

Balanced Webserv Project Work Split

Person A Core Networking & GET/DELETE

Focus: Server socket setup, client handling, and core HTTP functionality

Responsibilities:

- Set up sockets, bind(), listen(), and accept() logic
- Configure non-blocking mode and manage I/O via poll() (or equivalent)
- Implement the full GET method (static file serving, headers, status codes)
- Implement the DELETE method (file deletion with proper checks)
- Validate methods against allowed list (405 Method Not Allowed)
- Send correct HTTP status codes and default error pages
- Ensure connections are resilient and timeout properly
- Help test GET/DELETE edge cases and invalid requests

Effort Level: Moderate to High

Why it's fair: These tasks involve deep but focused C++/networking logic and match one domain.

Person B Configuration System & Routing

Focus: Parsing the configuration file and routing requests based on it

Responsibilities:

- Parse NGINX-style config file (host/port, routes, error pages, body limits, etc.)
- Setup per-route rules: allowed methods, root paths, redirections, upload dirs
- Assign default server per host/port pair
- Handle directory listing toggle and default file logic

- Communicate method restrictions to Person A for enforcement
- Handle parsing robustness and error reporting for bad configs
- Provide complete working config examples and documentation

Effort Level: Moderate

Why it's fair: This is mostly parsing + config logic, no low-level IO. Complexity lies in variety, not depth.

Person C POST, Uploads & CGI

Focus: Upload handling, CGI execution, and POST logic

Responsibilities:

- Fully implement the POST method for file uploads and CGI
- Save uploaded files to the right directory based on config
- Integrate CGI execution (run .php, .py, etc.)
 - Setup proper environment (PATH_INFO, working dir)
 - Handle chunked request body parsing
 - Handle CGI output and EOF detection
- Work on CGI fallback behavior (e.g. missing Content-Length)
- Apply Mac-specific flags (fcntl) if needed
- Help stress-test uploads and CGI concurrency

Effort Level: Moderate to High

Why it's fair: POST & CGI are trickier, but self-contained and high-impact. It balances Person As I/O depth with feature depth.

Shared Team Tasks

- Makefile (all participate must be non-relinking and complete)
- Testing (write curl scripts, browser tests, telnet sessions)
- Documentation (each documents their part; combine in README)
- Resilience (no crashing, blocking, or resource leaks)