

## Hướng dẫn nhiệm vụ chi tiết cho từng thành viên – Assignment 1

Dưới đây là phân chia chi tiết 5 file có #TODO trong project HTTP Server & Chat App cho 4 thành viên. Mỗi người phụ trách một nhóm file hoặc logic liên quan, đảm bảo mỗi phần có thể lập trình và test độc lập trước khi tích hợp.

### Khang – Proxy & Backend (Socket layer)

\*\*File phụ trách:\*\* **backend.py**, phần multi-thread trong **proxy.py**

- Hoàn thiện hàm run\_backend(): thêm vòng lặp accept và tạo thread mới cho mỗi client.
- Hoàn thiện hàm run\_proxy(): tương tự backend, mỗi client chạy trong 1 thread.
- Đảm bảo mỗi thread gọi handle\_client(ip, port, conn, addr, routes).
- Kiểm tra kết nối ổn định, không treo khi nhiều client song song.
- Thêm log hiển thị IP client và trạng thái kết nối.
- Test: chạy backend và proxy, truy cập qua curl http://127.0.0.1:8080/

### Khoa – Routing & Hook logic

\*\*File phụ trách:\*\* **proxy.py** (resolve\_routing\_policy), **httpadapter.py**

- Sửa resolve\_routing\_policy() trong proxy.py: fallback host/port khi không tìm thấy hostname.
- Xử lý nhiều proxy\_pass bằng chính sách round-robin hoặc least-conn nếu có.
- Trong HttpAdapter.handle\_client(): thực thi req.hook(headers, body) nếu có.
- Gắn kết quả trả về của hook vào Response trước khi gửi lại cho client.
- Hoàn thiện build\_proxy\_headers(): nếu có authentication thật, thêm header Proxy-Authorization.
- Test: gửi request tới các route của WeApRous (/login, /hello) và kiểm tra log hook.

### Nhiên – HTTP Request layer

\*\*File phụ trách:\*\* **request.py**

- Hoàn thiện prepare(): parse request line, headers (lower-case) và cookie.
- Tạo thuộc tính req.cookies = {k: v} từ header Cookie.
- Nếu Cookie có auth=true → req.isAuthorized = True, ngược lại False.
- Gắn req.hook = routes.get((METHOD, PATH)) nếu có WeApRous routes.

- Hoàn thiện `prepare_body()`, `prepare_content_length()`, `prepare_auth()` (nếu cần).
- Test: gửi request thủ công bằng netcat hoặc curl, kiểm tra in ra [Request] log đúng.

## Khánh – HTTP Response layer

\*\*File phụ trách:\*\* **response.py**

- Hoàn thiện `build_content()`: đọc file từ `base_dir`, trả (`length, bytes`). Nếu không tồn tại → trả 404.
- Hoàn thiện `build_response_header()`: tạo header HTTP/1.1 chuẩn (`Status, Content-Type, Length, Date, Connection`).
- Mở rộng `prepare_content_type()` để xử lý MIME khác (`application/json, xml, zip, video, ...`).
- Trong `build_response()`: nếu `request.is_authorized == False` thì trả 401 Unauthorized, ngược lại serve file.
- Test: curl -i http://127.0.0.1:9000/ với và không có Cookie auth=true để kiểm tra 401/200.

## Tích hợp & Kiểm thử toàn nhóm

- A khởi chạy backend (9000) và proxy (8080).
  - B đảm bảo proxy map đúng host → backend và hook gọi app thành công.
  - C xác thực cookie hoạt động đúng (`auth=true` mới vào được /).
  - D đảm bảo response đúng file và header chuẩn.
- 5. Test tích hợp: python `start_backend.py`, python `start_proxy.py`, python `start_sampleapp.py`, sau đó curl qua 8080.