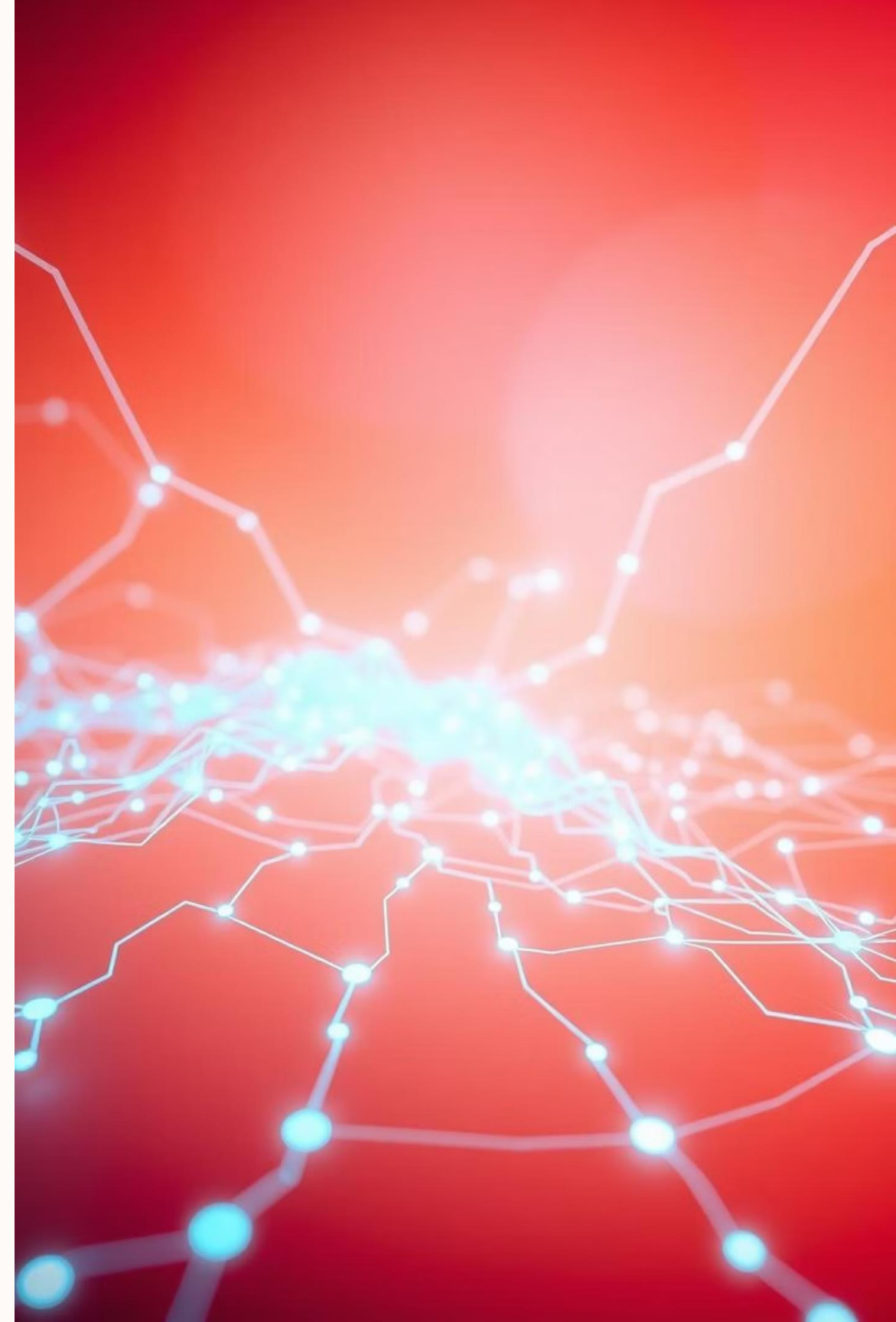


Lab01 - Kiến Thức Cơ Bản Về Client-Server

Môn học: Phát Triển Ứng Dụng Web Nâng Cao

Giảng viên: thầy Nguyễn Trọng Hiếu

Nhóm 22



CẤU TRÚC NHÓM

Giới Thiệu Đội Ngũ

1

Nguyễn Văn Anh Đức

2

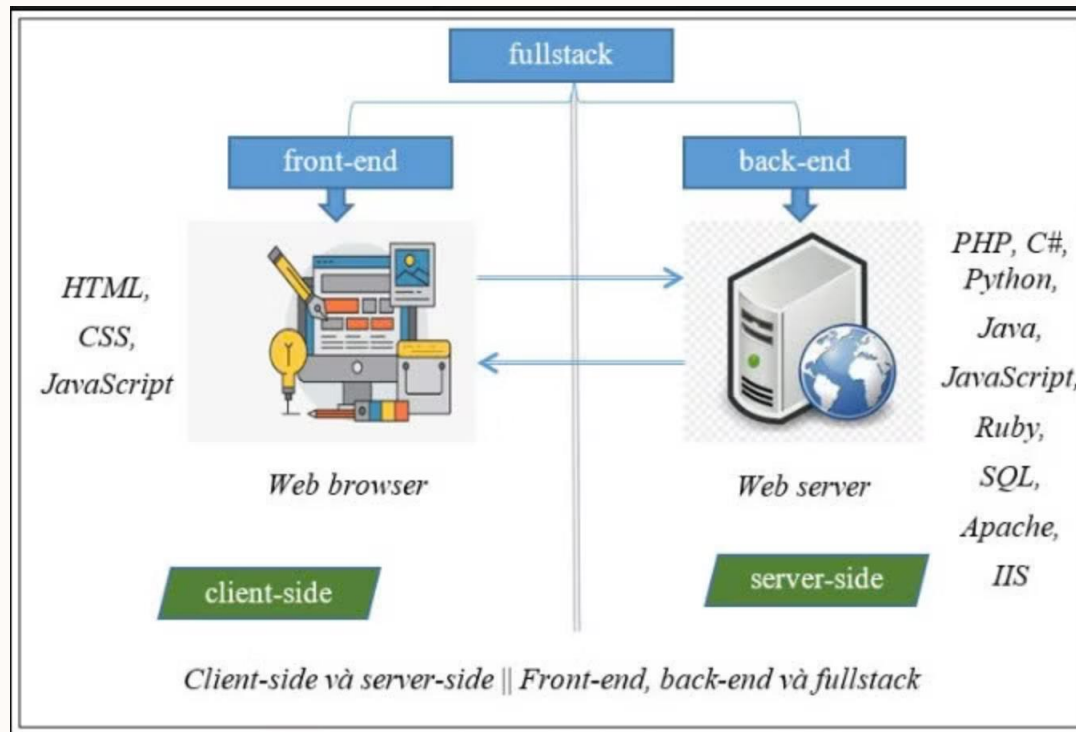
Phan Ngọc Vỹ

3

Nguyễn Hồ Đức Huy

TỔNG QUAN

Kiến Trúc Hệ Thống Client-Server



Các thành phần cốt lõi:

- **Frontend:** Giao diện web tương tác
- **Backend:** Server Express.js với API endpoints
- **Client HTTP Tùy Chỉnh:** Phát triển bằng Node.js
- **Giám Sát Mạng:** Theo dõi hiệu suất thời gian thực
- **Tích Hợp Bên Ngoài:** GitHub API, JSONPlaceholder

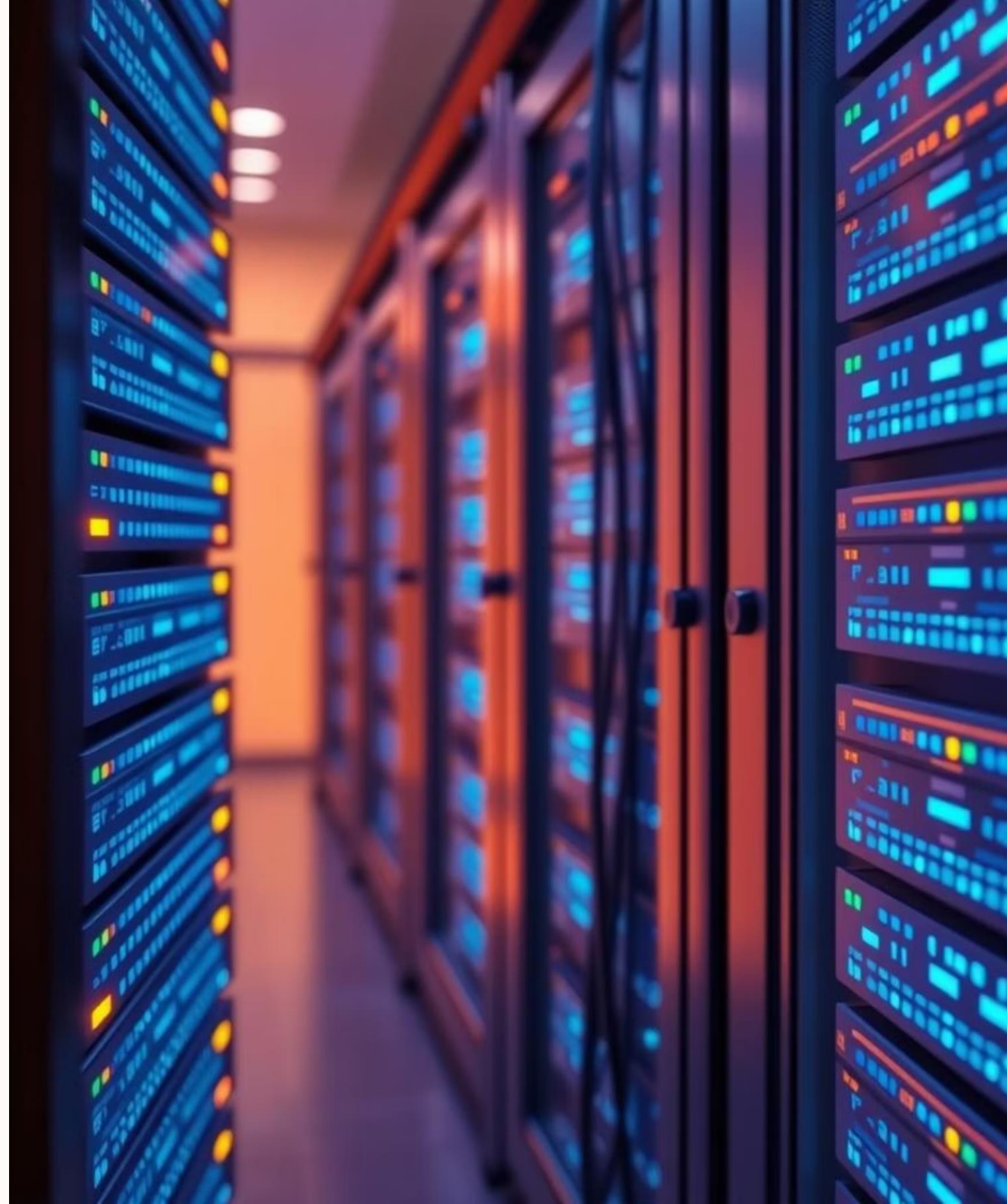
PHÍA SERVER

Triển Khai HTTP Server

```
// Express.js setupconst app =  
express();app.use(express.static('public'));  
// Custom headers middlewareapp.use((req, res,  
next) => {    res.header('X-Powered-By',  
'Lab01-Server');    res.header('X-Server-  
Time', new Date().toISOString());  
next();});// API  
endpointsapp.get('/api/server-info', (req,  
res) => {    res.json({ /* server information  
*/ });});
```

Các tính năng nổi bật:

- Phục vụ file tĩnh
- RESTful API endpoints
- Custom HTTP headers
- Xử lý lỗi (404, 500)
- Middleware ghi log



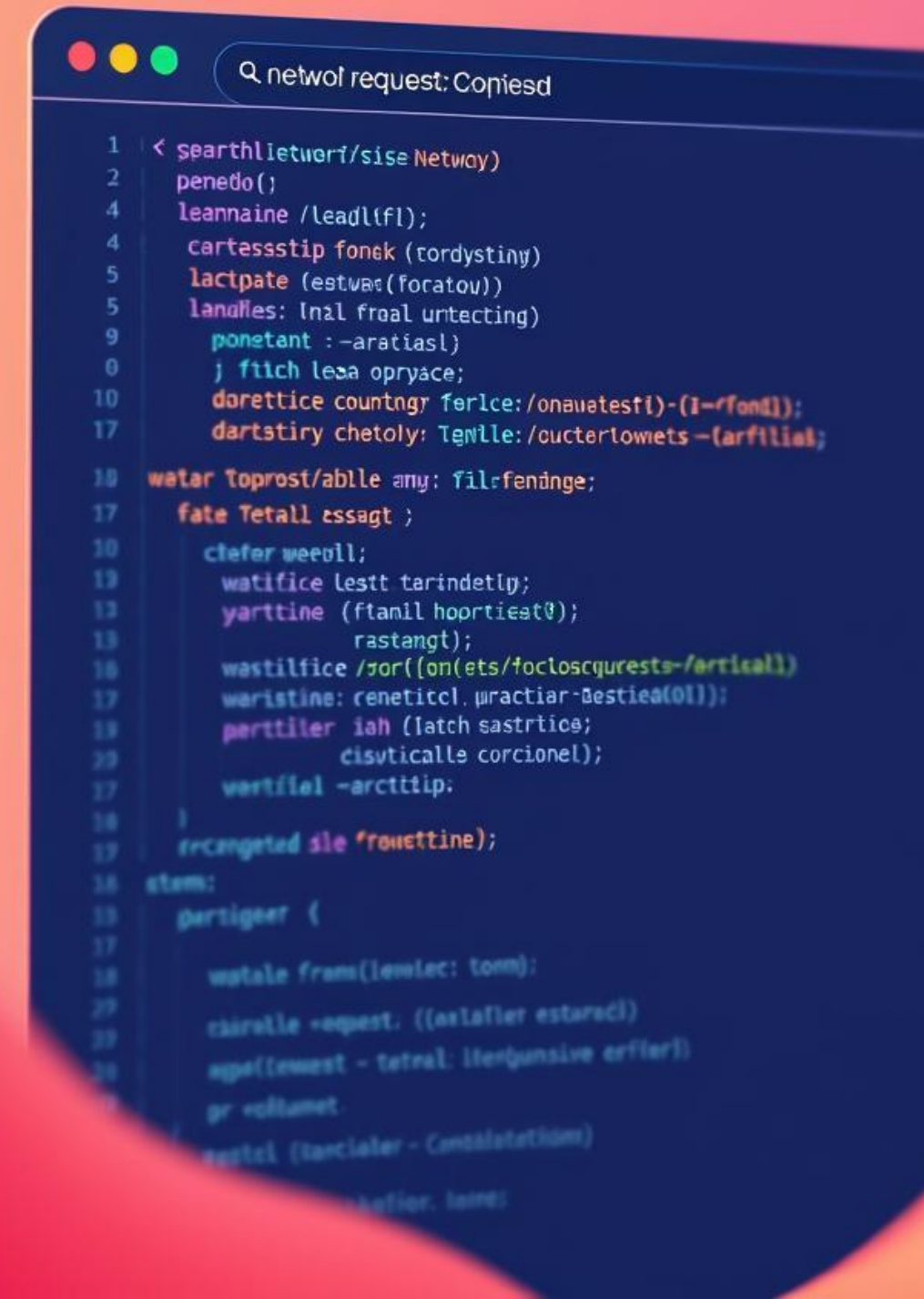
PHÍA CLIENT

Triển Khai HTTP Client

```
class HTTPClient {
  async request(options) {
    return new Promise((resolve, reject) => {
      const client = options.url.startsWith('https:') ?
        https : http;
      const req =
        client.request(options, (res) => {
          // Handle response
        });
      req.on('error', reject);
      req.end(options.data);
    });
  }
}
```

Các tính năng chính:

- Sử dụng Pure Node.js
- Hỗ trợ HTTP/HTTPS
- Phương thức GET, POST
- API dựa trên Promise
- Xử lý lỗi hiệu quả



Demo Ứng Dụng

Phía Server

```
HOÀN THÀNH TẤT CẢ KIỂM THỬ
PS D:\VS2022\Project\UDWNC\BTN1> node monitor.js
✓ Server is running, starting monitoring demo...

STARTING NETWORK MONITORING DEMO
=====
● Network monitoring started at 2025-08-12T02:00:40.915Z

Testing basic endpoints...
[2025-08-12T02:00:40.924Z] ✓ GET http://localhost:3000/api/health - 200 (7ms)
[2025-08-12T02:00:40.935Z] ✓ GET http://localhost:3000/api/server-info - 200 (10ms)
[2025-08-12T02:00:40.939Z] ✓ GET http://localhost:3000/api/time - 200 (4ms)

Testing POST request...
[2025-08-12T02:00:40.943Z] ✓ POST http://localhost:3000/api/echo - 200 (3ms)

Testing error handling...
[2025-08-12T02:00:40.947Z] ✗ GET http://localhost:3000/api/not-found - 404 (3ms)

PERFORMANCE BENCHMARK - FILE SIZES
=====

Testing Small File...
[2025-08-12T02:00:40.952Z] ✓ GET http://localhost:3000/ - 200 (3ms)
  ✓ Small File: 7238 bytes, 3ms
  📊 Transfer rate: 2412667 bytes/sec

Testing Medium File...
[2025-08-12T02:00:41.066Z] ✓ GET http://localhost:3000/style.css - 200 (3ms)
  ✓ Medium File: 7792 bytes, 4ms
  📊 Transfer rate: 1948000 bytes/sec

Testing Large File...
[2025-08-12T02:00:41.188Z] ✓ GET http://localhost:3000/script.js - 200 (7ms)
  ✓ Large File: 13688 bytes, 7ms
  📊 Transfer rate: 1955429 bytes/sec
```

Copilot

powershell

Demo Ứng Dụng

Phía Server

Lab01 - Client-Server Fundamentals

Bài thực hành về kiến trúc Client-Server và HTTP Protocol

i

Server Status

Status:

Online

Current Time:

17:03:30
12/08/2025

Uptime:

0h 3m 10s

Platform:

win32 x64

Refresh Server Info

Server Information

Server Information:

Timestamp: 2025-08-12T10:04:13.468Z

Server: Express.js v1.0.0

Uptime: 0h 4m 28s

Platform: win32 x64

Node.js: v22.14.0

Total Memory: 16063 MB

Free Memory: 5893 MB

CPU Cores: 8

Hostname: DESKTOP-7OC1TKJ

Network Interfaces: Wi-Fi, Loopback Pseudo-Interface 1

Get Server Info

API Testing

API Endpoint:

GET /api/server-info

Test API

Response:

Status: 200 OK
Response Time: 15ms
Content-Type: application/json; charset=utf-8

Headers:
connection: keep-alive
content-length: 306
content-type: application/json; charset=utf-8
date: Tue, 12 Aug 2025 10:04:51 GMT
etag: W/"132-idfbdGNfA6SE4Ny25NNSZ1plWgQ"
keep-alive: timeout=5
x-node-version: v22.14.0
x-powered-by: Lab01-Server
x-server-time: 2025-08-12T10:04:51.262Z

Response Data:
{

Network Monitor

Start Monitor

Stop Monitor

Clear Log

Total Requests: 3

Successful: 3

Failed: 0

Avg Response Time: 9ms

Network Log:

Network monitoring started at 17:04:48 12/08/2025

[17:04:51 12/08/2025] GET /api/server-info - 200 (15ms)

[17:07:18 12/08/2025] GET /api/health - 200 (6ms)

[17:07:44 12/08/2025] GET /api/time - 200 (6ms)

Những Bài Học Quan Trọng

Hiểu Sâu HTTP Protocol

Nắm vững cơ chế hoạt động của HTTP ở tầng thấp nhất.

Thiết Kế Client-Server Hiệu Quả

Xây dựng kiến trúc mạnh mẽ và phân tán.

Lập Trình Bất Đồng Bộ

Thực hành với Promise và async/await.

Phân Tích Hiệu Suất Mạng

Sử dụng công cụ giám sát để tối ưu hóa.

Xử Lý Lỗi Toàn Diện

Nâng cao độ ổn định và tin cậy của ứng dụng.

PHÁT TRIỂN TIẾP THEO

Các Cải Tiến Tương Lai



HTTPS & Chứng Chỉ

Tăng cường bảo mật kết nối.



WebSocket

Giao tiếp thời gian thực.



JWT Authentication

Xác thực người dùng an toàn.



Dashboard Phân Tích

Hiển thị số liệu chi tiết.



Tối Ưu Hiệu Năng

Nâng cao tốc độ xử lý.

TỔNG KẾT

Thành Tựu & Hướng Đi

Thành Tựu Đạt Được

- Hoàn thiện Client-Server
- Đạt mọi yêu cầu đề ra
- Triển khai tính năng nâng cao
- Tài liệu hóa chi tiết

Insight Quan Trọng

- Sức mạnh của HTTP
- Tầm quan trọng xử lý lỗi
- Lợi ích giám sát hiệu suất
- Thực hành phát triển web hiện đại