CP, I'm primarily a .net based developer working with GitHub Spec kit. I need to create a constitution. Here are some things I care about: code quality, testing standards, user experience consistency, security and performance. Can you help me write a prompt for spec kit to write a constitution for me?

Absolutely, Jake. Since you're working with GitHub's Spec Kit and want to generate a constitution that enforces high standards across your development workflow, here's a tailored prompt you can feed into Spec Kit to generate a solid foundation:

**Prompt for GitHub Spec Kit Constitution Generation:**

Create a constitution for a .NET-based software project that enforces high standards across the following domains:

* **Code Quality**: Enforce clean architecture, SOLID principles, and consistent naming conventions.
* **Testing Standards**: Require unit, integration, and regression testing with minimum coverage thresholds and CI enforcement.
* **User Experience Consistency**: Define UX guidelines for layout, accessibility, responsiveness, and interaction patterns across platforms.
* **Security**: Mandate secure coding practices, regular vulnerability scans, and dependency audits.
* **Performance**: Set benchmarks for load times, memory usage, and scalability, with profiling requirements during development.

The constitution should be modular, enforceable via GitHub workflows, and compatible with Spec Kit's policy enforcement model. Include examples of how each principle can be validated or automated.

Would you like to add governance rules (e.g., who can approve changes, how exceptions are handled), or tailor this for a specific team structure or repo layout? I can help you extend this into a full Spec Kit config or even generate sample enforcement specs.

That was very well done. thank you.

You're very welcome, Jake — I’m glad it hit the mark. You’ve got a sharp vision for quality and structure, and Spec Kit is a great match for codifying that. If you want help drafting enforcement specs, CI workflows, or even modular policy templates for each domain (like UX or security), I can help you scaffold those out too.

Want to take the next step and turn this into a living governance model for your repo?

Today

I'm at the plan stage with spec-kit and I've written a draft of my plan. Please review it, ask me question if there is anything unclear and show me how you would write it to be more comprehensive. Here is my intial plan: The application will use Microsoft Blazor, C#, JavaScript. The application needs to be responsive so that it can run on tablets, phones or normal computers. Data will be stored in Microsoft SQL server and we'll use Entity Framework as our ORM. Logging will use SeriLog and we'll include the New Relic agent for monitoring.

Thanks for sharing your draft, Jake — it’s a solid start and clearly reflects your technical direction. To help Spec Kit generate a constitution that enforces your standards, we’ll want to make the plan more comprehensive by:

* **Clarifying architectural decisions**
* **Defining quality and performance expectations**
* **Outlining enforcement mechanisms**
* **Specifying tooling and integration points**

**🔍 Questions to Refine Your Plan**

1. **Blazor Scope**: Are you using Blazor Server, WebAssembly, or Hybrid? This affects performance and hosting.
2. **JavaScript Usage**: Will JS be used for interop only, or are there standalone modules?
3. **Responsiveness**: Do you have a preferred CSS framework (e.g., Bootstrap, Tailwind) or layout strategy?
4. **Entity Framework**: Will you use EF Core with code-first or database-first? Any migration strategy?
5. **Security**: Any specific auth mechanism (e.g., ASP.NET Identity, OAuth2)? Will you enforce HTTPS, input validation, etc.?
6. **Testing**: What testing frameworks will you use (e.g., xUnit, bUnit)? Will you enforce coverage thresholds?
7. **CI/CD**: Are you planning GitHub Actions, Azure DevOps, or another pipeline?
8. **Performance Targets**: Any specific metrics for load time, API latency, or memory usage?
9. **Monitoring Scope**: Will New Relic monitor frontend, backend, or both?

**🧠 Suggested Rewrite for a More Comprehensive Spec Kit Plan**

markdown

## Project Plan for Spec Kit Constitution

### Tech Stack

- \*\*Frontend\*\*: Microsoft Blazor WebAssembly with optional JavaScript interop

- \*\*Backend\*\*: ASP.NET Core (C#)

- \*\*Database\*\*: Microsoft SQL Server with Entity Framework Core (code-first)

- \*\*Logging\*\*: SeriLog with structured logging and sinks to file and New Relic

- \*\*Monitoring\*\*: New Relic agent for backend performance and error tracking

### UX & Responsiveness

- Application must be fully responsive across desktop, tablet, and mobile devices

- Layouts will follow accessibility standards (WCAG 2.1) and consistent design patterns

- Tailwind CSS will be used for styling and layout consistency

### Code Quality

- Enforce SOLID principles, clean architecture, and consistent naming conventions

- All code must pass static analysis (e.g., Roslyn analyzers, StyleCop)

- Pull requests must include code reviews and pass linting checks

### Testing Standards

- Unit tests (xUnit), component tests (bUnit), and integration tests required

- Minimum 80% code coverage enforced via CI

- Regression tests must be run before major releases

### Security

- HTTPS enforced across all environments

- Input validation and output encoding required

- Authentication via ASP.NET Identity with optional OAuth2 integration

- Regular dependency audits and vulnerability scans

### Performance

- Initial page load under 2 seconds on mobile

- API response time under 500ms for 95th percentile

- Memory usage monitored and profiled during development

### CI/CD & Enforcement

- GitHub Actions used for CI/CD pipeline

- Constitution policies enforced via Spec Kit workflows

- Violations block merges unless explicitly overridden by designated maintainers

Would you like me to help scaffold the Spec Kit YAML or policy modules next? We can modularize this into enforceable specs for each domain (e.g., code-quality.yml, testing.yml, security.yml).

End of day – 2025-09-29