The specific tech stack required to contribute to a project in the Google Season of Docs (GSoD) program can vary depending on the project you choose to work on. GSoD is not limited to a single technology stack or programming language, as it encompasses a wide range of open-source projects and documentation work. However, here are some general guidelines to get started with GSoD contributions:

1. Research and Choose a Project: Start by browsing the list of participating open-source organizations in GSoD and select a project that aligns with your interests, skills, and experience. Each project may have its own tech stack and requirements, so it's essential to choose one that suits your expertise.
2. Communication and Collaboration Tools: Familiarize yourself with the communication and collaboration tools used by the project, which may include email, mailing lists, chat platforms (e.g., Slack, IRC), version control systems (e.g., Git), issue trackers (e.g., GitHub Issues, Jira), and project-specific documentation tools.
3. Documentation Tools: Learn about the documentation tools and formats used by the project. Common documentation formats include Markdown, reStructuredText, AsciiDoc, and HTML. Some projects may use static site generators like Jekyll or Hugo for building documentation websites.
4. Version Control: If the project uses Git (which is common), make sure you are comfortable with version control concepts, and you know how to fork a repository, create branches, commit changes, and submit pull requests. You may need to contribute to documentation that resides in a version-controlled repository.
5. Build and Testing: Understand the build and testing procedures for the documentation. Some projects may have automated build pipelines that generate documentation, and you may need to ensure your contributions pass the required tests.
6. Review and Collaboration: Be prepared to collaborate with the project's maintainers and other contributors. Understand their code review and collaboration processes. This may include submitting pull requests, responding to feedback, and iterating on your contributions.
7. Programming Languages and Tools: Depending on the project, you may need to have knowledge of specific programming languages or tools. For example, if the project is related to software development, you might need to understand the programming languages and libraries associated with it.
8. Documentation Style and Guidelines: Each project may have its own documentation style guide and guidelines. Familiarize yourself with these and follow them when creating or editing documentation.
9. Project-Specific Knowledge: Depending on the project's domain, you may need to have some domain-specific knowledge or experience to create effective and accurate documentation.
10. Documentation Best Practices: Brush up on documentation best practices, including clarity, organization, consistency, and user-friendliness.

Remember that GSoD projects can vary widely, so the key is to research the specific project's requirements and tech stack. Once you've chosen a project, reach out to the mentors and maintainers for guidance and clarification on their tech stack and requirements. Your success in contributing to GSoD will depend on your ability to adapt to the specific project's needs and work effectively with the project's community.