

BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
THÀNH PHỐ HỒ CHÍ MINH



LẬP TRÌNH MẠNG

Báo cáo cuối kỳ:

XÂY DỰNG HỆ THỐNG TRA CỨU
KẾT QUẢ HỌC TẬP VÀ PHÂN TÍCH GPA SINH VIÊN

Giảng viên hướng dẫn: TS. Đặng Nhân Cách

Sinh viên thực hiện: Nhóm 02

Nguyễn Văn Công	- 22H1120063
Nguyễn Văn Hoàng Giang	- 2251120206
Trà Ngọc Hiền	- 22H1120069
Phan Thành Phước	- 22H1120077
Nguyễn Quang Thọ	- 22H1120082
Nguyễn Trung Thông	- 2251120056
Nguyễn Huy Toàn	- 22H1120086

Thành phố Hồ Chí Minh, tháng 11 năm 2024

BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
THÀNH PHỐ HỒ CHÍ MINH



LẬP TRÌNH MẠNG

Bài tập cuối kỳ:

XÂY DỰNG HỆ THỐNG TRA CỨU
KẾT QUẢ HỌC TẬP VÀ PHÂN TÍCH GPA SINH VIÊN

Giảng viên hướng dẫn: TS. Đặng Nhân Cách

Sinh viên thực hiện: Nhóm 02

Nguyễn Văn Công	- 22H1120063
Nguyễn Văn Hoàng Giang	- 2251120206
Trà Ngọc Hiền	- 22H1120069
Phan Thành Phước	- 22H1120077
Nguyễn Quang Thọ	- 22H1120082
Nguyễn Trung Thông	- 2251120056
Nguyễn Huy Toàn	- 22H1120086

Thành phố Hồ Chí Minh, tháng 11 năm 2024

DANH SÁCH THÀNH VIÊN

Họ và tên	MSSV	Nội dung thực hiện	Đánh giá
Nguyễn Văn Công	22H1120063	Thực hiện code Android, viết báo cáo Word	100%
Nguyễn Văn Hoàng Giang	2251120206	Thực hiện code IOS, viết báo cáo Word	70%
Trà Ngọc Hiến	22H1120069	Thực hiện code Android, viết báo cáo Word	100%
Phan Thành Phước	22H1120077	Thực hiện code Website, slide Powerpoint	100%
Nguyễn Quang Thọ (Nhóm trưởng)	22H1120082	Thực hiện code Website, viết báo cáo Word, slide Powerpoint	100%
Nguyễn Trung Thông	2251120056	Thực hiện code IOS, viết báo cáo Word	70%
Nguyễn Huy Toàn	22H1120086	Thực hiện code Website, viết báo cáo Word, thuyết trình	100%

Ghi chú: Các thành viên có đánh giá 70% với lý do, không tham gia hoặc không hoàn các nhiệm vụ được giao.

MỤC LỤC

CHƯƠNG 1	GIỚI THIỆU ĐỀ TÀI	1
1.1	Tổng quan đề tài	1
1.1.1	Lý do chọn đề tài	1
1.1.2	Mục tiêu đề tài	1
1.2	Cấu trúc hệ thống	2
1.3	Đối tượng sử dụng	2
CHƯƠNG 2	CƠ SỞ LÝ THUYẾT	4
2.1	Mạng máy tính	4
2.1.1	Khái niệm mạng máy tính	4
2.1.2	Mô hình mạng máy tính	4
2.2	Lập trình mạng	4
2.2.1	Khái niệm lập trình mạng	4
2.2.2	Mục đích của lập trình mạng	5
2.2.3	Ứng dụng của lập trình mạng	5
2.3	Một số mô hình mạng	6
2.3.1	Mô hình OSI	6
2.3.2	Mô hình TCP/IP	7
2.4	Mô hình ứng dụng mạng	9
2.4.1	Mô hình Client-Server	9
2.4.2	Mô hình Peer to Peer	9
2.5	Một số giao thức mạng	10
2.5.1	Giao thức TCP	10
2.5.2	Giao thức UDP	11
2.5.3	Giao thức SMTP	11
2.5.4	Giao thức POP3	12
2.5.5	Giao thức FTP	13
2.6	Sơ lược về Socket	13
2.6.1	Khái niệm về Socket	13
2.6.2	Chức năng của Socket	14
CHƯƠNG 3	PHÂN TÍCH YÊU CẦU HỆ THỐNG	15

3.1	Mô hình hệ thống	15
3.1.1	Mô hình Client-Server.....	15
3.1.2	Nguyên tắc hoạt động.....	16
3.1.3	Sơ đồ User Case	17
3.2	Cơ sở dữ liệu	18
3.2.1	Bảng Subject	18
3.2.2	Bảng Class.....	19
3.2.3	Bảng Student	19
3.2.4	Bảng StudentGrades.....	19
3.3	Ngôn ngữ lập trình và công nghệ sử dụng.....	20
CHƯƠNG 4 XÂY DỰNG CHƯƠNG TRÌNH.....		23
4.1	Server	23
4.1.1	Xây dựng API RESTful	23
4.1.2	API phân tích GPA sinh viên.....	23
4.1.3	Cấu hình và khởi tạo Server.....	24
4.2	Client.....	25
4.2.1	Website	25
4.2.1.1	Trang tra cứu kết quả học tập sinh viên.....	25
4.2.1.2	Trang phân tích GPA.....	26
4.2.2	Android.....	27
4.2.2.1	Trang tra cứu kết quả học tập	27
4.2.2.2	Trang phân tích GPA.....	28
CHƯƠNG 5 KẾT QUẢ THỰC THI		29
5.1	Server	29
5.2	Website	30
5.3	Android	33
CHƯƠNG 6 KẾT LUẬN.....		35
6.1	Kết luận	35
6.2	Đề xuất	35

MỤC LỤC HÌNH ẢNH

Hình 2.1 Mô hình lập trình mạng	5
Hình 2.2 Mô hình OSI	6
Hình 2.3 Mô hình TCP/IP.....	8
Hình 2.4 Mô hình Client-Server	9
Hình 2.5 Mô hình Peer to Peer (P2P)	10
Hình 2.6 Giao thức TCP	10
Hình 2.7 Giao thức UDP	11
Hình 2.8 Giao thức SMTP	12
Hình 2.9 Giao thức POP3	13
Hình 2.10 Giao thức FTP	13
Hình 2.11 Mô hình Socket.....	14
Hình 3.1 Mô hình hệ thống Client-Server	16
Hình 3.2 Nguyên tắc hoạt động.....	17
Hình 3.3 Sơ đồ User Case	18
Hình 3.4 Các ngôn ngữ lập trình Backend	21
Hình 3.5 Các ngôn ngữ lập trình Frontend.....	21
Hình 3.6 Các ngôn ngữ và công cụ lập trình Android.....	22
Hình 5.1 Bắt đầu khởi chạy Server.....	29
Hình 5.2 Server trả request tra cứu điểm học tập	29
Hình 5.3 Server trả request phân tích GPA	30
Hình 5.4 Giao diện website khi chưa thực hiện tra cứu điểm	30
Hình 5.5 Giao diện website khi thực hiện tra cứu điểm.....	31
Hình 5.6 Giao diện website khi chưa thực hiện phân tích GPA	31
Hình 5.7 Giao diện website khi thực hiện phân tích GPA.....	32
Hình 5.8 Giao diện android khi tra cứu điểm.....	33
Hình 5.9 Giao diện app android khi phân tích GPA	34

MỤC LỤC BẢNG

Bảng 3.1 Bảng cơ sở dữ liệu Subject	18
Bảng 3.2 Bảng cơ sở dữ liệu Class.....	19
Bảng 3.3 Bảng cơ sở dữ liệu Student	19
Bảng 3.4 Bảng cơ sở dữ liệu StudentGrades	20

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời tri ân sâu sắc nhất đến Thầy – Tiến sĩ Đặng Nhân Cách, người đã tận tình hướng dẫn và giảng dạy chúng em trong suốt học phần Lập trình mạng. Qua những bài giảng đầy tâm huyết, Thầy không chỉ giúp chúng em tiếp thu những kiến thức chuyên môn về lập trình mạng mà còn truyền cảm hứng để chúng em hiểu rõ hơn về vai trò và tầm quan trọng của mạng máy tính trong thời đại công nghệ số hiện nay.

Nhờ sự chỉ dẫn tận tình của Thầy, nhóm chúng em đã có nền tảng để nghiên cứu, triển khai và hoàn thiện dự án này. Trong suốt quá trình thực hiện, chúng em đã cố gắng vận dụng tối đa kiến thức và kỹ năng lập trình mạng để tạo nên một sản phẩm chất lượng. Tuy nhiên, với những hạn chế về kinh nghiệm thực tế, bài làm của nhóm không thể tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý từ Thầy để có thể học hỏi thêm, từ đó rút kinh nghiệm và nâng cao năng lực của bản thân.

Chúng em cũng muốn bày tỏ lòng biết ơn sâu sắc về những lời khuyên, sự động viên và nhiệt huyết mà Thầy đã dành cho chúng em trong suốt khóa học. Những điều này không chỉ giúp chúng em hoàn thiện dự án mà còn mang lại những bài học quý báu cho hành trình phát triển trong tương lai.

Một lần nữa, nhóm xin chân thành gửi lời cảm ơn thầy Đặng Nhân Cách. Kính chúc Thầy thật nhiều sức khỏe, luôn giữ vững niềm đam mê trong sự nghiệp giảng dạy, và đạt được nhiều thành công trên hành trình gieo mầm tri thức.

- Nhóm 02 -

CHƯƠNG 1 GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan đề tài

1.1.1 Lý do chọn đề tài

Trong thời đại công nghệ số hiện nay, việc ứng dụng công nghệ thông tin vào lĩnh vực giáo dục không chỉ là xu thế tất yếu mà còn là một yêu cầu cấp thiết. Các trường đại học và tổ chức giáo dục ngày càng chú trọng triển khai các hệ thống quản lý thông tin học tập, hỗ trợ sinh viên theo dõi tiến trình học tập một cách khoa học, đồng thời đưa ra các quyết định chính xác và kịp thời để cải thiện kết quả học tập.

Tuy nhiên, nhiều hệ thống hiện tại vẫn còn hạn chế, chưa đáp ứng đầy đủ nhu cầu tra cứu và phân tích dữ liệu học tập của sinh viên. Do đó, việc xây dựng và phát triển một hệ thống hỗ trợ sinh viên tra cứu kết quả học tập theo từng kỳ là vô cùng cần thiết. Hệ thống này không chỉ giúp sinh viên dễ dàng theo dõi kết quả học tập mà còn cung cấp công cụ phân tích chỉ số GPA.

Việc tích hợp các tính năng này sẽ mang lại nhiều lợi ích thực tiễn, giúp sinh viên không chỉ có cái nhìn tổng quan về tiến trình học tập mà còn nâng cao năng lực tự quản lý và tối ưu hóa kết quả học tập.

Với mong muốn giải quyết các vấn đề trên và đóng góp vào việc cải thiện công cụ hỗ trợ học tập cho sinh viên, nhóm nghiên cứu đã quyết định lựa chọn đề tài: **“Xây dựng hệ thống tra cứu kết quả học tập và phân tích GPA sinh viên”**.

1.1.2 Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một hệ thống Client-Server, trong đó:

- **Server:** Được triển khai độc lập, chịu trách nhiệm quản lý, xử lý yêu cầu và cung cấp dữ liệu.
- **Client:** Bao gồm một website và một ứng dụng Android hoạt động độc lập, gửi yêu cầu đến server để tra cứu và phân tích dữ liệu học tập.

Hệ thống sẽ đáp ứng các chức năng chính như sau:

- **Tra cứu kết quả học tập:** Cho phép sinh viên dễ dàng truy cập và tra cứu điểm học tập theo từng kỳ, hỗ trợ theo dõi tiến trình học tập một cách thuận tiện.

- **Phân tích chỉ số GPA:** Cung cấp công cụ cho phép sinh viên nhập dữ liệu vào, dựa trên dữ liệu này, hệ thống sẽ phân tích và đề xuất lộ trình học tập, giúp sinh viên điều chỉnh kế hoạch để đạt được mục tiêu GPA mong muốn.

Mục tiêu cốt lõi của đề tài là phát triển một hệ thống hiện đại, dễ sử dụng và có tính ứng dụng cao, hỗ trợ sinh viên theo dõi, quản lý tiến trình học tập một cách hiệu quả.

1.2 Cấu trúc hệ thống

Hệ thống được chia thành ba phần chính, bao gồm:

- **Web Server:** Là thành phần chịu trách nhiệm lưu trữ và quản lý cơ sở dữ liệu của hệ thống, xử lý các yêu cầu từ người dùng và gửi lại kết quả dưới dạng JSON. Web Server sẽ thực hiện các tác vụ liên quan đến việc tính toán GPA, tra cứu điểm, và các yêu cầu khác của sinh viên.
- **Website:** Cung cấp giao diện trực quan cho người dùng (sinh viên), giúp họ dễ dàng tra cứu kết quả học tập theo từng kỳ. Bên cạnh đó, website cũng cung cấp công cụ phân tích GPA, cho phép sinh viên ước tính kết quả học tập trong tương lai dựa trên các dữ liệu đã có. Website sẽ kết nối với Web Server thông qua các API để truy xuất dữ liệu và hiển thị thông tin cho người dùng.
- **Ứng dụng Android:** Đây là ứng dụng di động cho phép sinh viên có thể tra cứu kết quả học tập và phân tích GPA ngay trên điện thoại của mình. Ứng dụng sẽ kết nối với Web Server để lấy dữ liệu và thực hiện các phép tính phân tích GPA một cách nhanh chóng và thuận tiện. Các tính năng của ứng dụng bao gồm tra cứu điểm học tập theo từng kỳ, tính toán và phân tích GPA, cùng với các tính năng hữu ích khác phục vụ cho việc học tập.

Với cấu trúc ba thành phần này, hệ thống đảm bảo cung cấp các dịch vụ học tập hiệu quả và linh hoạt. Sinh viên có thể dễ dàng truy cập và sử dụng, từ việc tra cứu kết quả đến việc phân tích và dự đoán chỉ số GPA, qua đó hỗ trợ họ quản lý và tối ưu hóa lộ trình học tập của mình.

1.3 Đối tượng sử dụng

Hệ thống được thiết kế đặc biệt để hỗ trợ sinh viên trong việc quản lý và theo dõi kết quả học tập một cách tiện lợi và hiệu quả. Các chức năng chính mà hệ thống cung cấp cho sinh viên bao gồm:

- **Tra cứu kết quả học tập:** Sinh viên có thể sử dụng hệ thống để tra cứu kết quả học tập của mình theo từng kỳ học. Điều này giúp họ dễ dàng theo dõi và đánh giá tiến trình học tập, từ đó có những điều chỉnh kịp thời cho kế hoạch học tập của mình.
- **Phân tích GPA:** Sinh viên có thể nhập các thông tin cá nhân liên quan đến học tập như: GPA hiện tại, GPA mong muốn, số tín chỉ đã hoàn thành, số tín chỉ cần hoàn thành, và số môn học hệ hai tín chỉ hoặc ba tín chỉ. Dựa trên dữ liệu này, hệ thống sẽ thực hiện phân tích và đưa ra dự đoán về số điểm cần đạt ở các môn hai tín chỉ và ba tín chỉ để đạt được GPA mục tiêu.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Mạng máy tính

2.1.1 Khái niệm mạng máy tính

Mạng máy tính là một hệ thống kết nối các thiết bị máy tính lại với nhau thông qua các phương tiện truyền dẫn có dây hoặc không dây. Sự liên kết này cho phép các máy tính trong mạng chia sẻ thông tin, dữ liệu, và tài nguyên một cách dễ dàng và hiệu quả.

2.1.2 Mô hình mạng máy tính

Hiện nay, có tổng cộng 4 mô hình mạng máy tính được sử dụng rộng rãi:

- Mạng LAN (Local Area Network): Đây là hệ thống mạng cục bộ, thường được sử dụng trong các khu vực nhỏ như văn phòng, trường học, hoặc các tòa nhà. Mạng LAN kết nối các thiết bị trong phạm vi hạn chế, giúp chia sẻ tài nguyên và dữ liệu nhanh chóng.
- Mạng MAN (Metropolitan Area Network): Mạng có phạm vi kết nối rộng hơn mạng LAN, thường được sử dụng trong các thành phố hoặc khu vực đô thị. Mạng MAN kết nối các hệ thống mạng LAN trong một khu vực địa lý lớn hơn.
- Mạng WAN (Wide Area Network): Mạng WAN có phạm vi kết nối rộng hơn cả mạng MAN, có thể bao phủ các khu vực lớn như quốc gia hoặc toàn cầu. Mạng WAN kết nối các mạng LAN và MAN từ xa, ví dụ như Internet, giúp truyền tải dữ liệu giữa các địa điểm khác nhau trên thế giới.
- Mạng PAN (Personal Area Network): Đây là mạng dùng để kết nối các thiết bị cá nhân trong phạm vi rất nhỏ, chẳng hạn như kết nối giữa điện thoại, máy tính bảng, laptop, hoặc các thiết bị Bluetooth của một người dùng. Mạng PAN thường có phạm vi kết nối trong khoảng vài mét.

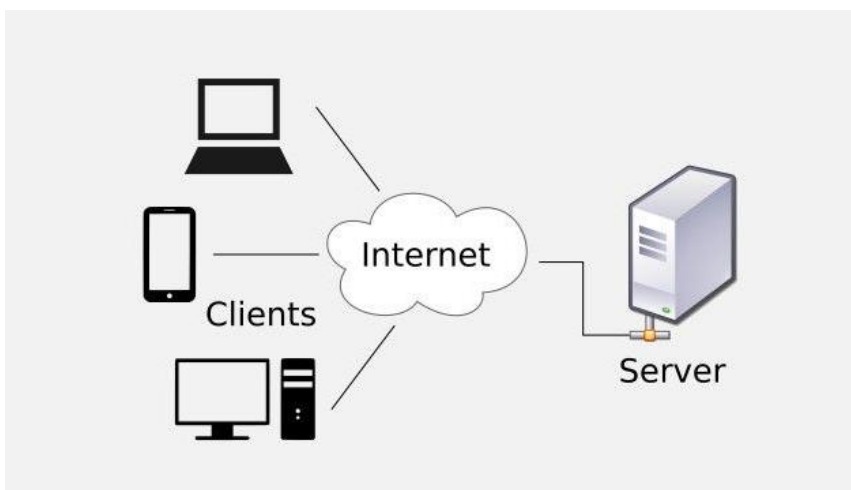
2.2 Lập trình mạng

2.2.1 Khái niệm lập trình mạng

Lập trình mạng có thể được hiểu đơn giản là quá trình xây dựng các ứng dụng thông qua mạng máy tính. Lập trình viên sử dụng các công cụ và phần mềm để phát triển các ứng dụng trong hệ thống của doanh nghiệp hoặc các dịch vụ trực tuyến.

Công thức để xây dựng lập trình mạng như sau:

Lập trình mạng = Kiến thức mạng + Mô hình lập trình mạng + Ngôn ngữ lập trình mạng



Hình 2.1 Mô hình lập trình mạng

2.2.2 Mục đích của lập trình mạng

- Giao tiếp dữ liệu: Lập trình mạng cho phép các ứng dụng trao đổi thông tin qua mạng máy tính, giúp chia sẻ dữ liệu, tệp tin, tin nhắn và các loại thông tin khác một cách dễ dàng.
- Mở rộng khả năng ứng dụng: Lập trình mạng giúp các ứng dụng không chỉ hoạt động trong một hệ thống cục bộ mà còn có thể truy cập và sử dụng trên mạng lớn như LAN hoặc Internet.
- Phát triển dịch vụ mạng: Lập trình mạng là cơ sở để cung cấp dịch vụ trực tuyến như email, trang web, dịch vụ đám mây và các ứng dụng tương tác qua mạng, đáp ứng nhiều nhu cầu trong các lĩnh vực khác nhau.

2.2.3 Ứng dụng của lập trình mạng

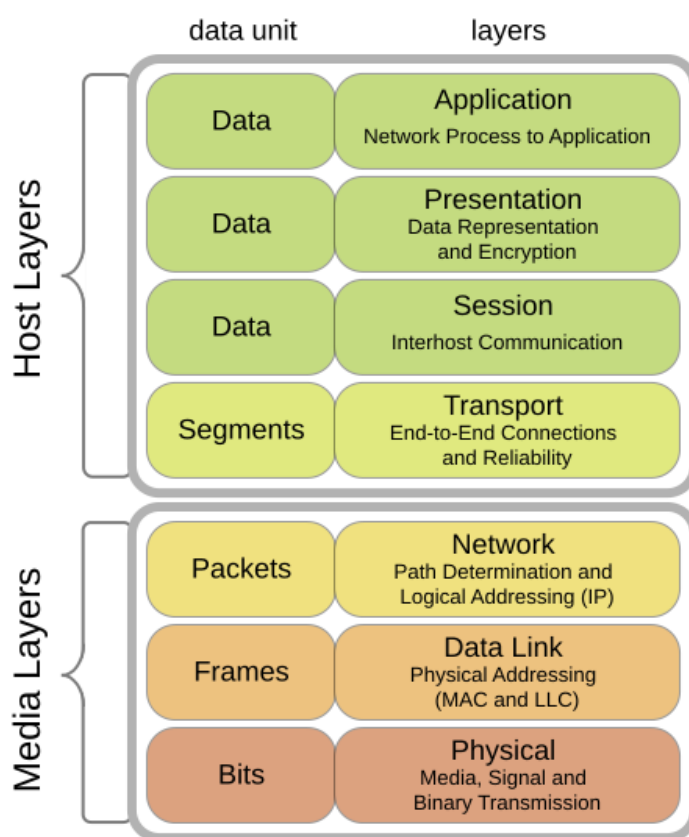
- Dịch vụ web: Tạo các trang web và ứng dụng web cho phép giao tiếp qua Internet.
- Ứng dụng chat và hội nghị truyền hình: Phát triển phần mềm như Skype, Zoom, Messenger để gọi điện, nhắn tin, video call.
- Trò chơi trực tuyến: Hỗ trợ các trò chơi yêu cầu giao tiếp thời gian thực, như trò chơi multiplayer.
- Hệ thống phân phối tệp tin: Phát triển ứng dụng chia sẻ dữ liệu lớn qua mạng, như FTP hoặc các dịch vụ đám mây (Google Drive, Dropbox).

- Internet of Things (IoT): Kết nối các thiết bị thông minh như đèn, tủ lạnh, máy điều hòa để điều khiển từ xa.

2.3 Một số mô hình mạng

2.3.1 Mô hình OSI

Mô hình OSI (Open Systems Interconnection) là một khung tham chiếu chia chức năng truyền thông mạng thành 7 tầng, mỗi lớp có chức năng riêng biệt và giao tiếp với lớp trên và lớp dưới, nhằm thiết kế và lý giải cách kết nối và giao thức mạng giữa các máy tính.



Hình 2.2 Mô hình OSI

Các tầng trong mô hình OSI bao gồm:

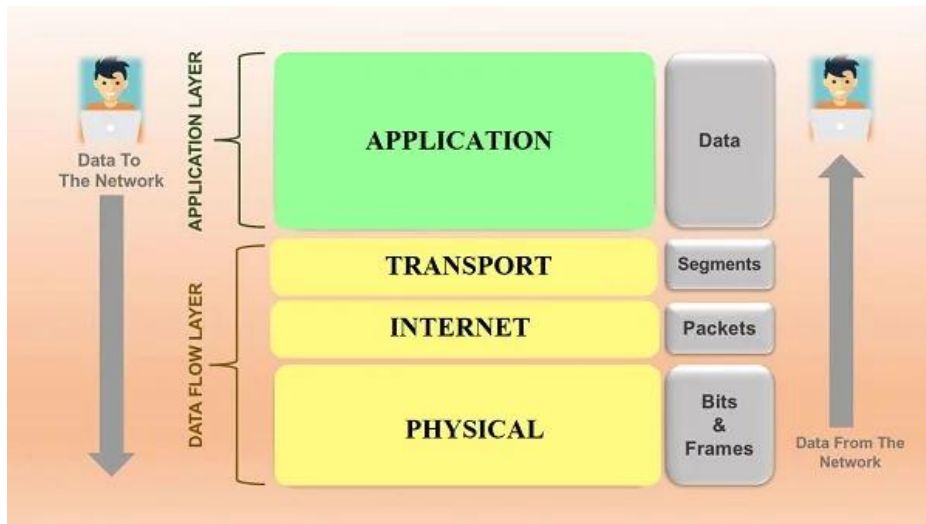
- Tầng 1: Vật lý (Physical Layer): Chịu trách nhiệm truyền tải tín hiệu điện tử, quang học hoặc vô tuyến giữa các thiết bị mạng, quy định các yếu tố như loại cáp, kết nối vật lý, và tín hiệu điện áp.
- Tầng 2: Liên kết dữ liệu (Data Link Layer): Điều khiển truy cập vào phương tiện truyền dẫn, phát hiện và sửa lỗi để đảm bảo dữ liệu được truyền chính xác. Tầng

này bao gồm hai lớp phụ: MAC (Medium Access Control) và LLC (Logical Link Control).

- Tầng 3: Mạng (Network Layer): Định tuyến và xác định đường đi của gói tin từ nguồn đến đích. Giao thức phổ biến: IP (Internet Protocol).
- Tầng 4: Giao vận (Transport Layer): Đảm bảo dữ liệu được truyền đến đúng ứng dụng ở máy nhận. Các giao thức phổ biến: TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).
- Tầng 5: Phiên (Session Layer): Quản lý các phiên kết nối giữa hai thiết bị, bao gồm việc thiết lập, duy trì và kết thúc phiên giao tiếp.
- Tầng 6: Trình bày (Presentation Layer): Chịu trách nhiệm chuyển đổi dữ liệu giữa định dạng mà mạng sử dụng và định dạng mà ứng dụng có thể hiểu. Tầng này bao gồm các công việc như mã hóa, giải mã, nén và giải nén dữ liệu.
- Tầng 7: Ứng dụng (Application Layer): Là tầng gần người dùng nhất, cung cấp giao diện trực tiếp giữa người dùng và mạng. Các giao thức ở tầng này bao gồm HTTP, FTP, SMTP, DNS,...

2.3.2 Mô hình TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) là một bộ giao thức dùng để trao đổi thông tin trong mạng Internet. Nó xác định cách thức đóng gói dữ liệu thành các gói tin, đồng thời quy định cách thức các gói tin này được gửi và nhận giữa các thiết bị kết nối trong mạng. TCP/IP là nền tảng của việc truyền tải dữ liệu qua mạng Internet và đảm bảo rằng thông tin được truyền đi chính xác và hiệu quả.



Hình 2.3 Mô hình TCP/IP

- Tầng truy cập mạng (Link Layer): Cung cấp giao diện tương tác với mạng vật lý, chịu trách nhiệm định dạng dữ liệu cho bộ phận truyền tải trung gian. Tầng này tạo địa chỉ dữ liệu cho các tiểu mạng dựa trên địa chỉ phần cứng vật lý và kiểm tra lỗi trong quá trình truyền tải.
- Tầng Internet (Internet Layer): Cung cấp địa chỉ logic cho dữ liệu để có thể di chuyển qua các tiểu mạng với cấu trúc vật lý khác nhau. Tầng này thực hiện chức năng định tuyến để truyền tải dữ liệu giữa các mạng và kết nối các mạng khác nhau (liên mạng). Nó liên kết địa chỉ vật lý và địa chỉ logic.
- Tầng vận chuyển (Transport Layer): Quản lý luồng dữ liệu, kiểm tra lỗi và xác nhận các dịch vụ cho liên mạng. Tầng này đóng vai trò làm giao diện giữa các ứng dụng mạng, giúp đảm bảo dữ liệu được truyền tải chính xác và hiệu quả.
- Tầng ứng dụng (Application Layer): Cung cấp các ứng dụng để giải quyết các vấn đề mạng, như vận chuyển tệp tin, điều khiển từ xa và các hoạt động trên Internet. Tầng này cũng hỗ trợ Giao diện Lập trình Ứng dụng (API) mạng, cho phép các chương trình trên một hệ điều hành có thể truy cập và giao tiếp qua mạng.

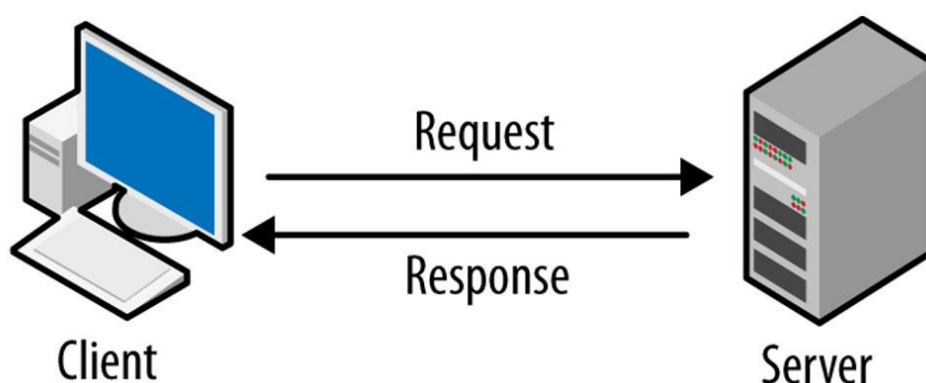
2.4 Mô hình ứng dụng mạng

2.4.1 Mô hình Client-Server

Mô hình Client-Server là một mô hình lập trình mạng phổ biến, trong đó các thiết bị khách hàng (Clients) gửi yêu cầu đến máy chủ (Server) để truy xuất dữ liệu hoặc dịch vụ. Mô hình này chia nhiệm vụ mạng thành hai loại thiết bị chính:

- Client: Là máy tính hoặc thiết bị gửi yêu cầu đến server qua mạng và nhận phản hồi. Client thực hiện các yêu cầu và hiển thị kết quả cho người dùng.
- Server: Là thiết bị lưu trữ tài nguyên và cung cấp các dịch vụ. Server xử lý yêu cầu từ client và gửi phản hồi lại.

Mô hình Client-Server rất phổ biến trong mạng máy tính và các trang web. Một server có thể kết nối với các máy chủ khác để tối ưu hóa hiệu suất, chẳng hạn như gửi yêu cầu đến một máy chủ cơ sở dữ liệu khi không thể tự xử lý.



Hình 2.4 Mô hình Client-Server

2.4.2 Mô hình Peer to Peer

Mô hình Peer-to-Peer (P2P) là mô hình mạng trong đó các thiết bị (hoặc máy tính) trên mạng hoạt động như nhau, không có máy chủ trung tâm. Các thiết bị này có thể chia sẻ tài nguyên trực tiếp với nhau, như dữ liệu, tệp tin, hoặc dịch vụ mà không cần thông qua một máy chủ trung gian.

Mô hình P2P thường được sử dụng trong các ứng dụng chia sẻ tệp tin hoặc các dịch vụ trực tuyến, nơi mỗi thiết bị vừa là máy khách, vừa là máy chủ, có thể tải xuống và chia sẻ dữ liệu đồng thời. Mô hình này giúp tăng cường tính linh hoạt và giảm tải cho các máy chủ trung tâm.



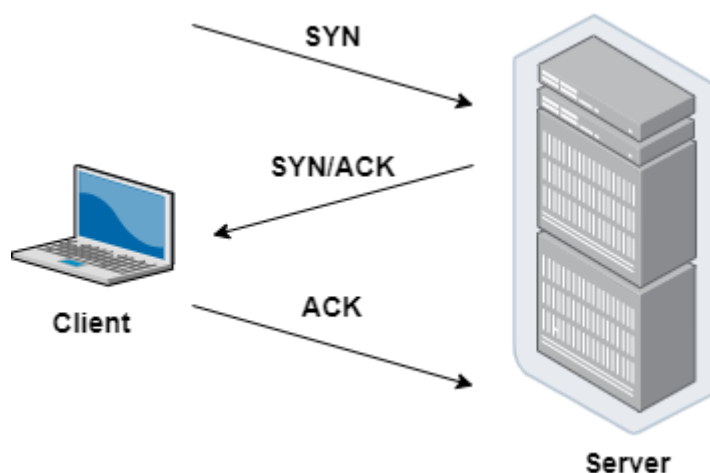
Hình 2.5 Mô hình Peer to Peer (P2P)

2.5 Một số giao thức mạng

2.5.1 Giao thức TCP

TCP (Transmission Control Protocol) là một giao thức quan trọng trong Internet Protocol Suite (gồm TCP/IP). TCP đảm bảo rằng các gói dữ liệu được truyền tải một cách tin cậy và đúng thứ tự giữa các thiết bị qua mạng. Giao thức này có khả năng kiểm soát kết nối, phát hiện lỗi và đảm bảo dữ liệu được truyền đầy đủ mà không thiếu sót.

Khi kết hợp với IP (Internet Protocol), TCP đảm bảo rằng các gói dữ liệu được gửi đúng địa chỉ đích và theo đúng thứ tự, trong khi IP chỉ định cách thức gửi các gói dữ liệu đó. TCP/IP là bộ giao thức cơ bản để trao đổi dữ liệu trên Internet và được sử dụng cho nhiều ứng dụng, bao gồm web, email, FTP, và các ứng dụng ngang hàng.

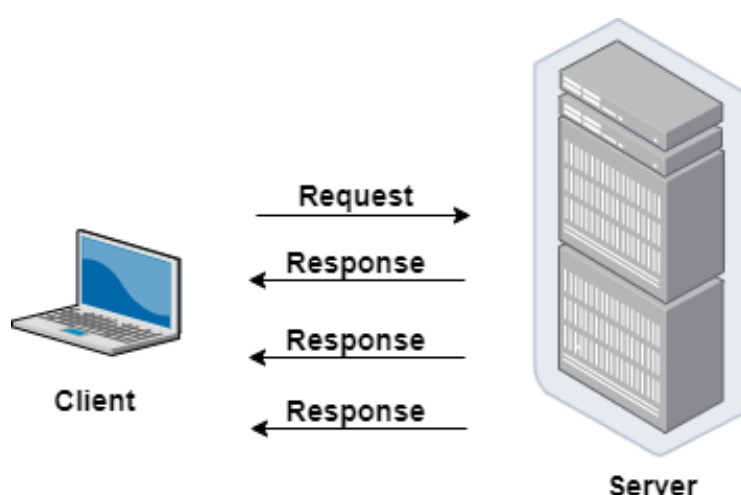


Hình 2.6 Giao thức TCP

2.5.2 Giao thức UDP

UDP (User Datagram Protocol) là một giao thức truyền thông được sử dụng phổ biến trong các ứng dụng yêu cầu tốc độ truyền tải nhanh và nhạy cảm với thời gian, như phát video trực tuyến hoặc tra cứu DNS. UDP không thiết lập kết nối trước khi gửi dữ liệu, điều này giúp giảm độ trễ và tăng tốc độ truyền thông. Tuy nhiên, nhược điểm của UDP là không có cơ chế đảm bảo rằng tất cả các gói dữ liệu sẽ đến đích, dẫn đến việc mất mát dữ liệu trong quá trình truyền tải.

UDP được sử dụng chủ yếu trong các ứng dụng cần truyền tải thời gian thực, như các cuộc gọi video trực tuyến, nơi mà sự mất mát của một vài gói tin có thể chấp nhận được vì các ứng dụng này được thiết kế để xử lý mức độ mất mát dữ liệu nhất định mà không gây ảnh hưởng lớn đến trải nghiệm người dùng. Việc sử dụng UDP giúp giảm độ trễ, mang lại trải nghiệm liên lạc mượt mà hơn so với giao thức TCP, mặc dù có thể dẫn đến rủi ro về an ninh, như các cuộc tấn công DDoS (Distributed Denial of Service).



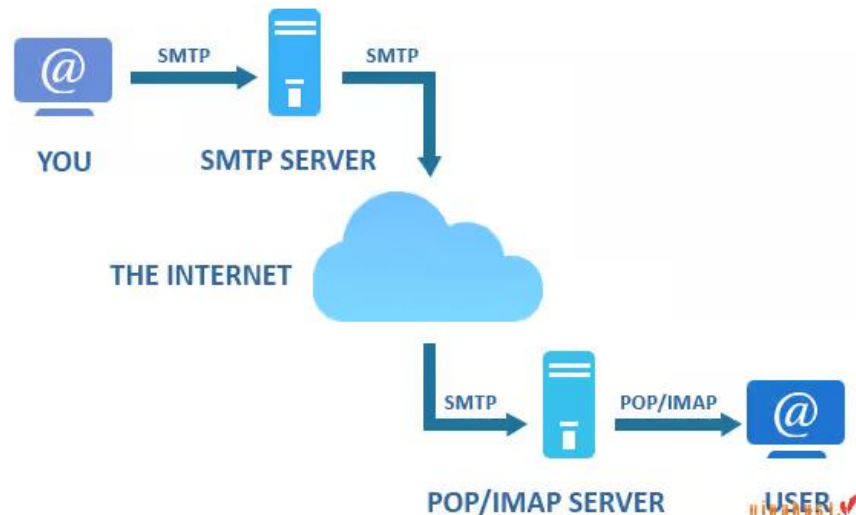
Hình 2.7 Giao thức UDP

2.5.3 Giao thức SMTP

SMTP (Simple Mail Transfer Protocol) là giao thức chuẩn dùng để gửi email qua mạng. Nó quy định cách thức mà các máy tính và máy chủ truyền tải email từ người gửi đến người nhận. SMTP hoạt động giống như một hệ thống phân phối thư, đảm bảo rằng email được chuyển từ máy gửi đến máy chủ thư của nhà cung cấp dịch vụ email.

Tuy nhiên, SMTP không dùng để lấy thư từ máy chủ; thay vào đó, các giao thức khác như POP3 (Post Office Protocol 3) hoặc IMAP (Internet Message Access Protocol) được

sử dụng để truy xuất email từ máy chủ và đưa vào hộp thư người nhận. SMTP chỉ đảm nhận nhiệm vụ chuyển phát email, tương tự như việc chuyển thư từ bưu cục đến hộp thư của người nhận.

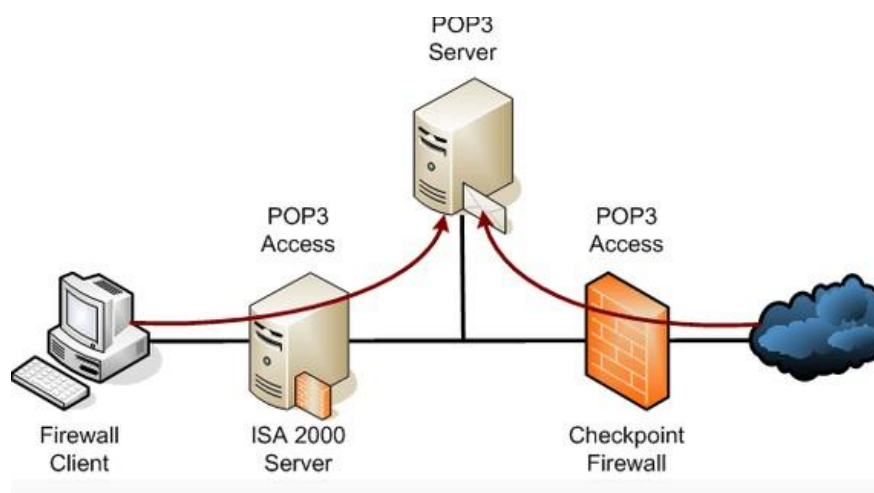


Hình 2.8 Giao thức SMTP

2.5.4 Giao thức POP3

POP3 (Post Office Protocol version 3) là giao thức nhận thư điện tử, cho phép người dùng tải email từ máy chủ mail về máy tính cá nhân thông qua một ứng dụng email client như Outlook, Thunderbird, hoặc Mac Mail. POP3 hoạt động qua kết nối TCP/IP và hỗ trợ người dùng đọc email ngoại tuyến (offline) sau khi đã tải xuống.

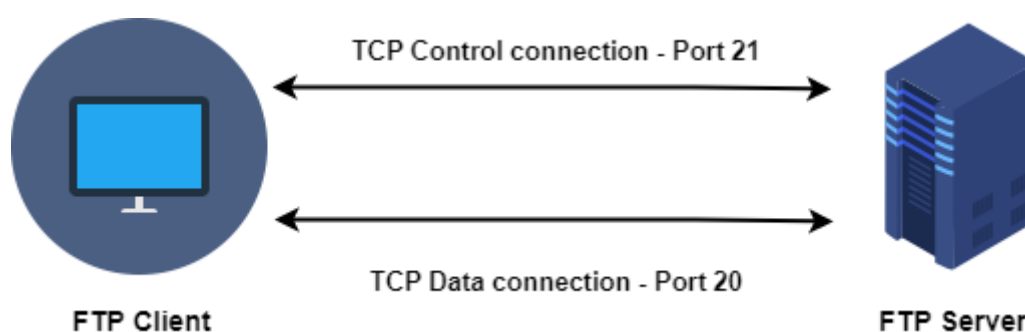
POP là một giao thức nhận mail có lịch sử lâu đời. Nó ra đời từ máy tính còn bị giới hạn bởi tốc độ, băng thông, vậy nên các kỹ sư đã tạo ra POP, một nỗ lực để làm đơn giản nhất có thể để tải các bản copy của email để đọc khi offline, sau đó xóa những email này từ remote server.



Hình 2.9 Giao thức POP3

2.5.5 Giao thức FTP

FTP (File Transfer Protocol) là giao thức dùng để truyền tải tập tin qua mạng TCP/IP, thường hoạt động trên cổng 20 và 21. Giao thức này cho phép các máy client kết nối với máy chủ FTP để gửi hoặc nhận dữ liệu từ xa. FTP được sử dụng rộng rãi trong các hoạt động kinh doanh, giúp trao đổi tập tin giữa doanh nghiệp, đối tác và nhân viên.



Hình 2.10 Giao thức FTP

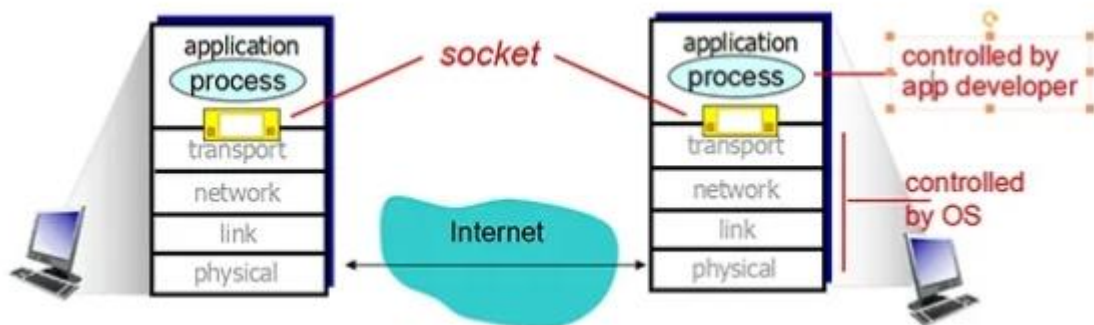
2.6 Sơ lược về Socket

2.6.1 Khái niệm về Socket

Socket là điểm cuối (endpoint) trong kết nối giao tiếp hai chiều, cho phép giao tiếp giữa máy chủ (Server) và khách hàng (Client) thông qua mạng. Mỗi Socket được gắn với một cổng (port) cụ thể, giúp các tầng lớp TCP/IP có thể xác định và xử lý dữ liệu một cách hiệu quả.

Socket đóng vai trò quan trọng trong việc truyền thông giữa các chương trình trên mạng. Nó cho phép các ứng dụng gửi và nhận dữ liệu liên tục, hỗ trợ các giao thức mạng như FTP, POP3, SMTP và nhiều ứng dụng khác. Đây là công cụ không thể thiếu để thực hiện

các kết nối mạng đồng bộ và hiệu quả, phục vụ nhu cầu của các ứng dụng trực tuyến, hệ thống phân phối dữ liệu và giao tiếp máy tính.



Hình 2.11 Mô hình Socket

2.6.2 Chức năng của Socket

Chức năng của Socket là tạo kết nối giữa client và server thông qua các giao thức TCP/IP hoặc UDP để truyền và nhận dữ liệu qua mạng Internet. Socket cung cấp một giao diện lập trình ứng dụng (API) giúp các ứng dụng giao tiếp với nhau một cách hiệu quả và ổn định.

Để Socket có thể hoạt động, các ứng dụng cần đáp ứng những điều kiện sau:

- Hai ứng dụng có thể nằm trên cùng một máy tính hoặc trên hai máy khác nhau.
- Khi hai ứng dụng nằm trên cùng một máy, mỗi ứng dụng phải sử dụng một số hiệu cổng (port) khác nhau. Điều này đảm bảo rằng mỗi ứng dụng có thể giao tiếp độc lập qua các cổng mạng mà không gây xung đột.

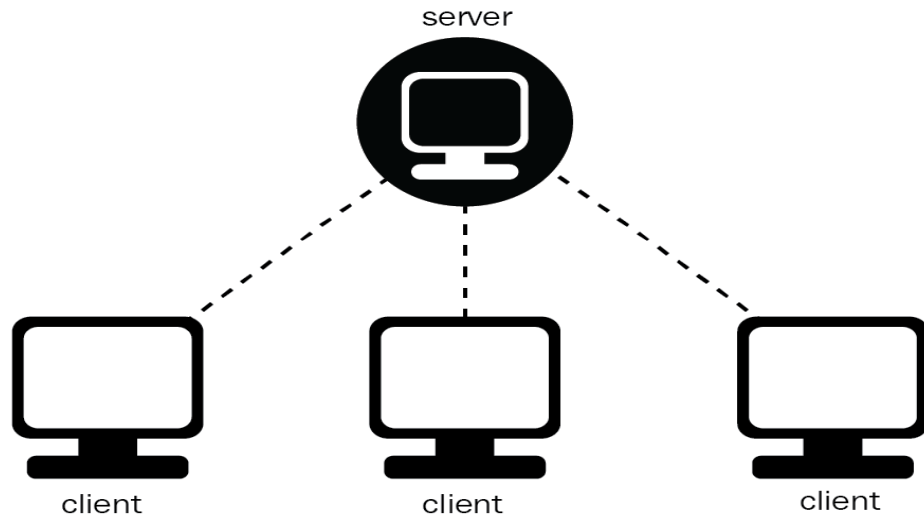
Socket sẽ tạo ra kết nối giữa hai ứng dụng, giúp chúng gửi và nhận dữ liệu qua mạng một cách ổn định và tin cậy, sử dụng các địa chỉ IP và số cổng để xác định nguồn và đích của

CHƯƠNG 3 PHÂN TÍCH YÊU CẦU HỆ THỐNG

3.1 Mô hình hệ thống

3.1.1 Mô hình Client-Server

- **Client (Ứng dụng phía người dùng)**
 - Là các ứng dụng phía người dùng, bao gồm một website và một ứng dụng di động (Android), cho phép sinh viên truy cập và sử dụng hệ thống.
- **Xây dựng các chức năng của Client:**
 - Giao diện người dùng: Cung cấp giao diện trực quan và dễ sử dụng để sinh viên có thể dễ dàng thao tác và tra cứu điểm học tập.
 - Gửi yêu cầu (requests): Gửi các yêu cầu HTTP/HTTPS đến server để truy xuất dữ liệu học tập, thực hiện tính toán GPA, và nhận thông tin phân tích.
 - Nhận phản hồi (responses): Nhận dữ liệu phản hồi từ server, chẳng hạn như kết quả điểm học tập, thông tin phân tích GPA, và hiển thị cho người dùng dưới dạng dễ hiểu (JSON hoặc XML).
- **Server (Máy chủ)**
 - Là phần xử lý trung tâm của hệ thống, chịu trách nhiệm tiếp nhận và xử lý các yêu cầu từ client, truy xuất dữ liệu từ cơ sở dữ liệu, thực hiện các phép toán tính toán GPA, và gửi phản hồi về cho client.
- **Xây dựng các chức năng của Server:**
 - Xử lý logic: Thực hiện các chức năng logic như tính toán GPA dựa trên các dữ liệu đầu vào (GPA hiện tại, GPA mong muốn, số tín chỉ đã hoàn thành, v.v.).
 - Quản lý và truy xuất dữ liệu: Kết nối với cơ sở dữ liệu (ví dụ: PostgreSQL, MySQL) để lưu trữ, truy xuất và quản lý điểm học tập và thông tin sinh viên.
 - Cung cấp API RESTful: Định nghĩa các API RESTful để client gửi yêu cầu và nhận dữ liệu từ server theo định dạng JSON hoặc XML, đảm bảo giao tiếp hiệu quả và bảo mật.



Hình 3.1 Mô hình hệ thống Client-Server

3.1.2 Nguyên tắc hoạt động

– **Client gửi yêu cầu đến Server**

- Giao diện người dùng: Website hoặc ứng dụng Android cung cấp các nút và giao diện để người dùng gửi yêu cầu đến server.
- Giao thức HTTP/HTTPS: Client sử dụng Fetch API để gửi yêu cầu HTTP hoặc HTTPS đến server. Ví dụ, khi người dùng nhấn nút “Tra cứu” trên website, yêu cầu GET sẽ được gửi đến endpoint `/api/getStudent/:mssv/:hocky` để lấy dữ liệu điểm học tập của sinh viên theo học kỳ cụ thể.

– **Server nhận và xử lý yêu cầu**

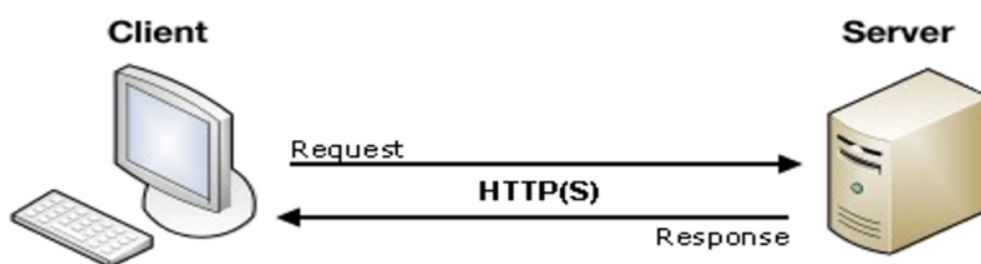
- Cấu trúc server: Hệ thống server được xây dựng bằng Node.js với framework Express, chịu trách nhiệm nhận và xử lý các yêu cầu từ client.
- Phân tích yêu cầu: Server phân tích thông tin yêu cầu, chẳng hạn như endpoint, tham số đầu vào (mã số sinh viên `:mssv`, học kỳ `:hocky`) và các dữ liệu liên quan.
- Xử lý logic: Server thực hiện các phép toán, truy vấn dữ liệu từ cơ sở dữ liệu (ví dụ: PostgreSQL, MySQL) hoặc gọi các dịch vụ khác nếu cần để xử lý yêu cầu.

- Trả kết quả: Sau khi xử lý xong, server trả về dữ liệu dưới dạng phản hồi JSON hoặc XML, trong đó mỗi object có các key và value tương ứng với thông tin điểm của sinh viên.

Ví dụ: Phản hồi từ server có thể là một danh sách các object chứa thông tin như họ tên, lớp, mã số sinh viên, khoa, học kỳ, cùng với các chi tiết điểm của từng môn học (mã môn học, tên môn học, điểm).

– Client nhận dữ liệu và hiển thị kết quả

- Nhận phản hồi: Fetch API tiếp nhận phản hồi từ server.
- Xử lý dữ liệu: Website hoặc ứng dụng Android xử lý dữ liệu nhận được từ server để chuẩn bị hiển thị.
- Hiển thị kết quả: Dữ liệu được hiển thị trên giao diện người dùng, giúp sinh viên dễ dàng tra cứu điểm học tập và phân tích kết quả theo học kỳ.



Hình 3.2 Nguyên tắc hoạt động

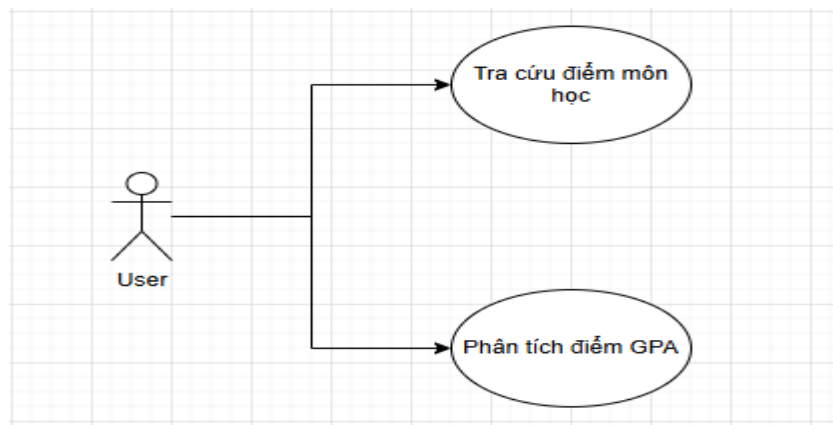
3.1.3 Sơ đồ User Case

Tra cứu điểm môn học

- Mô tả: Sinh viên có thể nhập mã số sinh viên và học kỳ để tra cứu điểm của mình.
- Tác nhân: Sinh viên.
- Mô tả chi tiết:
 - Sinh viên nhập mã số sinh viên và học kỳ.
 - Hệ thống gửi yêu cầu đến server để truy xuất dữ liệu điểm từ cơ sở dữ liệu.
 - Server trả về danh sách các môn học và điểm tương ứng.
 - Sinh viên xem kết quả trên giao diện của website hoặc app Android.

Phân tích điểm GPA

- Mô tả: Sinh viên có thể nhập các thông tin liên quan đến học tập như GPA hiện tại, GPA mục tiêu, số tín chỉ đã hoàn thành và số tín chỉ cần hoàn thành để hệ thống tính toán và phân tích GPA.
- Tác nhân: Sinh viên.
- Mô tả chi tiết:
 - Sinh viên nhập các thông tin như GPA hiện tại, GPA mục tiêu, số tín chỉ đã hoàn thành, và các môn học hệ hai tín chỉ hoặc ba tín chỉ.
 - Hệ thống gửi yêu cầu đến server để thực hiện các tính toán GPA dựa trên dữ liệu đã nhập.
 - Server thực hiện phân tích và trả về kết quả dự đoán điểm số cần đạt ở các môn học để đạt được GPA mục tiêu.
 - Sinh viên xem kết quả phân tích trên giao diện của website hoặc app Android.



Hình 3.3 Sơ đồ User Case

3.2 Cơ sở dữ liệu

3.2.1 Bảng Subject

Bảng Subject dùng để lưu trữ thông tin về các môn học của hệ thống, bao gồm mã môn, tên môn và số tín chỉ tương ứng.

Tên cột	Kiểu dữ liệu	Cho phép Null	Khóa
SubjectID	VARCHAR(10)	Không	Khóa chính (Primary Key)
SubjectName	VARCHAR(100)	Không	
Units	INT	Không	

Bảng 3.1 Bảng cơ sở dữ liệu Subject

3.2.2 Bảng Class

Bảng Class được sử dụng để lưu trữ thông tin liên quan đến các lớp học của sinh viên, bao gồm mã lớp, tên lớp, và học kỳ tương ứng.

Tên cột	Kiểu dữ liệu	Cho phép Null	Khóa
ClassID	VARCHAR(10)	Không	Khóa chính (Primary Key)
ClassName	VARCHAR(100)	Không	
<i>ClassYear</i>	<i>VARCHAR(10)</i>	<i>Không</i>	

Bảng 3.2 Bảng cơ sở dữ liệu Class

3.2.3 Bảng Student

Bảng Student được sử dụng để lưu trữ thông tin cơ bản của sinh viên, bao gồm mã số sinh viên, tên sinh viên, lớp học và khoa.

Tên cột	Kiểu dữ liệu	Cho phép Null	Khóa
StudentID	VARCHAR(10)	Không	Khóa chính (Primary Key)
StudentName	VARCHAR(100)	Không	
StudentClass	VARCHAR(50)	Không	
Department	VARCHAR(50)	Không	
ClassID	VARCHAR(10)	Không	Khóa ngoại (Foreign Key)

Bảng 3.3 Bảng cơ sở dữ liệu Student

3.2.4 Bảng StudentGrades

Bảng StudentGrades dùng để lưu trữ thông tin điểm số của sinh viên cho từng môn học trong từng lớp học cụ thể. Các khóa ngoại như student_id, subject_id, và class_id được sử dụng để liên kết với các bảng khác trong cơ sở dữ liệu, giúp dễ dàng truy vấn và phân tích dữ liệu.

Tên cột	Kiểu dữ liệu	Cho phép Null	Khóa
StudentID	VARCHAR(10)	Không	Một phần khóa chính (Primary Key) và khóa ngoại (Foreign Key)
SubjectID	VARCHAR(10)	Không	Một phần khóa chính (Primary Key) và khóa ngoại (Foreign Key)
Grades	DECIMAL(3, 2)	Có	

Bảng 3.4 Bảng cơ sở dữ liệu *StudentGrades*

3.3 Ngôn ngữ lập trình và công nghệ sử dụng

Để xây dựng hệ thống tra cứu kết quả học tập và phân tích GPA, nhóm đã lựa chọn một số công nghệ và công cụ hiện đại, giúp phát triển hệ thống hiệu quả và dễ bảo trì. Cụ thể như sau:

Backend:

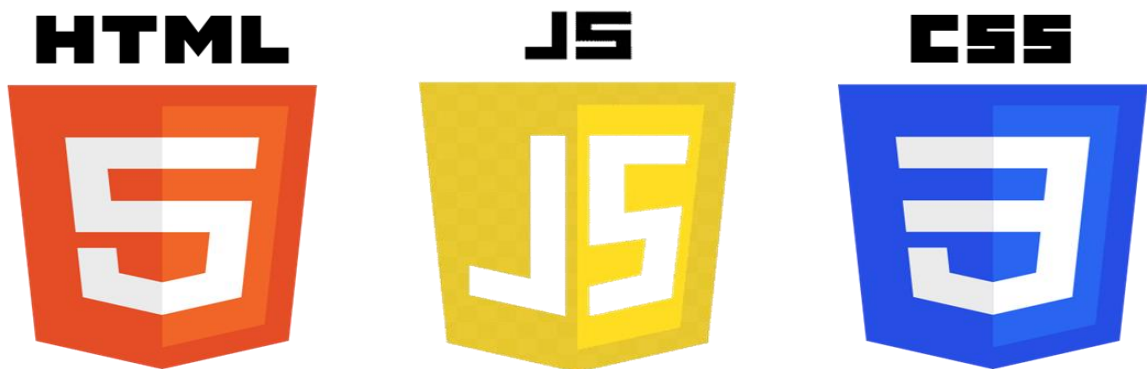
- **Node.js:** Node.js là môi trường chạy JavaScript phía server, nổi bật với khả năng xử lý các yêu cầu đồng thời một cách nhanh chóng nhờ vào kiến trúc không đồng bộ (asynchronous). Node.js cho phép xây dựng các ứng dụng server-side mạnh mẽ, dễ mở rộng và có hiệu suất cao, phù hợp với các ứng dụng web hiện đại.
- **Express.js:** Đây là một framework phổ biến và được ưa chuộng trong môi trường Node.js. Express.js hỗ trợ việc tạo các API RESTful dễ dàng và nhanh chóng, đồng thời cung cấp các tính năng cần thiết để xử lý các yêu cầu HTTP như GET, POST, PUT, và DELETE. Với Express.js, việc xây dựng logic backend trở nên đơn giản hơn, tiết kiệm thời gian và tăng tính linh hoạt cho hệ thống.
- **PostgreSQL:** Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, PostgreSQL nổi bật với khả năng hỗ trợ ACID, đảm bảo tính toàn vẹn dữ liệu. Nó cung cấp các tính năng tiên tiến như khả năng mở rộng, tính bảo mật cao và khả năng xử lý các truy vấn phức tạp một cách nhanh chóng. PostgreSQL rất phù hợp để lưu trữ và quản lý dữ liệu có cấu trúc, như thông tin sinh viên, điểm số, môn học, và các thông tin liên quan khác trong hệ thống.



Hình 3.4 Các ngôn ngữ lập trình Backend

Frontend:

- **HTML:** Được sử dụng để xây dựng cấu trúc của trang web, bao gồm các yếu tố như tiêu đề, đoạn văn, liên kết, hình ảnh, và các thành phần khác. HTML giúp định nghĩa cấu trúc nội dung của website, tạo nền tảng vững chắc để phát triển các phần tử khác.
- **CSS:** Được sử dụng để thiết kế và định dạng giao diện của trang web, từ layout đến màu sắc và phông chữ. CSS giúp làm cho giao diện của hệ thống trở nên bắt mắt, dễ nhìn và dễ sử dụng. Các tính năng như tính đáp ứng (responsive design) giúp giao diện hiển thị tốt trên nhiều thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động.
- **JavaScript:** Là ngôn ngữ lập trình phía client, cho phép xử lý các hành động của người dùng như gửi yêu cầu đến server mà không cần tải lại trang (sử dụng AJAX hoặc Fetch API). Ngoài ra, JavaScript có khả năng cập nhật và thay đổi nội dung của giao diện mà không làm mới toàn bộ trang, nâng cao trải nghiệm người dùng. Với JavaScript, hệ thống có thể tương tác động và xử lý các logic phức tạp trên giao diện, như việc nhận và hiển thị dữ liệu từ server.



Hình 3.5 Các ngôn ngữ lập trình Frontend

Ứng dụng Android:

- **Kotlin:** Kotlin là ngôn ngữ lập trình chính để phát triển ứng dụng Android, được Google chính thức hỗ trợ. Với cú pháp hiện đại, dễ đọc và dễ bảo trì, Kotlin giúp lập trình viên phát triển ứng dụng một cách nhanh chóng và hiệu quả hơn so với Java. Ngoài ra, Kotlin tích hợp tốt với Android Studio, cung cấp nhiều tính năng mạnh mẽ và cải tiến giúp tăng cường hiệu suất của ứng dụng.
- **RESTful API (Node.js, Express.js):** Để giao tiếp giữa ứng dụng Android và server backend (Node.js/Express.js), chúng tôi sử dụng API RESTful. Các API này cho phép ứng dụng gửi và nhận dữ liệu từ server thông qua các giao thức HTTP, giúp truyền tải thông tin giữa các hệ thống một cách nhanh chóng và an toàn.
- **Giao tiếp:** Frontend của ứng dụng Android sử dụng các phương thức như Fetch API hoặc các thư viện hỗ trợ như Retrofit để gửi yêu cầu HTTP đến server và nhận phản hồi. Thông qua giao tiếp này, ứng dụng có thể truy xuất dữ liệu từ server, xử lý và hiển thị thông tin cho người dùng.



Hình 3.6 Các ngôn ngữ và công cụ lập trình Android

CHƯƠNG 4 XÂY DỰNG CHƯƠNG TRÌNH

4.1 Server

Xây dựng các API RESTful tại server để phục vụ các yêu cầu từ client. Server sử dụng Node.js và framework Express.js để phát triển và triển khai các chức năng này.

4.1.1 Xây dựng API RESTful

API này sử dụng phương thức GET và nhận yêu cầu từ client với hai tham số: Mã số sinh viên và Học kỳ. Mục đích của API là truy vấn cơ sở dữ liệu để lấy thông tin điểm số của sinh viên trong học kỳ cụ thể.

Quy trình hoạt động:

- Nhận yêu cầu GET từ client với hai tham số là Mã số sinh viên và Học kỳ.
- Truy vấn cơ sở dữ liệu sử dụng Mã số sinh viên và Học kỳ để lấy thông tin từ các bảng Student, Class, Subject, và StudentGrades.
- Nếu truy vấn thành công (có dữ liệu điểm số sinh viên trong học kỳ đó), API trả về dữ liệu dưới dạng JSON. Dữ liệu trả về sẽ bao gồm thông tin như:
 - Thông tin sinh viên: mã số sinh viên, họ và tên, lớp, chuyên ngành, năm học.
 - Bảng điểm môn học: mã môn học, tên môn học, và điểm số.
- Nếu không có kết quả (không có điểm sinh viên trong học kỳ đó), API sẽ trả về mã lỗi và thông báo lỗi.

API này cung cấp thông tin điểm học tập của sinh viên trong các học kỳ cụ thể, giúp sinh viên dễ dàng tra cứu và đánh giá tiến trình học tập.

4.1.2 API phân tích GPA sinh viên

API này sử dụng phương thức POST và nhận dữ liệu từ phía client, bao gồm thông tin về GPA hiện tại, GPA mong muốn, số tín chỉ đã hoàn thành, số tín chỉ cần hoàn thành khóa học, và số môn 2 tín chỉ, 3 tín chỉ còn lại. API này giúp tính toán và đưa ra kế hoạch để đạt được GPA mong muốn.

Quy trình hoạt động:

- Nhận dữ liệu từ client qua phương thức POST.
- Chuyển đổi các giá trị nhận được từ kiểu chuỗi sang kiểu số để xử lý.
- Tính giá trị X (GPA cần đạt được từ các môn học còn lại):

- Nếu GPA hiện tại = 0 (chưa có GPA), thì $X = \text{GPA mong muốn}$.
- Nếu không, $X = 2 * \text{GPA mong muốn} - \text{GPA hiện tại}$.
- Kiểm tra giá trị X:
 - Nếu $X < 0$ hoặc $X > 4$, API sẽ thông báo rằng không thể đạt được GPA mong muốn và trả về GPA tối đa có thể đạt được.
 - Nếu X hợp lệ, sử dụng thuật toán vét cạn để tính toán các giá trị điểm số hợp lý từ mảng $\{1, 1.5, 2, 2.5, 3, 3.5, 4\}$ cho các môn 2 tín chỉ và 3 tín chỉ để đạt được GPA mong muốn.
- Kết quả trả về dưới dạng JSON, bao gồm các giá trị đã tính toán cho điểm môn 2 tín chỉ và 3 tín chỉ cần đạt được.

API này cung cấp khả năng tính toán các giá trị điểm học tập cần đạt để đạt được GPA mong muốn, giúp sinh viên có thể lên kế hoạch học tập hợp lý hơn.

4.1.3 Cấu hình và khởi tạo Server

- Import các thư viện cần thiết:
 - express: Framework chính để xây dựng server.
 - cors: Middleware để cho phép server chấp nhận các yêu cầu từ các nguồn khác nhau.
 - os: Thư viện để truy cập thông tin hệ thống.
 - config: Module để quản lý cấu hình kết nối với cơ sở dữ liệu.
- Middleware sử dụng:
 - cors(): Cho phép server chấp nhận các yêu cầu từ các nguồn khác nhau, hỗ trợ CORS.
 - express.json(): Giúp xử lý các yêu cầu HTTP có nội dung là JSON, giúp nhận dữ liệu từ client.
- Các route được thiết kế để xử lý các yêu cầu từ client, bao gồm:
 - Route GET /api/getStudent/:mssv/:hockey để tra cứu điểm sinh viên.
 - Route POST /api/analyzeGPA để phân tích GPA.
- Kết nối cơ sở dữ liệu:

- Kết nối server với cơ sở dữ liệu PostgreSQL thông qua module pg hoặc sequelize, đảm bảo rằng server có thể thực hiện các truy vấn và thao tác dữ liệu.
- Khởi chạy server:
 - Server được khởi chạy trên một cổng nhất định (ví dụ, `app.listen(3000)`), sẵn sàng nhận và xử lý các yêu cầu từ client.

4.2 Client

4.2.1 Website

Trang web của hệ thống sẽ cung cấp các giao diện trực quan để sinh viên thực hiện hai chức năng chính: tra cứu điểm sinh viên và phân tích GPA. Dưới đây là mô tả chi tiết về từng trang.

4.2.1.1 Trang tra cứu kết quả học tập sinh viên

Sinh viên có thể nhập Mã số sinh viên và lựa chọn học kỳ của mình để tra cứu điểm học tập.

Quy trình hoạt động:

- Nhập thông tin: Sinh viên nhập mã số sinh viên và chọn học kỳ trên form của trang web.
- Gửi yêu cầu GET:
 - Hệ thống sử dụng phương thức `fetch()` để gửi yêu cầu HTTP GET đến server, truyền kèm thông tin mã số sinh viên và học kỳ.
 - Yêu cầu sẽ được gửi tới endpoint của API server `/api/getStudent/:mssv/:hocky`.
- Nhận dữ liệu từ server:
 - Server nhận yêu cầu, thực hiện truy vấn cơ sở dữ liệu và trả về dữ liệu dưới dạng JSON.
- Xử lý và hiển thị dữ liệu:
 - Hệ thống xử lý dữ liệu nhận được và hiển thị các thông tin như: họ tên, mã số sinh viên, lớp học, học kỳ, và bảng điểm chi tiết.

- Nếu không có dữ liệu hoặc có lỗi, hệ thống sẽ hiển thị thông báo lỗi trên màn hình.

Xây dựng một hàm `getStudentGrade` phục vụ cho việc lấy và hiển thị thông tin sinh viên từ API của Server. Nếu dữ liệu đúng thì sẽ hiển thị đầy đủ thông tin cho sinh viên gồm: họ tên, mã số sinh viên, lớp, học kỳ và bảng điểm học tập. Nếu không có dữ liệu hoặc sai sẽ hiện thông báo ở trên màn hình.

4.2.1.2 Trang phân tích GPA

Trang phân tích GPA cho phép sinh viên nhập thông tin về GPA hiện tại, GPA mong muốn, số tín chỉ đã hoàn thành, số tín chỉ cần hoàn thành, và số môn học 2 tín chỉ và 3 tín chỉ còn lại. Hệ thống sẽ tính toán và trả về các điểm số cần đạt để đạt được GPA mong muốn, hoặc GPA tối đa có thể đạt được.

Quy trình hoạt động:

- Nhập thông tin: Sinh viên nhập các thông tin cần thiết vào form.
- Kiểm tra tính hợp lệ:
 - Hàm `calculateGPA()` sẽ kiểm tra các điều kiện cần thiết, ví dụ như tổng số tín chỉ hiện tại, số tín chỉ cần hoàn thành, và các môn học 2 tín chỉ, 3 tín chỉ.
 - Ví dụ, nếu số tín chỉ hiện tại là 78, số tín chỉ cần hoàn thành là 120, và số môn 2 tín chỉ, 3 tín chỉ là 3 và 12, thì $(2 * 3 + 3 * 12) = (120 - 78) = 42$.
- Gửi yêu cầu POST:
 - Nếu các điều kiện hợp lệ, dữ liệu sẽ được gửi lên server qua yêu cầu HTTP POST sử dụng `fetch()`, dưới dạng JSON.
 - Endpoint API server được sử dụng là `/api/analyzeGPA`.
- Nhận và xử lý dữ liệu từ server:
 - Server trả về kết quả tính toán, bao gồm các điểm số cần đạt cho môn 2 tín chỉ và 3 tín chỉ.
 - Nếu không thể đạt được GPA mong muốn, server sẽ trả về GPA tối đa có thể đạt được nếu các môn còn lại đều đạt điểm tối đa là 4.
- Hiển thị kết quả:

- Hệ thống xử lý kết quả nhận được và hiển thị thông tin cho sinh viên. Nếu có lỗi hoặc dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi.

Cung cấp một phương thức tính toán để sử dụng và trực quan cho sinh viên trong việc phân tích GPA để sinh viên có thể lên kế hoạch học tập trong tương lai có thể tốt hơn.

4.2.2 Android

Ứng dụng Android của hệ thống cung cấp hai tính năng chính: tra cứu điểm sinh viên và phân tích GPA. Dưới đây là mô tả chi tiết về từng tính năng.

4.2.2.1 Trang tra cứu kết quả học tập

Ứng dụng cho phép sinh viên nhập mã số sinh viên và chọn học kỳ để tra cứu điểm học tập. Kết quả sẽ hiển thị thông tin sinh viên và bảng điểm của các môn học trong học kỳ đã chọn.

Quy trình hoạt động:

- Khởi tạo giao diện chính:
 - Lớp MainActivity được tạo và thiết lập giao diện người dùng thông qua phương thức setContent.
 - Giao diện GradeCheckerDisplay bao gồm các thành phần như logo trường, các trường nhập liệu (mã số sinh viên, học kỳ), và nút "Tra cứu".
- Nhập và gửi yêu cầu:
 - Sinh viên nhập mã số sinh viên và chọn học kỳ. Khi nhấn nút "Tra cứu", ứng dụng lấy giá trị mã số sinh viên và học kỳ đã chọn, sau đó tạo URL yêu cầu API để gửi đến máy chủ.
- Gửi yêu cầu và nhận dữ liệu:
 - Ứng dụng sử dụng OkHttpClient để gửi yêu cầu GET tới API của máy chủ.
 - Phương thức getStudentInfoFromServer() sẽ được gọi để gửi yêu cầu và nhận phản hồi từ máy chủ.
- Xử lý dữ liệu JSON trả về:
 - Máy chủ trả về dữ liệu dưới dạng JSON chứa thông tin sinh viên và bảng điểm các môn học.

- Hàm `parseStudentInfoFromJson()` phân tích dữ liệu JSON và chuyển đổi thành các đối tượng `StudentInfo` và `GradeInfo`.
- Mỗi đối tượng `StudentInfo` chứa thông tin sinh viên (tên, mã số sinh viên, lớp học, khoa) và một danh sách các đối tượng `GradeInfo`, mỗi `GradeInfo` chứa thông tin về môn học và điểm số.

4.2.2.2 Trang phân tích GPA

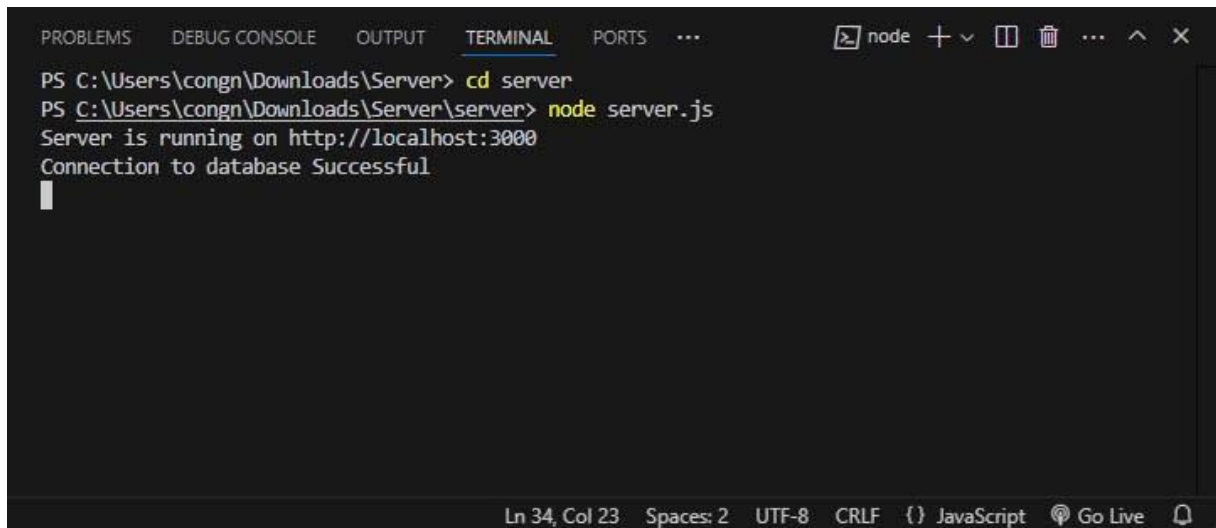
Ứng dụng cung cấp tính năng phân tích GPA để giúp sinh viên tính toán điểm số cần đạt được để đạt được GPA mong muốn. Sinh viên nhập các thông tin như GPA hiện tại, GPA mong muốn, số tín chỉ hiện tại, và số môn học 2 tín chỉ và 3 tín chỉ còn lại.

Quy trình hoạt động:

- Khởi tạo giao diện phân tích GPA:
 - Lớp `GPA`, kế thừa từ `ComponentActivity`, được tạo và giao diện người dùng được thiết lập trong phương thức `onCreate`.
 - Giao diện sử dụng `setContent` để hiển thị `GPA(navController)` là màn hình chính của ứng dụng.
- Nhập dữ liệu và gửi yêu cầu POST:
 - Người dùng nhập các thông tin về GPA và tín chỉ vào các trường văn bản.
 - Khi nhấn nút "Phân tích", ứng dụng gửi yêu cầu POST tới máy chủ qua Retrofit. Dữ liệu từ form được gửi dưới dạng JSON trong một đối tượng `GPARequest`.
- Nhận và xử lý dữ liệu từ máy chủ:
 - Máy chủ trả về kết quả dưới dạng một danh sách các đối tượng `Result`, mỗi đối tượng chứa thông tin điểm của các môn học 2 tín chỉ và 3 tín chỉ.
 - Ứng dụng hiển thị các môn học cần chú ý để đạt GPA mong muốn hoặc thông báo GPA tối đa có thể đạt được.

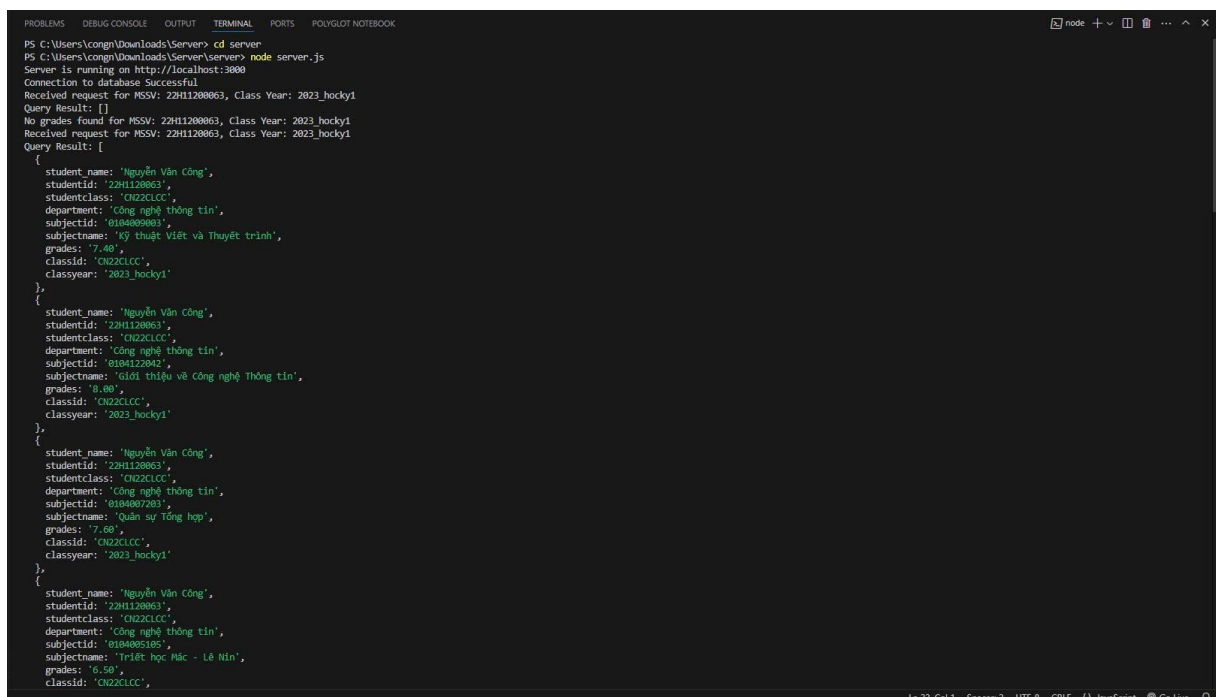
CHƯƠNG 5 KẾT QUẢ THỰC THI

5.1 Server



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS  ...  node + - [ ] [ ] ... ^ X
PS C:\Users\cong\Downloads\Server> cd server
PS C:\Users\cong\Downloads\Server\server> node server.js
Server is running on http://localhost:3000
Connection to database Successful
█
Ln 34, Col 23  Spaces: 2  UTF-8  CRLF  ( ) JavaScript  Go Live  [ ]
```

Hình 5.1 Bắt đầu khởi chạy Server



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS  POLYGLOT NOTEBOOK  node + - [ ] [ ] ... ^ X
PS C:\Users\cong\Downloads\Server> cd server
PS C:\Users\cong\Downloads\Server\server> node server.js
Server is running on http://localhost:3000
Connection to database Successful
Received request for MSSV: 22H11200063, Class Year: 2023_hocky1
Query Result: []
No grades found for MSSV: 22H11200063, Class Year: 2023_hocky1
Received request for MSSV: 22H11200063, Class Year: 2023_hocky1
Query Result: [
  {
    student_name: 'Nguyễn Văn Công',
    studentId: '22H11200063',
    studentclass: 'OI22CLCC',
    department: 'Công nghệ thông tin',
    subjectId: '0104000003',
    subjectname: 'Kỹ thuật Viết và Thuyết trình',
    grades: '7.40',
    classId: 'OI22CLCC',
    classyear: '2023_hocky1'
  },
  {
    student_name: 'Nguyễn Văn Công',
    studentId: '22H11200063',
    studentclass: 'OI22CLCC',
    department: 'Công nghệ thông tin',
    subjectId: '01040120042',
    subjectname: 'Giới thiệu về Công nghệ Thông tin',
    grades: '8.00',
    classId: 'OI22CLCC',
    classyear: '2023_hocky1'
  },
  {
    student_name: 'Nguyễn Văn Công',
    studentId: '22H11200063',
    studentclass: 'OI22CLCC',
    department: 'Công nghệ thông tin',
    subjectId: '0104007203',
    subjectname: 'Quản sự Tổng hợp',
    grades: '7.60',
    classId: 'OI22CLCC',
    classyear: '2023_hocky1'
  },
  {
    student_name: 'Nguyễn Văn Công',
    studentId: '22H11200063',
    studentclass: 'OI22CLCC',
    department: 'Công nghệ thông tin',
    subjectId: '01040005105',
    subjectname: 'Triết học Mác - Lê Nin',
    grades: '6.50',
    classId: 'OI22CLCC',
    classyear: '2023_hocky1'
  }
]
Ln 32, Col 1  Spaces: 2  UTF-8  CRLF  ( ) JavaScript  Go Live  [ ]
```

Hình 5.2 Server trả request tra cứu điểm học tập

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS
PS C:\Users\cong\Downloads\Server> cd server
PS C:\Users\cong\Downloads\Server\server> node server.js
Server is running on http://localhost:3000
Connection to database successful
Dữ liệu nhận từ client: {
  currentGPA: '2.5',
  desiredGPA: '3.0',
  currentCredits: '50',
  totalCredits: '120',
  remaining2CreditCourses: '5',
  remaining3CreditCourses: '10'
}
Kết quả tính toán: [
  { Y: '2.00', Z: '4.00' },
  { Y: '2.50', Z: '4.00' },
  { Y: '3.00', Z: '4.00' },
  { Y: '3.50', Z: '3.50' },
  { Y: '3.50', Z: '4.00' },
  { Y: '4.00', Z: '3.50' },
  { Y: '4.00', Z: '4.00' }
]
Dữ liệu nhận từ client: {
  currentGPA: '3.60',
  desiredGPA: '3.75',
  currentCredits: '70',
  totalCredits: '120',
  remaining2CreditCourses: '3',
  remaining3CreditCourses: '12'
}
Kết quả tính toán: [
  { Y: '3.00', Z: '4.00' },
  { Y: '3.50', Z: '4.00' },
  { Y: '4.00', Z: '4.00' }
]
}
```

Hình 5.3 Server trả request phân tích GPA

5.2 Website

The screenshot shows the website of the University of Transport Ho Chi Minh City (UTH). The header features the university's logo and name. Below the header, there are two main navigation buttons: "TRA CỨU ĐIỂM SINH VIÊN" (Student Grade Lookup) and "PHÂN TÍCH ĐIỂM GPA" (GPA Analysis). The "TRA CỨU ĐIỂM SINH VIÊN" button is highlighted. Below this button, there is a form with the following fields:

- Mã số sinh viên: (Student ID field)
- Chọn học kỳ: (Select semester dropdown menu)
- Học kỳ 1 năm học 2022-2023 (Selected semester)
- TRA CỨU ĐIỂM (Lookup Grade button)

Hình 5.4 Giao diện website khi chưa thực hiện tra cứu điểm

TRA CỨU ĐIỂM SINH VIÊN

Mã số sinh viên:

22H1120063

Chọn học kỳ:

Học kỳ 1 năm học 2023-2024

TRA CỨU ĐIỂM

Họ và tên: Nguyễn Văn Công Mã số sinh viên: 22H1120063

Lớp: CN22CLCC Khoa: Công nghệ thông tin

Mã môn học	Tên môn học	Điểm
0104009003	Kỹ thuật Viết và Thuyết trình	7.40
0104122042	Giới thiệu về Công nghệ Thông tin	8.00
0104007203	Quản sự Tổng hợp	7.60
0104005105	Triết học Mác - Lê Nin	6.50
0104124012	Tin học cơ sở	8.10
0104007201	Chiến lược Quốc phòng và An ninh của Đảng Cộng sản Việt Nam	8.20
0104007202	Hoạt động Quốc phòng và An ninh	7.70
0104001202	Giải tích 1	6.80
0104001205	Toán nâng cao 1	7.90
0104007204	Kỹ thuật và chiến thuật chiến đấu bộ binh	8.30
0104005004	Pháp luật đại cương	6.60
0104122002	Toán rời rạc	7.50
0104124001	Kỹ thuật lập trình	7.80

Hình 5.5 Giao diện website khi thực hiện tra cứu điểm

PHÂN TÍCH GPA

GPA hiện tại:

GPA mong muốn:

Tin chỉ hiện tại:


Tin chỉ cần hoàn thành:

Số môn 2 chỉ còn lại:

Số môn 3 chỉ còn lại:

PHÂN TÍCH

Hình 5.6 Giao diện website khi chưa thực hiện phân tích GPA



UNIVERSITY

OF TRANSPORT

HOCHIMINH CITY

TRA CỨU ĐIỂM SINH VIÊN

PHÂN TÍCH ĐIỂM GPA

PHÂN TÍCH GPA

GPA hiện tại:

3.68

GPA mong muốn:

3.75

Tín chỉ hiện tại:

78

Tín chỉ cần hoàn thành:

120

Số môn 2 chỉ còn lại:

3

Số môn 3 chỉ còn lại:

12

PHÂN TÍCH

Chúng tôi tính điểm của bạn khoảng từ GPA mong muốn đến nhỏ hơn 4.0 nha, ai mà chẳng muốn mình giỏi hơn đúng không nè:)))

Số điểm các môn 2 tín chỉ và 3 tín chỉ bạn cần đạt để đúng theo mong muốn của bạn là:

Môn 2 chỉ : 3.00
Môn 3 chỉ : 4.00.

Môn 2 chỉ : 3.50
Môn 3 chỉ : 4.00.

Môn 2 chỉ : 4.00
Môn 3 chỉ : 4.00.

Cảm ơn bạn đã quan tâm đến chúng tôi, chúc bạn một ngày tốt lành!!!

BẢNG QUY ĐỔI ĐIỂM

Điểm hệ 10	Điểm hệ 4
8.5 – 10	4.0
8.0 – 8.4	3.5
7.0 – 7.9	3.0
6.0 – 6.9	2.5
5.5 – 5.9	2.0
5.0 – 5.4	1.5
4.0 – 4.9	1.0
2.1 – 3.9	0.5
0 – 2	0

Hình 5.7 Giao diện website khi thực hiện phân tích GPA

5.3 Android

The left screenshot shows the login screen of the UTH mobile app. It features the UTH logo and the text "UNIVERSITY OF TRANSPORT HOCHIMINH CITY". Below the logo, there are two tabs: "TRA CỨU ĐIỂM SINH VIÊN" and "PHÂN TÍCH GPA". The "TRA CỨU ĐIỂM SINH VIÊN" tab is selected. There are two input fields: "Mã số sinh viên" (Student ID) and "Lựa chọn học kỳ" (Select semester). The "Mã số sinh viên" field contains the text "22H1120063". The "Lựa chọn học kỳ" dropdown menu is set to "Học kỳ 1 năm học 2022-2023". A "Tra cứu" (Search) button is located at the bottom of the form.

The right screenshot shows the GPA analysis screen. It features the same UTH logo and text. Below the logo, there are two tabs: "TRA CỨU ĐIỂM SINH VIÊN" and "PHÂN TÍCH GPA". The "PHÂN TÍCH GPA" tab is selected. There are two input fields: "Mã số sinh viên" (Student ID) and "Lựa chọn học kỳ" (Select semester). The "Mã số sinh viên" field contains the text "22H1120063". The "Lựa chọn học kỳ" dropdown menu is set to "Học kỳ 1 năm học 2023-2024". A "Tra cứu" (Search) button is located at the bottom of the form. Below the search button, the following information is displayed:

Họ và Tên: Nguyễn Văn Công
Mã số sinh viên: 22H1120063
Lớp: CN22CLCC
Khoa: Công nghệ thông tin
Học kỳ: Học kỳ 1 năm học 2023-2024

Mã môn học	Tên môn học	Điểm
0104009003	Kỹ thuật Viết và Thuyết trình	7.40
0104122042	Giới thiệu về Công nghệ Thông tin	8.00
0104007203	Quản sự Tổng hợp	7.60
0104005105	Triết học Mác - Lê Nin	6.50
0104124012	Tin học cơ sở	8.10

Hình 5.8 Giao diện android khi tra cứu điểm



Hình 5.9 Giao diện app android khi phân tích GPA

CHƯƠNG 6 KẾT LUẬN

6.1 Kết luận

Dựa trên mô hình môn Lập trình mạng, hệ thống đã thành công trong việc triển khai các chức năng tra cứu điểm học tập và phân tích GPA. Các API RESTful được xây dựng đảm bảo khả năng xử lý truy vấn và phản hồi dữ liệu chính xác, hiệu quả. Hệ thống cho phép sinh viên dễ dàng tra cứu thông tin điểm số theo học kỳ và hỗ trợ tính toán để đạt được GPA mục tiêu. Điều này không chỉ đáp ứng đúng yêu cầu của bài toán mà còn tạo ra một công cụ hữu ích cho việc quản lý và lập kế hoạch học tập của sinh viên.

Việc áp dụng các công nghệ hiện đại và mô hình Client-Server đã giúp hệ thống đạt được tính ổn định, khả năng mở rộng và dễ dàng bảo trì. Hệ thống đã chứng minh được tính thực tiễn và hiệu quả trong việc giải quyết bài toán của môn học, đồng thời mở ra tiềm năng phát triển thêm trong tương lai.

6.2 Đề xuất

Để nâng cao hơn nữa khả năng của hệ thống, chúng tôi đề xuất tích hợp các công nghệ trí tuệ nhân tạo (AI) nhằm phân tích và hỗ trợ sinh viên trong việc học tập, cụ thể như sau:

- Tư vấn học tập cá nhân hóa: Sử dụng AI để phân tích lịch sử học tập, điểm số, và sở thích của từng sinh viên nhằm đưa ra các gợi ý tối ưu. Đưa ra các đề xuất:
 - Lịch học cá nhân hóa, cân đối giữa môn học khó và dễ.
 - Danh sách môn học cần chú ý dựa trên các môn có xu hướng điểm số thấp hoặc ảnh hưởng lớn đến GPA.
 - Chiến lược học tập và lời khuyên cải thiện điểm số dựa trên phân tích dữ liệu cá nhân.
- Phân tích xu hướng học tập: AI sẽ phân tích xu hướng điểm số của sinh viên qua từng học kỳ, nhận diện các môn học có dấu hiệu sa sút hoặc xu hướng không ổn định. Từ đó đưa ra các đề xuất:
 - Cảnh báo sớm cho sinh viên khi có dấu hiệu suy giảm về thành tích học tập.
 - Cung cấp báo cáo phân tích chi tiết để sinh viên và nhà trường can thiệp kịp thời.

- Gợi ý môn học tối ưu: AI sẽ phân tích điểm mạnh, điểm yếu và tiến trình học tập của sinh viên, từ đó đưa ra các gợi ý như:
 - Danh sách môn học phù hợp để tối ưu hóa thời gian học tập và kết quả GPA.
 - Lộ trình học tập khoa học để sinh viên dễ dàng đạt được mục tiêu tốt nghiệp đúng hạn hoặc xuất sắc.

Việc tích hợp AI không chỉ tăng cường tính năng của hệ thống mà còn mang đến trải nghiệm học tập thông minh hơn, hỗ trợ sinh viên đạt được mục tiêu học tập hiệu quả và dễ dàng hơn trong kỷ nguyên công nghệ hiện đại.