

TRƯỜNG ĐẠI HỌC BÁCH KHOA
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MẠNG MÁY TÍNH (CO3094)

BÀI TẬP LỚN

*Implement HTTP server
and chat application*

Giáo viên hướng dẫn: Bùi Xuân Giang, CSE-HCMUT

Sinh viên: Nguyễn Tân Phát - 2352888 (*CN04*)
Vũ Hà Như Ngọc - 2352818 (*CN04*)
Lê Diệu Quỳnh - 2353036 (*CN04*)
Lương Đức Huy - 2352384 (*CN04*)

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 11 NĂM 2025



Mục lục

Danh sách Ký hiệu	3
Danh sách Từ viết tắt	3
Danh sách Hình ảnh	5
Danh sách Bảng	5
Danh sách thành viên & khối lượng công việc	5
Chương 1: Mở đầu	6
1. Đặt vấn đề và mục tiêu	6
2. Phạm vi thực hiện	6
3. Bố cục báo cáo	7
Chương 2: Cơ sở lý thuyết	8
1. Mô hình Client-Server và Peer-to-Peer (P2P)	8
2. Lập trình Socket và Luồng (Socket & Threading)	8
3. Giao thức HTTP và Cookie	8
4. Giao thức WebSocket	8
Chương 3: Phân tích và thiết kế hệ thống	9
1. Kiến trúc tổng thể	9
2. Thiết kế các Module chính	9
3. Thiết kế Giao thức	9
Chương 4: Hiện thực chương trình	10
1. Môi trường và Công cụ	10
2. Hiện thực các chức năng chính	10
Chương 5: Kiểm thử và kết quả	11
1. Kịch bản kiểm thử	11
2. Kết quả thực nghiệm	11
Chương 6: Kết luận	12
1. Tổng kết kết quả đạt được	12



2. Hạn chế	12
3. Hướng phát triển	12



Danh sách Ký hiệu

Danh sách Từ viết tắt

Danh sách Hình ảnh

Danh sách Bảng



Danh sách thành viên & khối lượng công việc

STT	Họ Tên	MSSV	Vai trò	% Hoàn thành
1	Nguyễn Tấn Phát	2352888	Code	100%
2	Vũ Hà Như Ngọc	2352818	Code	100%
3	Lê Diệu Quỳnh	2353036	Code	100%
4	Lương Đức Huy	2352384	Code	100%

Bảng 1: Danh sách thành viên & khối lượng công việc



Chương 1: Mở đầu

1. Đặt vấn đề và mục tiêu

Trong bối cảnh Internet và các ứng dụng mạng ngày càng phát triển mạnh mẽ, việc hiểu rõ và áp dụng được các nguyên lý cơ bản của mạng máy tính là điều cần thiết đối với sinh viên ngành Khoa học máy tính. Các mô hình giao tiếp như **Client-Server** và **Peer-to-Peer (P2P)**, cùng với kỹ thuật lập trình socket, đóng vai trò nền tảng giúp kết nối các hệ thống và tạo nên các dịch vụ mạng hiện nay.

Bài tập lớn này được thực hiện nhằm áp dụng những kiến thức lý thuyết đã học trong môn *Mạng máy tính* vào việc xây dựng một hệ thống phần mềm thực tế, qua đó giúp người học củng cố và hiểu sâu hơn cách các tiến trình mạng tương tác với nhau trong môi trường phân tán.

Cụ thể, đề tài tập trung vào hai nhiệm vụ chính sau:

- **Xây dựng HTTP Server có cơ chế quản lý phiên bằng Cookie:** Nhiệm vụ này yêu cầu sinh viên hiện thực một máy chủ web sử dụng thư viện *socket* cơ bản của Python, có khả năng xử lý các yêu cầu HTTP (GET/POST) và đặc biệt là triển khai cơ chế xác thực người dùng. Hệ thống cần duy trì trạng thái đăng nhập thông qua việc sử dụng *HTTP Cookie*, giúp làm rõ cách thức vượt qua tính chất *stateless* của giao thức HTTP.
- **Xây dựng ứng dụng chat lai (Hybrid Chat Application):** Ứng dụng này kết hợp linh hoạt hai mô hình Client-Server và P2P. Trong đó, mô hình *Client-Server* được dùng để quản lý danh sách người dùng và kênh chat thông qua một *Tracker Server*, còn mô hình *P2P* cho phép các người dùng (Peer Client) kết nối trực tiếp và trao đổi tin nhắn mà không cần qua máy chủ trung gian. Mục tiêu là giúp sinh viên hiểu rõ cách thiết kế một hệ thống giao tiếp hai chiều có tính mở rộng cao và phản hồi thời gian thực.

Qua việc hoàn thành hai nhiệm vụ trên, đề tài hướng đến mục tiêu xây dựng một ứng dụng mạng hoàn chỉnh, minh họa rõ cách phối hợp giữa các tiến trình *client*, *server* và *tracker* trong một kiến trúc hybrid hiện đại.

2. Phạm vi thực hiện

Để đáp ứng các mục tiêu đã đề ra, dự án tập trung thiết kế và hiện thực một hệ thống hoàn chỉnh với các chức năng cốt lõi sau:

- **HTTP Server và Proxy cơ bản:** Xây dựng thành công máy chủ HTTP từ thư viện *socket* cấp thấp, có khả năng xử lý yêu cầu GET và POST. Hệ thống cũng bao gồm Proxy Server đơn giản để chuyển tiếp request HTTP, minh họa cho kiến trúc nhiều lớp trong ứng dụng web thực tế.
- **Cơ chế xác thực và quản lý phiên:** Hiện thực quy trình xác thực người dùng. Sau khi đăng nhập thành công, máy chủ thiết lập HTTP Cookie trên trình duyệt để duy trì trạng thái phiên – một yếu tố cơ bản trong các ứng dụng web hiện đại.
- **Kiến trúc Ứng dụng Chat Hybrid:** Gồm hai thành phần chính:
 - *Tracker Server:* Đóng vai trò máy chủ trung tâm, quản lý việc đăng ký, xác thực và khám phá người dùng.
 - *Peer Client:* Vừa là client của Tracker Server, vừa là server trong các kết nối P2P.
- **Giao thức Client-Server cho việc khám phá Peer:** Cho phép Peer Client gửi thông tin (địa chỉ IP, port) lên Tracker Server để đăng ký, đồng thời truy vấn danh sách các peer khác đang trực tuyến.



- **Giao thức Peer-to-Peer để giao tiếp trực tiếp:** Hai Peer Client có thể thiết lập kết nối TCP trực tiếp sau khi biết địa chỉ của nhau. Cơ chế “bắt tay” (handshake) được triển khai để xác nhận kết nối trước khi trao đổi dữ liệu.
- **Hỗ trợ đa dạng hình thức Chat:**
 - Chat riêng giữa hai peer.
 - Chat nhóm qua các kênh chung (#general, #test, ...).
 - Kênh phát sóng (#broadcast) gửi tin đến toàn bộ người dùng trực tuyến.
- **Tích hợp giao diện Web:** Xây dựng giao diện người dùng bằng HTML, CSS và JavaScript, mang lại trải nghiệm chat trực quan và thân thiện.
- **Giao tiếp thời gian thực qua WebSocket:** Sử dụng giao thức WebSocket để kết nối hai chiều giữa backend Python và giao diện người dùng trên trình duyệt, đảm bảo phản hồi tức thì khi có tin nhắn mới.
- **Duy trì trạng thái đăng nhập phía Client:** Lưu thông tin phiên làm việc bằng *localStorage*, giúp người dùng tự động kết nối lại mà không cần thao tác thủ công, nâng cao trải nghiệm sử dụng.

3. Bố cục báo cáo

Để trình bày một cách hệ thống và rõ ràng quá trình thực hiện đề tài, nội dung báo cáo được cấu trúc thành các chương sau:

- **Chương 1 – Mở đầu:** Giới thiệu tổng quan về đề tài, xác định các mục tiêu, phạm vi chức năng đã được hiện thực và trình bày bố cục tổng thể của báo cáo.
- **Chương 2 – Cơ sở lý thuyết:** Trình bày các kiến thức nền tảng và công nghệ cốt lõi được sử dụng trong dự án, bao gồm các mô hình mạng Client-Server và Peer-to-Peer, lập trình Socket đa luồng, giao thức HTTP, cơ chế hoạt động của Cookie và vai trò của giao thức WebSocket.
- **Chương 3 – Phân tích và Thiết kế hệ thống:** Phân tích kiến trúc tổng thể của ứng dụng, mô tả chi tiết thiết kế của các thành phần chính như Tracker Server và Peer Client, cùng các giao thức giao tiếp giữa chúng.
- **Chương 4 – Hiện thực chương trình:** Trình bày quá trình hiện thực hệ thống, mô tả các đoạn mã quan trọng và giải thích cách triển khai các chức năng như xử lý đa luồng, kết nối P2P, và truyền thông real-time giữa backend và frontend.
- **Chương 5 – Kiểm thử và Kết quả:** Giới thiệu các kịch bản kiểm thử, phân tích kết quả thực nghiệm thông qua hình ảnh giao diện và log hệ thống, minh chứng cho tính chính xác và ổn định của chương trình.
- **Chương 6 – Kết luận:** Tổng kết những kết quả đạt được, đối chiếu với mục tiêu ban đầu và đề xuất hướng phát triển trong tương lai.



Chương 2: Cơ sở lý thuyết

1. Mô hình Client-Server và Peer-to-Peer (P2P)
2. Lập trình Socket và Luồng (Socket & Threading)
3. Giao thức HTTP và Cookie
4. Giao thức WebSocket



Chương 3: Phân tích và thiết kế hệ thống

1. Kiến trúc tổng thể
2. Thiết kế các Module chính
3. Thiết kế Giao thức



Chương 4: Hiện thực chương trình

1. Môi trường và Công cụ
2. Hiện thực các chức năng chính



Chương 5: Kiểm thử và kết quả

1. Kịch bản kiểm thử
2. Kết quả thực nghiệm



Chương 6: Kết luận

1. Tổng kết kết quả đạt được
2. Hạn chế
3. Hướng phát triển