gió, và độ ẩm theo giờ (ví dụ: "14:00 - 65%").
  - Người dùng có thể cuộn dọc để xem toàn bộ 24 giờ trong ngày và nắm rõ sự biến đổi độ ẩm trong ngày.

Ý nghĩa:

- Việc hiển thị độ ẩm hiện tại và theo giờ giúp người dùng:
- Dự đoán thời tiết thay đổi như thế nào trong ngày.
- Đưa ra quyết định hợp lý trong việc sinh hoạt, tập thể thao, đi lại hoặc bảo quản thiết bị điện tử.
- Nhận biết khi nào độ ẩm lên cao – dễ gây nồm ẩm, khó chịu.



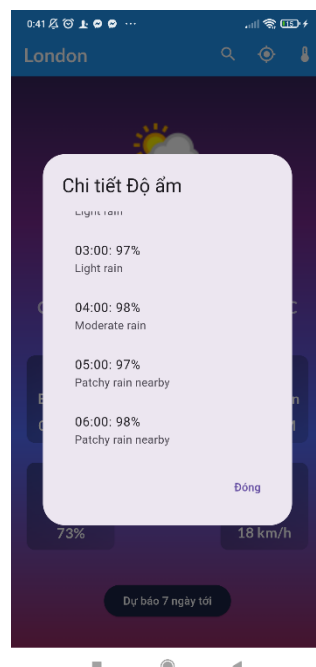
Hình 2-6 Giao diện chi tiết độ ẩm trong ngày theo giờ.

#### **D. Giao diện xem tốc độ gió trong ngày**

Bên cạnh nhiệt độ và độ ẩm, tốc độ gió là một yếu tố quan trọng trong việc cảm nhận thời tiết thực tế, đặc biệt đối với các hoạt động ngoài trời, du lịch hoặc trong các lĩnh vực như hàng

không, hàng hải. Ứng dụng thời tiết được xây dựng đã tích hợp đầy đủ chức năng hiển thị tốc độ gió tại hai cấp độ:

- Tốc độ gió tại thời điểm hiện tại
  - Ngay tại màn hình chính, tốc độ gió thời điểm hiện tại được hiển thị rõ ràng.
  - Tốc độ gió được thể hiện bằng đơn vị km/h.
  - Giá trị được làm tròn bằng .round() để đảm bảo dễ nhìn và trực quan.
- Tốc độ gió theo từng giờ trong ngày
  - Trên màn hình chi tiết của từng ngày, tốc độ gió được cập nhật cho từng giờ trong ngày, tương ứng với 24 giá trị cho 24 giờ.
  - Thông tin tốc độ gió từng giờ hiển thị cùng thời gian, độ ẩm và nhiệt độ, giúp người dùng dễ dàng đối chiếu.



Hình 2-7 Giao diện chi tiết tốc độ gió trong ngày.

### 2.2.3 Dự báo thời tiết 7 ngày

Màn hình này thiết kế để hiển thị danh sách dự báo thời tiết cho 7 ngày tiếp theo, dựa trên dữ liệu WeatherAPI. Mỗi mục trong danh sách đại diện cho một ngày cụ thể, cung cấp các thông tin tổng quan giúp người dùng nắm bắt nhanh tình hình thời tiết thời gian sắp tới.

- Màn hình chi tiết hiển thị
  - Giờ trong ngày: ví dụ 00:00, 01:00,... đến 23:00.
  - Nhiệt độ tại mỗi giờ
  - Tốc độ gió (km/h) tại mỗi khung giờ.
  - Độ ẩm (%) tại các khung giờ trong ngày.
  - Trạng thái thời tiết tại thời điểm đó (mô tả + icon).

Thông tin thời tiết theo giờ giúp người dùng có kế hoạch chi tiết hơn cho từng thời điểm trong ngày ( ví dụ: chuẩn bị ra đường vào buổi trưa nắng nóng, hoặc đề phòng những cơn mưa bất chợt)

### A. Giao diện thời tiết 7 ngày tới

- Khi người dùng nhấn xem thời tiết 7 ngày tới ở màn hình chính thì ngay lập tức chuyển qua giao diện xem thời tiết 7 ngày tiếp theo.
- Tại đây, người dùng xem được thời tiết từng ngày, nhiệt độ trung bình, nhiệt độ cao nhất, nhiệt độ thấp nhất, mưa, nắng, mây,... theo ngày.
- Khi người dùng nhấn vào một ngày nào đó trong 7 ngày thì lập tức đưa người dùng qua giao diện xem chi tiết thời tiết của ngày hôm đó.



Hình 2-8 Giao diện màn hình thời tiết 7 ngày tiếp theo.

### B. Giao diện thời tiết chi tiết những ngày tiếp theo

Màn hình này được thiết kế nhằm cung cấp thông tin chi tiết cho những ngày tiếp trong danh sách dự báo 7 ngày. Người dùng có thể truy cập màn hình bằng cách chạm vào một ngày bất kỳ trong danh sách dự báo. Giao diện được xây dựng bố cục rõ ràng, dễ đọc, hỗ trợ cả chế độ sáng và tối.

Các thông tin chính hiển thị trong màn hình chi tiết:


- Nhiệt độ theo từng giờ
  - Ứng dụng hiển thị đầy đủ dữ liệu nhiệt độ của 24 khung giờ trong ngày (từ 00:00 đến 23:00).
  - Nhiệt độ được hiển thị kèm đơn vị (°C hoặc °F) và có thể chuyển đổi linh hoạt.
  - Mỗi mục bao gồm biểu tượng thời tiết và nhiệt độ tại thời điểm đó.

- Nhiệt độ trung bình trong ngày
  - Nằm ở đầu màn hình, giúp người dùng hình dung được xu hướng thời tiết tổng thể trong ngày.
  - Được tính dựa trên giá trị trung bình của toàn bộ mảng `hour[]` trong ngày đó.
- Độ ẩm theo từng giờ
  - Mỗi giờ được hiển thị kèm chỉ số độ ẩm (dưới dạng phần trăm – %).
  - Giúp người dùng nhận biết thời điểm không khí khô hanh hay ẩm ướt trong ngày.
- Tốc độ gió theo từng giờ
  - Mỗi giờ hiển thị thêm tốc độ gió (đơn vị: km/h), hỗ trợ người dùng có các hoạt động ngoài trời an toàn hơn.
  - Giá trị được làm tròn và dễ đọc.
- Trạng thái thời tiết theo giờ
  - Ứng dụng hiển thị mô tả chi tiết trạng thái thời tiết tại mỗi thời điểm trong ngày, ví dụ: “Nắng nhẹ”, “Trời nhiều mây”, “Mưa nhỏ”,...
  - Kèm theo biểu tượng minh họa tương ứng (☀️ ☁️ 🌧️ v.v.) được lấy từ API.




Hình 2-9 Giao diện màn hình thời tiết chi tiết của ngày được xem.

### 2.2.4 Tìm kiếm thành phố

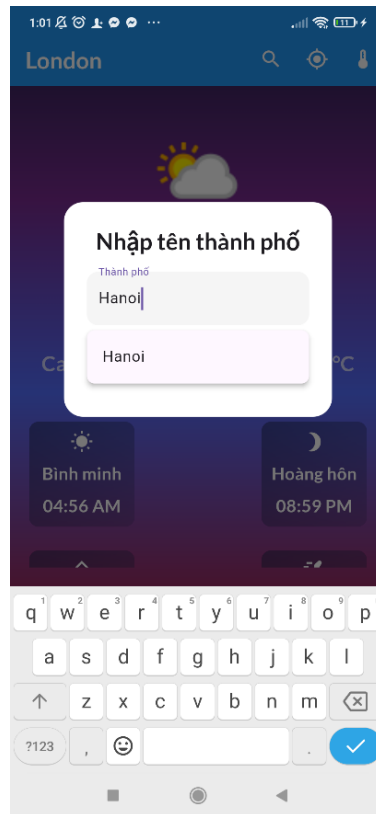
- Người dùng có thể nhấn vào biểu tượng  để mở khung tìm kiếm.
- Giao diện tìm kiếm trong ứng dụng được xây dựng bằng cách sử dụng widget `flutter_typeahead`, một thư viện mạnh mẽ hỗ trợ gợi ý theo thời gian thực khi người dùng nhập liệu. Tính năng này giúp nâng cao trải nghiệm người dùng trong việc tìm kiếm và lựa chọn thành phố cần xem thông tin thời tiết.
- Cơ chế hoạt động:
  - Khi người dùng bắt đầu gõ một vài ký tự đầu tiên của tên thành phố, widget `TypeAheadField` sẽ tự động gửi yêu cầu HTTP (request) đến endpoint tìm kiếm của WeatherAPI.
  - Dữ liệu trả về là danh sách các thành phố có tên gần giống với từ khóa mà người dùng vừa nhập.
  - Các gợi ý được hiển thị thành một danh sách động bên dưới ô tìm kiếm.
- Tương tác người dùng:
  - Người dùng có thể cuộn danh sách và chọn một thành phố cụ thể từ các gợi ý.
  - Ngay sau khi thành phố được chọn, ứng dụng sẽ tự động gửi yêu cầu đến API chính của WeatherAPI để lấy toàn bộ thông tin mới nhất (gồm thời tiết hiện tại và dự báo 7 ngày tới) tương ứng với thành phố đó.
  - Giao diện chính sẽ tự động cập nhật để hiển thị thông tin thời tiết cho thành phố mới được chọn.
- Ưu điểm của cách triển khai:
  - Tránh lỗi nhập sai tên thành phố vì đã có gợi ý
  - Truy xuất dữ liệu theo thời gian thực, không cần danh sách địa phương cứng trong ứng dụng.
  - Tối ưu hóa bằng cách chỉ gửi yêu cầu API khi có nhập liệu và khi chọn.

#### A. Giao diện tìm kiếm thành phố

Ứng dụng cung cấp một giao diện tìm kiếm đơn giản, trực quan để người dùng có thể dễ dàng tra cứu thông tin thời tiết của bất kỳ thành phố nào trên thế giới.

- Cách sử dụng:
  - Khi người dùng nhấn vào biểu tượng  (tìm kiếm) trên thanh tiêu đề, một hộp thoại hiện ra giữa màn hình.
  - Giao diện hiển thị trường nhập liệu với tiêu đề “Nhập tên thành phố”, cùng một ô nhập và gợi ý bên dưới.
- Tương tác người dùng:
  - Người dùng bắt đầu gõ tên thành phố vào ô tìm kiếm.
  - Khi nhập từ khóa, ứng dụng sẽ tự động gọi API gợi ý tên thành phố (autocomplete) từ WeatherAPI.

- Các gợi ý hiện ra bên dưới theo thời gian thực, người dùng có thể chọn một trong số đó.
- Sau khi chọn:
  - Khi người dùng nhấn “Xác nhận”, giao diện chính sẽ cập nhật:
    - Thay đổi tên thành phố hiển thị.
    - Lấy và hiển thị thời tiết hiện tại, dự báo, nhiệt độ, độ ẩm, gió,... cho thành phố vừa chọn.



Hình 2-10 Giao diện tìm kiếm thành phố



Hình 2-11 Kết quả tìm kiếm

### 2.2.5 Tính năng chuyển đổi giữa °C và °F

- Ứng dụng có nút chuyển đổi đơn vị nhiệt độ.
- Khi nhấn, nhiệt độ trên giao diện (nhiệt độ hiện tại, nhiệt độ cao nhất, nhiệt độ thấp nhất) sẽ chuyển từ độ C sang độ F hoặc ngược lại.
- Việc chuyển đổi được xử lý ngay trong ứng dụng bằng công thức:
  - $\text{double cToF}(\text{double c}) \Rightarrow (c * 9/5) + 32;$
  - $\text{double fToC}(\text{double f}) \Rightarrow (f - 32) * 5/9;$
- Sử dụng công thức để không cần gọi lại API, giúp tăng hiệu suất và giảm độ trễ.

#### A. Giao diện chuyển đổi giữa °C và °F

Ứng dụng cho phép người dùng linh hoạt chuyển đổi giữa hai đơn vị nhiệt độ phổ biến là Celsius (°C) và Fahrenheit (°F) một cách nhanh chóng và trực quan.

- Cách thao tác:
  - Trên màn hình chính, người dùng có thể dễ dàng nhận thấy biểu tượng nhiệt kế 🌡️ nằm ở góc trên bên phải của thanh tiêu đề.
  - Khi nhấn vào biểu tượng này, đơn vị nhiệt độ sẽ lập tức được chuyển đổi trên toàn bộ màn hình.
    - Nếu đang hiển thị theo độ C (°C), ứng dụng sẽ chuyển sang độ F (°F).
    - Ngược lại, nếu đang hiển thị theo độ F (°F), sẽ chuyển ngược lại thành độ C (°C).

- Cách hiển thị sau khi chuyển đổi:
  - Tất cả giá trị liên quan đến nhiệt độ, bao gồm:
    - Nhiệt độ hiện tại
    - Nhiệt độ cao nhất và thấp nhất trong ngày
    - Nhiệt độ theo giờ (nếu đang ở màn hình chi tiết)
- Cơ chế kỹ thuật:
  - Khi người dùng nhấn biểu tượng, một biến trạng thái sẽ được cập nhật và giao diện được gọi để hiển thị lại dữ liệu với định dạng mới.
  - Các nhiệt độ được định dạng và làm tròn thông minh: hiển thị số nguyên nếu là số tròn (ví dụ 22°C), hoặc kèm một chữ số thập phân nếu cần (ví dụ 74.3°F).
- Lợi ích:
  - Cho phép người dùng tùy chọn đơn vị quen thuộc theo vùng miền hoặc thói quen.
  - Tránh gây nhầm lẫn khi sử dụng ứng dụng tại các quốc gia dùng hệ đo khác nhau.
  - Giao diện thay đổi linh hoạt, không làm ảnh hưởng đến dữ liệu hoặc hiệu suất.



Hình 2-12 Nhiệt độ ở dạng °C





Hình 2-13 Giao diện nhiệt độ ở dạng °F

### 2.2.6 Tích hợp GPS để lấy thời tiết theo vị trí hiện tại

Ứng dụng tích hợp tính năng định vị vị trí hiện tại của người dùng nhằm cung cấp dữ liệu thời tiết chính xác tại nơi họ đang đứng, thay vì chỉ dựa vào tên thành phố nhập thủ công. Chức năng này được xây dựng dựa trên thư viện mã nguồn mở phổ biến của Flutter là geolocator.

- Cách hoạt động:
  - Ứng dụng sử dụng thư viện geolocator để truy cập vị trí hiện tại của thiết bị.
  - Khi người dùng nhấn vào biểu tượng định vị (📍), ứng dụng sẽ:
    - Kiểm tra xem dịch vụ định vị của thiết bị đã được bật hay chưa.
    - Yêu cầu quyền truy cập vị trí từ người dùng (nếu chưa cấp).
    - Nếu được cấp phép, lấy tọa độ vĩ độ và kinh độ chính xác của thiết bị (latitude, longitude).
- Gửi yêu cầu tới API:
  - Sau khi có tọa độ GPS, ứng dụng sẽ gửi yêu cầu HTTP đến WeatherAPI theo đúng định dạng tham số: `&q=$latitude,$longitude`
  - Đây là tính năng của WeatherAPI cho phép truy xuất dữ liệu thời tiết theo vị trí GPS thay vì tên thành phố.
  - Nhờ vậy, ứng dụng có thể hiển thị thời tiết chính xác tại vị trí hiện tại của người dùng, bao gồm:
    - Nhiệt độ.

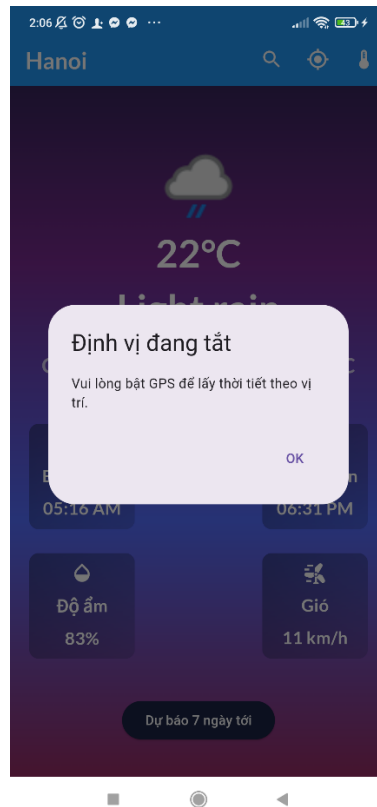
- Trạng thái thời tiết.
- Dự báo 7 ngày tại địa điểm đang sử dụng.
- Trường hợp không được cấp quyền:
  - Nếu người dùng từ chối cấp quyền truy cập vị trí hoặc dịch vụ định vị đang bị tắt:
    - Ứng dụng sẽ không cố gắng truy cập vị trí nữa
    - Một thông báo lỗi sẽ được hiển thị, ví dụ:
      - "Không thể lấy vị trí hiện tại. Vui lòng bật GPS hoặc cấp quyền truy cập vị trí cho ứng dụng."

Việc này giúp tránh crash ứng dụng và cải thiện trải nghiệm người dùng trong các tình huống bất ngờ.

#### A. Giao diện khi lấy thời tiết hiện tại có tích hợp GPS



Hình 2-14 Giao diện khi chưa bật GPS



Hình 2-15 Giao diện thông báo GPS chưa bật



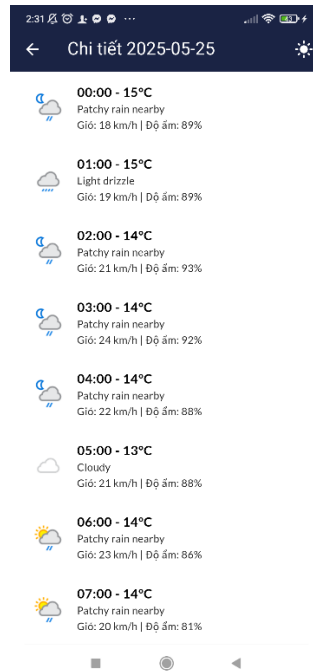
Hình 2-16 Giao diện khi bật GPS và đã cập nhật vị trí

### 2.2.7 Giao diện hỗ trợ Dark Mode

Ứng dụng được thiết kế để hỗ trợ cả chế độ sáng (Light Mode) và chế độ tối (Dark Mode), đảm bảo khả năng sử dụng tốt trong mọi điều kiện ánh sáng và mang lại trải nghiệm trực quan, dễ chịu cho người dùng.

- Thiết kế màu sắc:
  - Ứng dụng lựa chọn tông màu hài hòa và có độ tương phản cao để tối ưu hóa khả năng hiển thị:
    - Trong chế độ sáng, nền sử dụng màu trắng kết hợp với các màu biểu tượng tươi sáng, chữ màu đen hoặc xám đậm giúp dễ đọc và rõ ràng.
    - Trong chế độ tối, nền chuyển sang màu đen hoặc xám rất đậm, chữ chuyển sang màu trắng hoặc trắng mờ để tạo độ tương phản mạnh, giúp hiển thị tốt trong điều kiện thiếu sáng.
- Đảm bảo hiển thị:
  - Chữ trắng nổi bật trên nền tối, không bị mờ hoặc hòa màu.
  - Icon và hình ảnh thời tiết hiển thị rõ ràng, không bị mờ khi chuyển chế độ.
  - Không gây chói mắt vào ban đêm, phù hợp sử dụng vào buổi tối hoặc trong môi trường thiếu sáng.
- Chuyển đổi chế độ thủ công:
  - Người dùng có thể chuyển đổi nhanh giữa hai chế độ bằng cách nhấn vào biểu tượng nằm ở góc trên bên phải:
    - Khi biểu tượng hiển thị là ☀ (mặt trời), ứng dụng đang ở chế độ sáng (Light Mode).
    - Khi biểu tượng hiển thị là 🌙 (mặt trăng), ứng dụng đang ở chế độ tối (Dark Mode).
- Khi nhấn vào biểu tượng:
  - Toàn bộ giao diện của ứng dụng sẽ chuyển đổi tức thời giữa hai chế độ.
  - Không cần khởi động lại app hay load lại dữ liệu.
  - Màu nền, màu chữ, icon, và các thành phần tương ứng được cập nhật lại để phù hợp với chế độ được chọn.

## A. Giao diện Light Mode



Hình 2-17 Giao diện Light Mode

## B. Giao diện Dark Mode



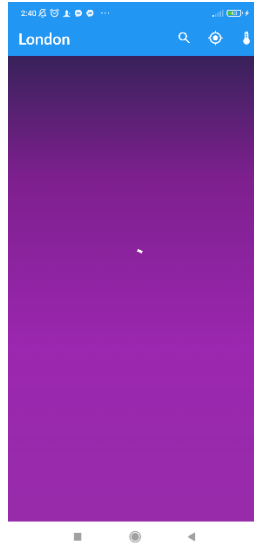
Hình 2-18 Giao diện Dark Mode

### 2.2.8 Kiểm tra và xử lý lỗi

Trong quá trình phát triển và thử nghiệm ứng dụng, em đã tiến hành kiểm tra và xử lý các lỗi thường gặp nhằm đảm bảo tính ổn định và độ tin cậy của hệ thống. Dưới đây là một số tình huống lỗi tiêu biểu cùng với phương án khắc phục đã thực hiện.

Các lỗi thường gặp:

- Lỗi mất kết nối mạng:
  - Khi thiết bị không có kết nối Internet, ứng dụng không thể truy xuất dữ liệu thời tiết từ API. Giao diện lúc này chỉ hiển thị màn hình nền và vòng tải (loading) nhưng không có dữ liệu hiển thị.



Hình 2-19 Lỗi không có kết nối mạng

- Lỗi phản hồi từ API:
  - Trong một số trường hợp, API trả về lỗi (ví dụ: sai tên thành phố, hết hạn API key,...). Nếu không xử lý đúng cách, ứng dụng có thể crash hoặc không phản hồi.
- Từ chối quyền truy cập vị trí (GPS):
  - Nếu người dùng không cấp quyền truy cập vị trí khi được yêu cầu, ứng dụng sẽ không thể lấy tọa độ để truy vấn thời tiết theo vị trí hiện tại.
- Giải pháp xử lý:
  - Đối với lỗi mất kết nối mạng:
    - Kiểm tra trạng thái kết nối Internet trước khi gọi API (Connectivity package hoặc try-catch với timeout).
    - Hiển thị thông báo lỗi rõ ràng cho người dùng, ví dụ: “Không có kết nối Internet. Vui lòng kiểm tra mạng và thử lại.”
  - Đối với lỗi từ API:
    - Kiểm tra mã trạng thái phản hồi HTTP.
    - Bắt và hiển thị lỗi qua Snackbar hoặc popup.
  - Đối với lỗi từ chối GPS:
    - Kiểm tra quyền truy cập bằng thư viện geolocator.
    - Nếu bị từ chối, hiển thị hướng dẫn cấp quyền hoặc đưa ra tùy chọn nhập thành phố thủ công thay thế.

## **2.3 Kết chương**

Chương 2 đã mô tả chi tiết quá trình thực hiện đề tài, từ việc thiết kế giao diện, tích hợp API thời tiết, xử lý dữ liệu JSON, đến việc phát triển các chức năng chính như tìm kiếm, định vị vị trí, chuyển đổi nhiệt độ, và hiển thị dự báo 7 ngày. Các bước kiểm thử và xử lý lỗi cũng được thực hiện nhằm đảm bảo tính ổn định của ứng dụng.

Chương tiếp theo sẽ tập trung vào việc đánh giá kết quả thực tập, mức độ áp dụng kiến thức đã học và định hướng phát triển đề tài trong tương lai.

## CHƯƠNG 3 ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

### 3.1 Đánh giá kết quả thực tập

#### 3.1.1 Về kiến thức

- Trong quá trình thực tập, em đã có cơ hội áp dụng các kiến thức đã học tại trường vào một dự án thực tế. Em nhận thấy rằng:
  - Khỏi kiến thức chuyên ngành như lập trình Flutter, xử lý API, thao tác với JSON, lập trình hướng đối tượng (OOP) được vận dụng rất nhiều.
  - Kiến thức nền tảng về UI/UX, các mô hình thiết kế giao diện và tư duy cấu trúc code giúp em làm việc logic, dễ mở rộng và dễ bảo trì hơn.
  - Các khái niệm về ánh xạ dữ liệu, xử lý ngoại lệ (exception handling) và quản lý trạng thái (state management cơ bản) được hiểu sâu sắc hơn khi triển khai thật.

#### 3.1.2 Về kỹ năng

##### A. Kỹ năng lập trình

- Em được thực hành trực tiếp những kiến thức đã được học trên lớp thông qua việc triển khai các chức năng cụ thể, như:
  - Tìm kiếm thành phố bằng flutter\_typeahead
  - Định vị GPS sử dụng geolocator
  - Chuyển đổi đơn vị nhiệt độ ( $^{\circ}\text{C} \leftrightarrow ^{\circ}\text{F}$ )
  - Xử lý dữ liệu JSON trả về từ API thời tiết
- Các kỹ năng viết code theo mô hình hướng đối tượng (OOP), quản lý trạng thái đơn giản (setState), và phân chia logic theo lớp (WeatherService) được vận dụng rõ ràng, bài bản.
- Việc được tiếp xúc với logic thực tế giúp em viết code có cấu trúc hơn, dễ bảo trì và hiểu rõ tầm quan trọng của viết code “sạch”.

##### B. Kỹ năng sử dụng công cụ

- Trong quá trình làm việc, em đã thành thạo hơn với các công cụ phát triển phổ biến, cụ thể:
  - Android Studio: sử dụng để code, debug và chạy ứng dụng.
  - Postman: để test API độc lập, kiểm tra kết quả JSON trả về.
  - Git & GitHub: để quản lý phiên bản mã nguồn, đẩy code lên repo riêng và làm quen với quy trình commit – push – pull.
  - Quản lý thư viện qua pubspec.yaml: hiểu cách thêm thư viện, xử lý dependency, và dùng flutter pub get đúng cách.



### **C. Kỹ năng giao tiếp**

- Em biết cách trình bày ý tưởng rõ ràng hơn khi thảo luận với người hướng dẫn hoặc báo cáo tiến độ.
- Học được cách tiếp nhận góp ý tích cực, sửa lỗi theo hướng dẫn và phản hồi nhanh khi gặp vấn đề.
- Giao tiếp qua bình luận trong mã nguồn, báo cáo lỗi rõ ràng giúp làm việc nhóm hiệu quả hơn.

### **D. Kỹ năng làm việc độc lập**

- Em chủ động nghiên cứu thêm ngoài chương trình học, đặc biệt là những thư viện chưa từng học chính thức trên lớp như:
  - geolocator để lấy vị trí
  - http để gửi request
  - flutter\_typeahead để làm ô tìm kiếm gợi ý
- Khi gặp khó khăn, em có thể tự tìm kiếm tài liệu trên trang chính thức hoặc StackOverflow, học cách đọc tài liệu kỹ thuật tiếng Anh.

### **E. Kỹ năng giải quyết vấn đề**

- Khi gặp lỗi trong quá trình lập trình (ví dụ: sai định dạng dữ liệu JSON, lỗi gọi sai endpoint, null data), em học được cách:
  - Debug có hệ thống, kiểm tra từng bước thay vì đoán mò.
  - Đặt câu hỏi rõ ràng để dễ xác định lỗi.
  - Phân tích nguyên nhân gốc và chọn phương án xử lý tối ưu, ngắn gọn, dễ mở rộng về sau.

### **F. Kỹ năng làm việc nhóm**

- Em nhận thấy rằng làm việc nhóm đóng vai trò quan trọng trong phát triển phần mềm. Khi có sự phối hợp nhịp nhàng giữa các thành viên, tiến độ công việc được đảm bảo, tinh thần làm việc được cải thiện.
- Em học cách trao đổi đúng trọng tâm, chia sẻ tiến độ, hỗ trợ lẫn nhau khi gặp lỗi, đồng thời biết cách lắng nghe và tôn trọng quan điểm của người khác.
- Qua đó, em nhận ra tầm quan trọng của trách nhiệm cá nhân trong một tập thể và kỹ năng phối hợp hiệu quả trong nhóm để hướng đến một sản phẩm chung chất lượng.

#### ***3.1.3 Đánh giá mức độ áp dụng kiến thức đã học tại trường với thực tế công việc trong quá trình thực tập.***

Những kiến thức (học phần) được áp dụng:

- Lập trình di động (Flutter): là học phần áp dụng trực tiếp xuyên suốt thực tập.

- Lập trình hướng đối tượng: được sử dụng để thiết kế các class như WeatherService, tổ chức code rõ ràng.
- Cơ sở dữ liệu và API: giúp hiểu được cách làm việc với dữ liệu JSON, RESTful API.
- Giao diện người dùng: từ học phần Thiết kế Web và Ứng dụng, áp dụng vào Flutter UI.
- Kỹ thuật phần mềm: hỗ trợ trong việc viết code có cấu trúc và logic.

Kiến thức mới được tiếp cận:

- Tích hợp vị trí GPS bằng Geolocator
- Làm việc với flutter\_typeahead để tạo trải nghiệm tìm kiếm mượt mà
- Chuyển đổi đơn vị nhiệt độ theo ngữ cảnh người dùng
- Quản lý chế độ sáng – tối (Dark/Light Mode) trong ứng dụng

Các kiến thức mới này có liên hệ mật thiết với các học phần đã học, nhưng mở rộng về cách áp dụng thực tế và tư duy xử lý linh hoạt hơn.

#### **3.1.4 Tự đánh giá khả năng đáp ứng công việc thực tế**

Qua quá trình thực tập, em nhận thấy bản thân đã có thể:

- Tự đảm nhận việc phân tích yêu cầu, chia module và triển khai các tính năng nhỏ đến trung bình trong một ứng dụng.
- Hiểu được quy trình làm việc thực tế, phối hợp giữa giao diện – logic – dữ liệu.
- Sẵn sàng học thêm kiến thức mới ngoài chương trình học, linh hoạt khi gặp công nghệ hoặc thư viện chưa từng tiếp cận.

Tuy nhiên, em cũng nhận thấy mình cần rèn luyện thêm:

- Kỹ năng tối ưu code, cấu trúc ứng dụng lớn
- Kỹ năng làm việc nhóm trong môi trường chuyên nghiệp
- Tư duy quản lý trạng thái (state management) ở mức cao hơn (như Provider, Bloc,...)

Em tin rằng, nếu tiếp tục học hỏi và thực hành, em có thể đáp ứng tốt các yêu cầu công việc lập trình viên ứng dụng di động trong tương lai gần.

### **3.2 Định hướng phát triển của đề tài**

### **3.3 Kết chương**

Chương 3 đã tổng hợp quá trình tự đánh giá kết quả thực tập, bao gồm kiến thức chuyên môn, kỹ năng mềm và khả năng thích ứng với môi trường làm việc thực tế.

## KẾT LUẬN

Qua quá trình thực tập tại Công ty Cổ phần NetNam và thực hiện đề tài "Xây dựng ứng dụng dự báo thời tiết thời gian thực với WeatherAPI", em đã có cơ hội tiếp cận và áp dụng những kiến thức lý thuyết vào thực tiễn, đặc biệt trong lĩnh vực phát triển ứng dụng di động sử dụng Flutter.

Trong suốt thời gian thực tập, em đã từng bước:

- Tìm hiểu về Flutter framework, ngôn ngữ Dart và kiến trúc ứng dụng đa màn hình.
- Tích hợp thành công WeatherAPI để lấy dữ liệu thời tiết theo thời gian thực.
- Xây dựng các chức năng thiết yếu như: hiển thị thời tiết hiện tại, dự báo 7 ngày, tìm kiếm thành phố, tích hợp GPS, chuyển đổi đơn vị nhiệt độ ( $^{\circ}\text{C} \leftrightarrow ^{\circ}\text{F}$ ), và hỗ trợ chế độ giao diện Dark Mode.
- Tối ưu giao diện người dùng, đảm bảo tính thân thiện, rõ ràng, dễ sử dụng.
- Xử lý các lỗi phổ biến và đảm bảo ứng dụng hoạt động ổn định trên nhiều thiết bị.

Đề tài không chỉ giúp em nâng cao đáng kể kỹ năng lập trình ứng dụng di động bằng Flutter, mà còn mang lại cho em cái nhìn rõ ràng và thực tế hơn về toàn bộ quy trình phát triển phần mềm – từ phân tích yêu cầu, thiết kế, viết mã, đến tích hợp dữ liệu từ API và kiểm thử sản phẩm. Em cũng đã rèn luyện được khả năng đọc hiểu tài liệu kỹ thuật, sử dụng các công cụ hỗ trợ phát triển như Git, cũng như cải thiện kỹ năng tự học và giải quyết vấn đề độc lập.

Thông qua việc xây dựng ứng dụng dự báo thời tiết sử dụng WeatherAPI, em đã tiếp cận và hoàn thành nhiều kỹ thuật quan trọng như: gọi và xử lý dữ liệu từ API, hiển thị thời tiết hiện tại và theo giờ, tìm kiếm vị trí, tích hợp GPS, chuyển đổi đơn vị nhiệt độ, và hỗ trợ chế độ giao diện sáng – tối (Light/Dark Mode). Ứng dụng đã đáp ứng được các chức năng theo đúng đề cương thực tập đề ra.

Đây là một trải nghiệm rất có ý nghĩa với em, giúp em trưởng thành hơn cả về chuyên môn lẫn thái độ làm việc chuyên nghiệp.

Em xin chân thành cảm ơn Công ty Cổ phần NetNam đã tạo điều kiện để em có cơ hội thực tập trong môi trường chuyên nghiệp, năng động và đầy tính thực tiễn. Trong suốt thời gian thực tập, em đã nhận được rất nhiều sự giúp đỡ quý báu từ các anh/chị trong công ty.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến Tiến sĩ Nguyễn Thị Hữu Phương – giảng viên hướng dẫn, người đã luôn đồng hành, chỉ dẫn tận tình và hỗ trợ em trong việc định hướng đề tài, đánh giá kết quả, cũng như giải quyết các vướng mắc kỹ thuật trong suốt quá trình thực hiện.

Em cũng xin cảm ơn anh Đỗ Văn Ánh – TeamLead bộ phận phát triển giải pháp tại công ty, người trực tiếp hướng dẫn em trong công việc hàng ngày, chia sẻ những kinh nghiệm thực

tế và hỗ trợ chuyên môn sát sao để em có thể hoàn thiện sản phẩm đúng tiến độ và đạt được chất lượng tốt.

Em rất trân trọng những gì đã học hỏi được từ đợt thực tập này, và sẽ tiếp tục cố gắng hơn nữa trên con đường học tập và phát triển nghề nghiệp trong tương lai.

## **TÀI LIỆU THAM KHẢO**

- [1] NetNam, “NetNam,” NetNam, [Online]. Available: <https://netnam.com/>. [Acesso em 25 05 2025].
- [2] “Lập trình ứng dụng di động – Viết ứng dụng IOS/Android,” MONA.Media, 05 04 2020. [Online]. Available: <https://surl.li/hmlvup>. [Acesso em 17 04 2025].
- [3] H. Đàm, “Lập trình ứng dụng di động – Xu hướng nghề nghiệp tương lai,” Teky, 25 10 2023. [Online]. Available: [https://teky.edu.vn/blog/lap-trinh-ung-dung-di-dong/#Lap\\_trinh\\_ung\\_dung\\_di\\_dong\\_la\\_gi](https://teky.edu.vn/blog/lap-trinh-ung-dung-di-dong/#Lap_trinh_ung_dung_di_dong_la_gi). [Acesso em 25 05 2025].
- [4] “Dart là gì? Cú pháp, Đặc điểm, Ứng dụng thực tế của Dart,” ITViec, 25 03 2025. [Online]. Available: <https://itviec.com/blog/dart-la-gi/>. [Acesso em 18 04 2025].
- [5] “Flutter documentation,” [Online]. Available: <https://flutter.dev/docs>. [Acesso em 17 4 2025].
- [6] “API (Giao diện lập trình ứng dụng) là gì?,” aws, [Online]. Available: <https://aws.amazon.com/vi/what-is/api/>. [Acesso em 17 04 2025].
- [7] H. nội, “API là gì? Những loại API phổ biến và đặc điểm nổi bật,” 200Lab, 13 06 2024. [Online]. Available: <https://200lab.io/blog/api-la-gi>. [Acesso em 25 05 2025].
- [8] “GPS là gì? Ứng dụng của định vị toàn cầu GPS trên ô tô,” TOYOTA, 19 05 2023. [Online]. Available: <https://www.toyota.com.vn/tin-tuc/thong-tin-bo-tro/gps-la-gi-35592>. [Acesso em 18 04 2025].
- [9] A. Tuner, Introduction to Neogeography, Sebastopol, CA USA: O'Reilly Media, 2006.
- [10] “Flutter là gì?,” aws, [Online]. Available: <https://aws.amazon.com/vi/what-is/flutter/>. [Acesso em 17 04 2025].