JSON Schema

Build more. Break less. Empower others.

JSON Schema

Google Summer of Code 2025

Project Information:

Name: Comprehensive JSON Schema linting for encouraging best practices and catching anti-patterns early (link)
Length: 350 Hours (Large)
Mentor: @jviotti (Juan Cruz Viotti)
Assisting Mentor: @Relequestual (Ben Hutton)

Student Details:

Full Name: Karan Shailesh Palan
Email: karanpalan007@gmail.com
GitHub: karan-palan
LinkedIn: karan-palan-476472289/
Twitter/X: Karan_Palan7
JSON Schema Slack ID: Karan Palan
Preferred Language: I'm proficient in English for communication, both spoken and written
Location: Mumbai (Maharashtra), India
Time Zone: Indian Standard Time (IST) (UTC +05:30)
Phone Number: +91 8928670471
Institution: Manipal University Jaipur
Program: B.Tech in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning
Stage of completion: Sophomore Year (expected graduation June 2027)

About Me:

I've been using open-source software for as long as I can remember—long before I even knew what "open-source" meant. Whether it was watching videos with VLC Media Player or browsing the web with Mozilla, I've always appreciated the power of community-driven software. Over time, I've made it a point to contribute to open-source in any way I can.

Before entering college, I had no prior knowledge of Computer Science or programming. However, I committed to learning consistently every day, which not only helped me build a strong foundation but also developed my ability to pick up new technologies quickly. Initially, I explored competitive programming but soon realized my true interests lay elsewhere. In my first semester, I dived into Web Development and Machine Learning, working on some exciting projects that deepened my passion for

building and learning.

My open-source journey began as a Freshman with Hacktoberfest 23', where I got hands-on experience with Git, GitHub, and the entire open-source workflow—from writing pull requests to creating and managing issues. By the end of the event, I had 7 PRs merged into an open-source online C compiler, which I shared in this LinkedIn post. I was also a contributor at SWOC 23'. I am also an MLSA (Microsoft Student Learn Ambassador) and a Postman API student expert. In my second semester, I actively started freelancing and participating in hackathons—both through MLH and various colleges—and had the incredible experience of winning 5 of them! (Tweet).

My open-source journey deepened when I started contributing to Sugar-Labs and became involved in Music-Blocks-v4, a complete rewrite of Music-blocks using modern tooling and frameworks for better future maintainability. Encouraged by a mentor, I submitted a GSoC proposal—and got accepted!

Beyond open-source, I also interned at my hostel, Sundarone, where I built their website and collaborated on developing a mobile application to enhance their services.

Since then, I've remained actively involved in open-source, becoming a member of Sugar-Labs and Layer5, while also contributing to various projects of the JSON Schema community and Chromium. Open-source has been a core part of my journey, and I'm always eager to learn, build, and give back to the community. I've delivered talks on open-source, mentored at hackathons, and guided juniors in kick starting their coding journeys, helping them navigate the world of development with confidence.

I've also been actively sharing my journey and learnings in public. Here are some of my posts:

1. GSoC experience and guidance blog
2. Promises explained – A made a thread about promises in JavaScript that Devs loved!
3. Hacktoberfest Experience – Made a thread about my journey of navigating through open source.
4. 30 days of Code – I learned a lot in this period as I didn't have any exams in college.
5. An update on 4 months of learning to code (Link).

A lot more stuff can be found on my socials!

My general interest lies in building creative and user-centric applications that can solve real-world problems. I am well aware of:

Programming Languages: JavaScript, TypeScript, Solidity, Python, C++, Rust
Libraries/Frameworks: React, Zustand, Recoil, Sass, BootStrap, TailwindCSS, Jest, Node.js, ExpressJs, Honojs, Nextjs Electronjs, Tensorflow, CosmWasm
Databases: MongoDB, PostgreSQL, Redis
Tools/Platforms: Git, GitHub, Gerrit, Vercel, AWS, VS Code, Postman, Docker, Kubernetes, MATLAB

Work Experience:

1. **Member @Sugarlabs, @Layer5** *Present*
   Nextjs, MaterialUI, Reactjs, Typescript, Rjsf, Docker, Kubernetes
   - Invited to Layer5 after consistent contributions to Sistent, Meshery, and the Layer5 and Meshery websites. After becoming a member, I also contributed to Cloud, working on bug fixes, feature improvements, and co-hosting community meetings.
   - Working on development of Workspace in MusicBlocks v-4 project and guiding new contributors to understand the project better (discussion).

2. **Google Summer of Code 24' @Sugarlabs** *May-Sep 2024*
   React, Typescript, Vite, Zustand
   - Developed the Masonry Framework using React and TypeScript. (Project Page | Final Report | Code)
   - As a prototype project, significant time was dedicated to discussions and agreements within the community (discussion).
   - Implemented dynamic SVG rendering, collision detection using Quadtree, and drag-and-drop functionalities for graphical blocks.

3. **SDE Intern** at **Sundarone Private Limited**
   Feb-May 2024
   - Developed and maintained the Sundarone Hostels website and mobile app and led continuous iteration cycles, incorporating user feedback to enhance UI/UX and functionality, resulting in a 20% increase in user retention.
   - Integrated a payment gateway (Razorpay) into the mobile app, enabling secure and efficient payments, leading to a 15% boost in bookings.
   - Collaborated with backend developers to integrate APIs, enhancing booking and user account management features.

Previous Open source Projects around collaboration:

I have always followed a project-based approach whilst learning any new technology. Thus, I have built a lot of projects. Here are some of them:

1. **Arcadia Ecosystem**
   Websites - React, Tailwind, Contracts - Solidity, Move etc (track-specific).
   Arcadia | Arcadia.pump | Zenith Wallet | Arcadia.slcm
   - Built two projects from scratch and expanded two previous projects to form a unified ecosystem for different hackathon tracks.
   - Arcadia is an AI-NFT generation platform, and Arcadia.pump is a clone of pump.fun on the *Aptos Blockchain*. This project won its track in the hackathon.
   - Zenith Wallet is UPI like wallet and Arcadia.slcm is implementation of college attendance portal in *Educhain*.

2. **SightAi**
   Website - Reactjs, TailwindCSS, ML model - Tensorflow, Coco-api etc.
   Deployed Website | ML model | Demo
   - This Project helped us to qualify for *round 3 of the Microsoft Imagine Cup*. This project empowers the Blind to be more independent with the help of AI.
   - Made an ML model with a teammate using Tensorflow, Yolo, Coco-API, and Pytorch which gives an accuracy of 90-95 % for static objects and 60-75% for moving objects. Developed the complete landing page for our product in under 5 hours using React and TailwindCSS.

3. **CHAEN COLLECTIVE**
   HTML, CSS(Bootstrap), Javascript, Aeternity Javascript SDK
   Deployed Website | Code | Demo
   - We won *First Place* in the *Aeternity track* of a Hackathon with 800+ participants and had a meeting with the founder about scaling the project.LinkedIn post
   - Created a DAO using the Aeternity blockchain, learned the Sophia language, and deployed the chain to Testnet.

4. **NEXUS SPHERE**
   React, Solidity, Infura, Web3js, Truffle, Ganache-cli, Tailwind, Rust, CosmWasm
   Github | Demo | Pitch
   - Made this for the Soonami Ventureathon, got shortlisted in 2 rounds and *ranked 11th across the world.*
   - This involves experience with voting mechanisms, proposal creation and evaluation, tokenomics, and overall strategic decision-making within decentralized organizations. We also created an extension of this project Nexus-Estate, which involved tokenization of parts of real-estate properties.

These projects helped me understand how to lead a team, collaborate, and build great stuff with other people.
I also built some projects in a few hours to learn a particular technology like Decendrive to learn solidity, Insight-Isle, Ai Paragraph summarizer and a simple docs clone to learn the MERN stack and am currently building a Youtube clone to learn Devops (video processing, CI-CD pipeline etc.) and a Terminal based Fps game in c++.

Why JSON Schema?:

I first got to know about **JSON Schema Org** through my friend, Dhairya Majmudar (now a part of the website triage team). He shared his experiences working with JSON Schema, and his story intrigued me. I decided to explore it further and started my journey by contributing to the JSON Schema website.

As I became more familiar with JSON Schema, I realized I had already used it in my GSoC project with Sugar Labs while working on Music Blocks v4 lib. Later, while working with Meshery Schemas, I encountered JSON Schema once again—this time in a more infrastructure-focused setting. As Meshery models infrastructure components, I saw firsthand how standardized schema definitions played a key role in ensuring interoperability between diverse systems. JSON Schema emerged as the perfect solution, offering a declarative methodology to validate, document, and structure JSON-based data models, seamlessly fitting into our use case. My experience working with Meshery's constructs allowed me to experiment with schemas, helping enforce uniformity in model validation.

However, I also recognized that JSON Schema is both easy and hard at the same time. While its declarative approach simplifies schema definition, real-world implementations can introduce complex challenges, especially with schema versioning and compatibility issues. At Layer5, I saw developers struggle when integrating models, such as connecting AWS CloudTrail to other systems, due to differences in schema draft

references. This fragmentation made it difficult to maintain consistency across various tools and infrastructures.

This is where the JSON Schema linting project can have a profound impact. The project aims to finalize comprehensive linting rules that help developers identify schema inconsistencies, enforce best practices, and ensure compatibility across different JSON Schema drafts. By providing automated validation for some rules and guidance, it will streamline schema development, reduce errors, and enhance interoperability between platforms and services.

Now, through GSoC, I am excited about the opportunity to contribute directly to JSON Schema—enhancing its ecosystem, improving linting capabilities, and ensuring that developers worldwide can leverage it to build more scalable, robust, and standardized applications.

Here is a detailed summary of the work and contributions that I did before the GSoC proposal period to JSON Schema Org and related tooling:

Issues and Pull Requests:

1. JSON Schema Organisation (https://github.com/json-schema-org/)
   Issues created: 13 Issues created, 8 accepted, 4 under Triage process

1.	Merged	Fix: Consistent Card Heights on Ambassadors Page and Update Twitter Logo to X #1256
2.	Merged	Fix: Ignore Brackets Inside Strings in getPartsOfArrayContent #1268
3.	Merged	Fix: correct vertical alignment issue in schema example #1297
4.	Merged	Fix : pro help sidebar alignment and add buttons #1310
5.	Merged	Fix[dark-mode-toggle]: fix dark/light mode toggle on mobile #1332
6.	Merged	Fix - markdown render in tools page #1351
7.	Merged	Feat[tools]: add micronaut json schema as tooling #1356
8.	RTM	Fix: arrow svg compatible with dark mode and responsiveness of text #1468
9.	WIP	- Enhancement: Review the implementation of the JSON editor to better management of state/style #560 (comment) - Enhancement: Special sidebar behaviour for pages without sidebar item #1011

10.	Merged	Remove additional gmail link #863
11.	WIP	Enhancement: Add TSC Governance Details to /Community Page 1344 (comment)

2. Sourcemeta (https://github.com/sourcemeta/)

1.	WIP	Feat: Add --path Option to Target Specific Schema Parts in validate Command #220
2.	Merged	Fix(cli): ensure unknown commands other than help return exit code 1 #230
3.	Merged	Feat: add --json to lint command #228
4.	Merged	Refactor read_json to accept optional callback and update related tests and functions #1547

3. Hyperjump (https://github.com/hyperjump-io/)

1.	Merged	Make the website mobile responsive #16

Through my consistent code contributions, I have developed a comprehensive understanding of the contributing guidelines, build procedures, and intricacies within the different codebases. This experience has endowed me with the confidence to swiftly identify the pertinent files requiring modification or integration of new components. Whether addressing bugs or implementing new features, I am adept at navigating the codebase and effectuating precise solutions.

Presence in the Community:

After spending 2 weeks as a contributor in the community, I started to look for various JSON Schema tooling to contribute to (thread). While exploring JSON Schema validation tools, I found an outdated VS Code extension that was poorly rated, used ajv and only supported Draft 4. This made it unusable for modern JSON Schema workflows. Seeing this gap, I decided to build a better tool from scratch, leveraging the JSON Schema CLI for validation and linting as suggested by mentor Juan.
To make my extension more effective, I needed structured output from the CLI. However, the *lint* command lacked JSON output support, making it difficult to integrate with VS Code. To fix this, I contributed to the cli by *adding a --json flag to the lint command* (PR #228). (discussion). Here is the link to the extension -

jsonschema-blaze-vscode and this PR showcases the linting capabilities of the extension. (~100 downloads)
I was new to C++-based contributions, as my prior experience was purely academic. However, Juan was incredibly helpful—guiding me through the codebase, best practices, and contribution process. Thanks to his mentorship, I've grown confident in making meaningful C++ contributions and am eager to continue contributing to Sourcemeta's repositories beyond GSoC.

I've also been involved within the community and have regular participation in Open community Working meetings and GSoC specific calls where I got to speak with the mentors and clear my doubts.
As an open-source contributor and maintainer, I prioritize assisting new contributors to engage with the project's issues. I have helped other developers set up the project and make contributions to GitHub. Apart from these, I have had in-depth conversations about the project with my mentors Juan and Ben.

Qualification Task:

Github pages - https://karan-palan.github.io/JSON-Schema-Linting-Qualification-Task/
Drive PDF link - Here
Notion - Here

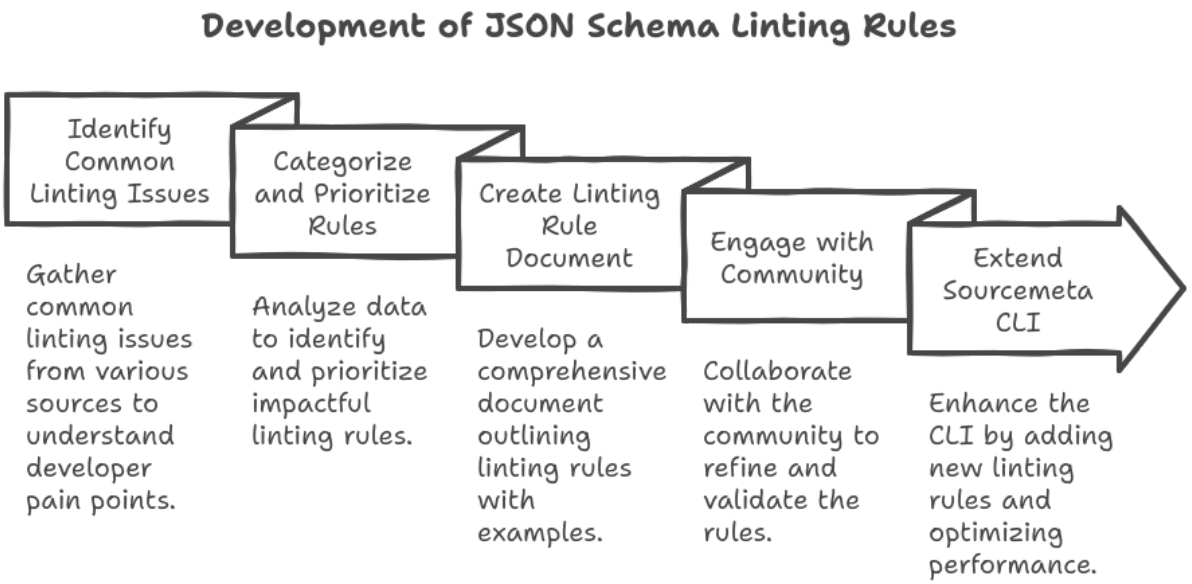
Project Details:

The project aims to improve the JSON Schema Linting by focusing on defining, standardizing rules after reaching consensus on the anti-patterns and best practices across dialects. The goal is to publish each rule with a stable URL, allowing any linter to link to them when reporting failures on the JSON Schema website. Additionally, the project will involve the implementation of these linting rules by extending the Sourcemeta JSON Schema CLI and other tooling.

Although JSON Schema linting has been a focus of prior efforts—such as those documented in https://github.com/json-schema-org/json-schema-linting, Issue #1079, and Discussion 323, this project will take those foundational contributions and refine them into a more cohesive, user-friendly solution. By building a comprehensive and scalable linting framework, we aim to provide developers with a clear, consistent set of rules that can be universally applied across JSON Schema projects. This will not only improve the overall quality and reliability of schemas but will also help developers adhere more closely to best practices and avoid common mistakes. With the implementation of standardized rules, the linting process will be automated and easily applicable at scale, making it easier for developers to ensure their schemas are error-free and production-ready. Through this initiative, the project

will foster collaboration within the JSON Schema community and contribute to the widespread adoption of JSON Schema in various development environments. It aims to improve the consistency, maintainability, and usability of JSON Schema documents, ensuring that developers adhere to best practices and avoid common pitfalls.

Deliverables:



Here are the major objectives on which I want to work as a part of Google Summer of Code 2025.

1. **Research and Identification of Common Linting Issues:** To create a set of linting rules based on common issues, ensuring that the rules align with official JSON Schema standards.

Implementation:

- Source Collection: Gather common linting issues from StackOverflow, Slack, and GitHub discussions to understand the pain points developers face.
- Discussions with community: Reach agreement on what the anti-patterns and best practices are across dialects.
- Rule Categorization: Analyze the data collected to identify the most common

and impactful linting rules, categorizing and prioritizing them based on their frequency and severity.

2. **Creation of Linting Rule Document:** To create a document implementing rules based on consensus on the anti-patterns and best practices across dialects, ensuring that the rules align with official JSON Schema standards.

Implementation:

- Document creation: Create a style guide or advanced document on the JSON Schema website where each rule is published with a stable URL. This document will serve as a reference for developers and linters and will outline each linting rule with examples, explanations, and the reasoning behind their selection. Some guidance on common cases will be documented on https://www.learnjsonschema.com/2020-12/ as “common pitfalls” sections.
- Community Engagement: Engage with the community to ensure that the rules represent the best practices and widely accepted standards. Gather feedback and iterate the document to incorporate any missing rules or clarification needed from the community.
- Linting test suite: We may also maintain a public structured collection of JSON Schemas that demonstrate common anti-patterns and their corrected forms taking inspiration from the JSON Schema Test Suite. This can serve as a valuable resource for contributors, linter implementers, and tooling authors to understand the rationale and autofix behavior for each rule.

3. **Extending the Sourcemeta JSON Schema CLI:** To enhance the functionality and performance of the cli by adding autofix to lint command for rules decided in the document above.

Implementation:

- Add New Linting Rules: Expand the existing linting framework to include new rules based on the document with all the rules. This includes rules like: Enforcing the $schema keyword at the root level of schemas, replacing single-value enum with const for clarity etc.
- Optimize Performance: Enhancing the parsing and validation process to handle larger schema files efficiently.

Other objectives on which I'll work as a part of GSoC 2025 are:

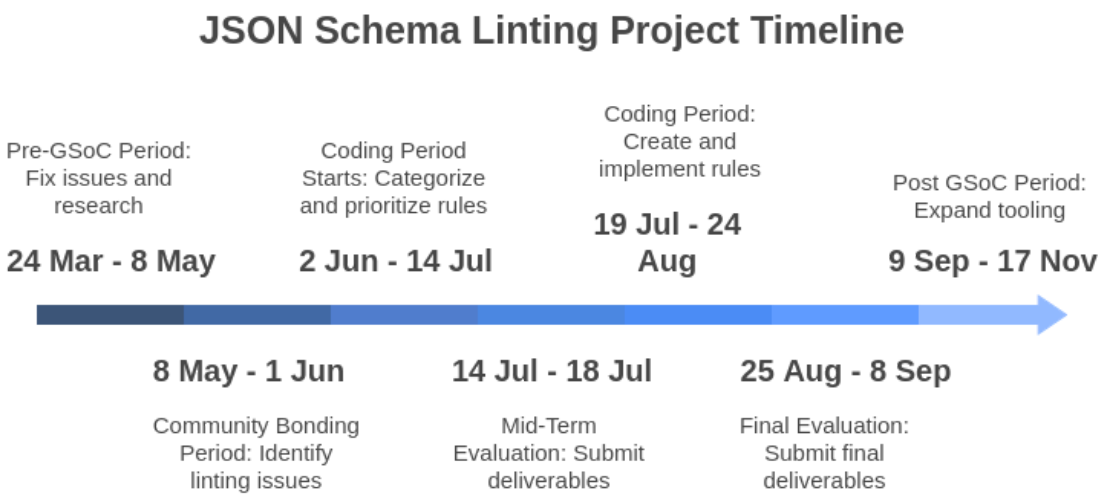
1. Extending other tooling: Help Implement the rules defined in other open source tooling.
2. Fixing bugs in JSON Schema community repositories: Fix bugs and implement features to various JSON Schema community repostories and collaborate with others.
3. Collaborating with the Canonicalization Project (#857): Many linting rules also serve as normalization rules. I plan to collaborate with the canonicalization initiative to ensure that the rule engine is reusable across both linting and normalization pipelines. This could enable unified tooling where the same rule definitions can drive schema diagnostics (`--lint`) and automatic cleanup (`--normalize`), reducing maintenance overhead and improving schema quality across the ecosystem.

Impact on JSON Schema:

The JSON Schema organization aims to standardize and validate the structure of JSON data, making it essential for maintaining the integrity and consistency of data-driven applications. By improving the linting system, this project will significantly enhance the user experience for developers working with JSON Schema. The implementation of standardized linting rules will help detect common mistakes and enforce best practices, ensuring that schemas are consistent and reliable across various projects.

Post implementation of the decided rules, the linter can be applied at scale, to thousands of schemas, cleaning them all at once without user intervention (exposed as a publicly available API. It will help developers avoid common pitfalls, such as the improper use of validation keywords and the inclusion of deprecated properties, thereby reducing the risk of errors in production code.

Timeline:



Pre-GSoC Period	24 Mar - 8 May	- Fix existing issues in cli, core and Blaze and other assigned issues in Website etc. - Read the book "Unifying Business, Data, and Code" and Thesis written by Juan to get a better understanding of JSON Schema and its application at scale - Research and Identify common linting issues, have discussions with mentors and community, update the qualification task (basic set of 10 rules) and classify rules better. Update the extension's lint capabilities.
Community Bonding Period	8 May - 1 Jun	Research and Identification of Common Linting Issues - Gather data from various sources (StackOverflow, Slack, GitHub discussions) on common linting issues.

		- Analyze the collected data to identify recurring themes and prioritize the issues. - Discuss findings with the community to validate and refine the list of common linting issues. Deliverables: List of common linting issues and prioritized categories. Note: Since I have been active in the community and in constant touch with mentors, the time here will be utilized in collecting queries and topics about JSON Schema linting and talking to other people in slack to discuss Linting rules.
Coding Period Starts	2 Jun - 14 Jul	Categorization and Prioritization of Rules - Organize the linting rules based on their impact (e.g., high-severity issues like missing *$schema* keyword). - Create a detailed categorization of linting rules. - Continue engaging with the community for feedback and refinement. - Start implementing decided rules to the CLI. Deliverables: Complete linting rule document with examples and explanations.
Mid-Term Evaluation	14 Jul - 18 Jul	Submit the Mid-term deliverables and start implementing the linting rules page in JSON schema website
Coding Period	19 Jul - 24 Aug	Creation of Linting Rule Document and implementing the rules in the cli

		- Write a detailed document that describes each linting rule with a link to it, including examples and explanations. - Implement the rules in cli, expand other repositories linked to it whenever needed(core, blaze etc.) Deliverables: Add linting rule document to the JSON Schema website and other relevant places, implementation of major rules in the cli.
Final Evaluation	25 Aug - 8 Sep	Submit the final deliverables and work on implementing other left out rules in cli.
Post GSoC Period	9 Sep - 17 Nov	Implement all the rules in the document to the cli and expand other open source tooling

Availability:

I plan to dedicate 35-50 hours per week to the project and will be most active between Thursday and Sunday from 8 AM to 7 PM IST. I will have my End-semester exams between 1 May - 18 May i.e. the community bonding period and will be able to dedicate 2-3 hours a day during that period.

Progress Report:

I will be updating and discussing with mentors daily on Slack and demonstrating my work through meetings or async. Additionally, I will write a blog every week on https://medium.com/@karanpalan007 about my progress and share it on my social media profiles.

Post-GSoC Plans:

After the completion of GSoC 2025, I plan to continue contributing actively to the JSON Schema ecosystem.

- Finalizing All Linting Rule Implementations: I will ensure that any remaining rules from the linting document that were not implemented during the standard GSoC

period are added to the Sourcemeta JSON Schema CLI.

- Implementing some rules in other tooling: I will work on expanding other JSON Schema tooling repositories like Corvus, JSONSchema.NET and others.

- Community Feedback Integration: I will review and respond to issues and pull requests from other contributors. This includes incorporating community suggestions and refining the linting behavior to reflect evolving best practices.

- As a future stretch goal, I would also like to explore exposing the linting engine as a public API, potentially hosted at `https://schemas.sourcemeta.com`. This would allow schema authors and tooling to directly consume linting functionality over HTTP, making it easier to integrate into editors, CI/CD pipelines, and other schema-driven workflows.

I am committed to staying involved with the JSON Schema community even beyond the post-GSoC period.

My focus is on creating a robust and community-driven linting system that adheres to JSON Schema standards and serves developers across various use cases. I view this project not as a one-time contribution but as the beginning of a long-term involvement with the JSON Schema organization.

Conclusion:

Thank you for reading. I have provided a detailed overview of my project and how I plan to execute it. For GSoC 2025, my main goal is to further enhance my understanding of the project by building on my practical experience and research.

The linting system I will develop will have a significant impact, as it will benefit developers across industries who rely on JSON Schema for data validation. The linting rules will help ensure that JSON Schema remains a reliable and consistent tool for creating interoperable, maintainable, and error-free data models. This will have a direct positive effect on organizations using JSON Schema in their production environments, improving the quality of their codebases and reducing potential errors.

As for the technology stack, I am well-versed in all the necessary technologies required for this project. I have extensive experience working with Web technologies and am refining my C++ skills. I am confident that I can complete this project within the given timeline and take full responsibility for implementing all the crucial and valuable features that will elevate JSON Schema linting to the next level.

I am 100% dedicated to JSON Schema and have no plans whatsoever to submit a proposal to any other organization.