Good code solves only existing issues, without extra universality, doesn't contain anti-patterns and covered by tests to be able to evolve.

Full-stack programmer.
Rust - since 2015,
Angular - since 2011,
PHP - since 2004.

Technologies, apps and services I have experience with:
Rust, TypeScript, JavaScript, PHP; Angular, Ionic, Karma, Protractor, Twitter Bootstrap; MySQL, PostgreSQL, Redis; Cloudflare, Mandrill, PayPal, Stripe, Nexmo; GAE, AWS: EC2, S3, Cloudfront, Pipeline, CodeCommit, RDS; Jira, YouTrack, Git, Travis CI; SOA, REST, SOLID...

I prefer to solve mathematically-challenging tasks (love to apply math to solve real-world issues).
I follow best practices and my years of experience to design apps architecture clean, scalable, easy for collaboration and code reuse.

Have a lot of experience with caching (also with mutexes, semaphores, preventing dog-pile and race condition effects, RAII control of resources).

My current achievement is 100% Rust REST API, serving multiple Web and mobile apps (CRM, Task Manager, mobile apps, websites).
Have experience of writing CRM, ERP and WMS for big e-commerce and small startups (PHP, Rust).

In Angular, I can write directives (components) of any level of complexity - including interactive maps (Angular+D3.js), nested trees, SVG elements and other fancy things.
My components are always reusable. I write e2e and unit tests.
Prefer to write frontend code in TypeScript.

I have experience with Ionic, I've created multiple mobile apps with this framework, including sophisticated WMS (Warehouse Management System), consisting of mobile apps (scanning, instructions, price stickers printing, items movement, delivery scheduling).

My favorite books are Clean code, Patterns of Enterprise Application Architecture.
I like and respect SOLID principles, especially the Separation of Concerns.