Subject: Re: Book idea: U++ Web development
Posted by BetoValle on Mon, 18 May 2026 15:13:30 GMT
View Forum Message <> Reply to Message

Hi,

I definitely think such a book would fill an important gap in the U++ ecosystem, especially for developers building real production web applications.

One suggestion from my own experience running a Skylark SaaS in production on a Debian VPS for the last years: please dedicate strong attention to multithreading, object lifetime, RAII and thread-safe database access patterns.

Coming from Delphi and Java, one of the hardest parts for me was understanding how shared SQL sessions and global objects behave in a native multithreaded environment. After redesigning my database handling to be request/thread-local and ensuring proper destruction scope, the server became dramatically more stable.

I would strongly suggest a dedicated production-oriented chapter focused on REAL-WORLD PROBLEMS that developers actually face after deployment, for example:

* Common production mistakes
* Why native web servers crash
* Threading pitfalls
* Shared state problems
* Global object misuse
* Database session lifetime
* Request isolation
* Race conditions
* Debugging random crashes
* Memory ownership and destruction timing

This kind of knowledge is extremely valuable because many crashes are not caused by the framework itself, but by subtle concurrency and lifetime mistakes that are very difficult for newcomers to diagnose.

Most modern web frameworks hide these details, but native C++ web development exposes them directly, and understanding them is essential for building stable long-running servers.

I also think chapters about:

* production deployment
* watchdogs / health checks
* nginx reverse proxy
* logging
* scaling strategies
* containerization

would be extremely valuable.

Another very interesting perspective could be:
"From Delphi/Java/PHP to U++ web development"

because many developers interested in U++ are not necessarily traditional C++ experts.

I truly believe U++/Skylark deserves more visibility as a lightweight native web platform.

Thanks!
