Collaboration Guidelines

https://github.com/ecumenic/project/blob/master/Ecumenic%20Creative%20Operations%20Collaboration.pdf

1. Thanks For Seeking To Collaborate With Our Project:

Anyway you may support our works feeling awesome is very welcome. Be kind to read this document to understand your obligations and rights. And if you have any questions or suggestions please contact us.

2. Collaboration Types:

◇ **Sporadic Voluntary Contributions:** We are open for sporadic collaborations where you can simply contact us with your idea through our facilitating form (https://bit.do/ecumenic-contribution).

◇ **Development Community Commitment:** If you love our work and want to join our development community or one of our projects processes continuously you may start as a voluntary. We hire our contributors in paid jobs through fair-trade whenever possible.

◇ **Financial Business Contract:** If you wish to contract us for a financial business please read our project (https://bit.do/ecumenic-project), contract model (https://bit.do/ecumenic-contract), and send your proposal through our email ecumenic@protonmail.com.

3. Collaboration Compliances Commitment:

To join our voluntary development community you must commit to our project (https://bit.do/ecumenic-project) and its license agreement (https://bit.do/ecumenic-license), code of conduct (https://bit.do/ecumenic-conduct), these guidelines, and our legal regularization financial contract model (https://bit.do/ecumenic-contract).

4. Collaboration Decision:

◇ **Overall Project Review:** Understanding our project you may decide on what type of collaboration you want to offer us, so please take some time to analyze it and its current needs:

◇ **Organization Continuous Operations:** We are in constant need for issues and suggestions analysis, processes optimization, documentations and templates improvement, methodologies translation, new partnership pitching, faith-inspired creativity meetups and groups stablishing, workshops and conferences production, code cd-ci devops, coherent design and layout, educational research and reports, merchandise craft and distribution, advertising campaigns, social media material curatorship, computational applications development, new tutorials, review others codes, and mentoring.

◇ **Understand Your Own Potentialities:** Analyze how your talents and gifts may be of use for our compassionate peacebuilding laic-compliant interfaith-based project.

5. Proposing A Collaboration

Send us an email to ecumenic@protonmail.com with a paragraph about you (include special needs) and your work, your yet undeveloped desired contribution concise proposal, and feasable proofs you are able to develop and deploy your proposal in your proposed schedule, such as (but not limited to) your portfolio, execution project, and curriculum vitae. If you are a computational developer or programmer, follow our GitHub repository (https://github.com/ecumenic) and send also your username on the platform.

6. How We Analyze Collaborations:

◇ **Reliability Alignment** to our project (https://bit.do/ecumenic-project), our global standards charters (https://bit.do/ecumenic-compliance), and its license agreement (https://bit.do/ecumenic-license), code of conduct (https://bit.do/ecumenic-conduct), these guidelines, our ecumenic ecosystem trends indicators report & bibliography (https://bit.do/ecumenic-ecosystem), our ecumenic ecomuseology program (https://bit.do/ecumenic-ecomuseology), and our ecumenic ecocreativity research (https://bit.do/ecumenic-ecocreativity).

◇ **Priorities Agenda** of our entrepeneurial multi-level governance and planned processual optimization. We may not develop your idea right away but schedule it for a proper moment.

◇ **Collaborative Decision** from our community with the final commitment from our administrative board.

7. What Happens After A Proposal:

◇ **Collaboration May Not Get Accepted:** We'll try to explain suscintly why it wasn't accepted. Ultimately, however, you'll need to respect this decision. You're always welcome to make other propositions, and if you really believe your solution must be developed, announce us your fork project so we may analyze how to help you.

◇ **Collaboration Gets Accepted:** We will get back to you with a welcome letter including proceding instructions, scheduled methodology proposal, your role authorizations and project duties, and tips on presenting yourself to our community. It's helpful to watch the project for a while (on GitHub, you can click "Watch" to be notified of all conversations), and get to know community members, before doing work that might not get accepted.

8. What We Expect From Accepted Collaborators:

◇ That you follow our guidelines and methodologies.

◇ That you execute your proposed contribution following our agreement under the rule-of-law.

◇ If you can't find a solution for a problem, ask someone in charge.

◇ If you can't find a necessary feature, propose it immediatelly.

◇ Promote and help new user to fit into our enterprise.

9. Development Guidelines:

◇ **Version Controlled Methodology:** We use git methodology (https://en.wikipedia.org/wiki/Git) in our decision-making and conflict resolution. We "issue" problems analysis or proposals, and "pull" solutions reports in the project. This way we are able to have our projects changes tracking historic and get features back when necessary.

◇ **Public Non-Violent Communication:** Keep all communication public. Although it's tempting, don't reach out to maintainers privately unless you need to share sensitive information (such as a security issue or serious conduct violation). When you keep the conversation public, more people can learn and benefit from your exchange. Discussions can be, in themselves, contributions.

◇ **Build Relations:** Whether you're a one-time contributor or trying to join our community, working with others compassionately is one of the most important skills you'll develop in open-source cultural enterprises, so please be kind to present yourself as poethically and concisevely possibly, you can try a haiku with an explanatory paragraph and a weblink for your portfolio, example. Value others good developments and help those in need respectfully.

◇ **Check Dependencies:** Understand our own files formatting styles, cross-references such as weblinks which will need updating, and improve it as much as possible, or propose changes to it (such as developing an automated code formatter (https://github.com/dotnet/codeformatter). Before you open an issue or pull request review the project and its code (https://github.com/features/code-review), communicate to all the community to help your ideas come across effectively.

◇ **Synthetic Precision:** Keep requests short and direct. Give concise synthetic context. If you're suggesting a new idea, explain why you think it'd be useful to the project.

◇ **Collaborative Autonomy:** Do your homework beforehand. It's ok not to know things, but show that you tried. If you're running into an error, explain what you're trying to do and how to reproduce it. It's okay to ask questions (but be patient!). Everybody was new to the project at some point, and even experienced contributors need to get up to speed when they look at a new project. By the same token, even longtime maintainers are not always familiar with every part of the project. Show them the same patience that you'd want them to show to you.

◇ **Community Decisions Respect:** Respect community decisions. Your ideas may differ from the majority priorities or vision, they may offer feedback or decide not to pursue your idea. While you should discuss and look for compromise, maintainers have to live with your decision longer than you will. If you disagree with their direction, you can always work on a fork research or develop a fork project.

◇ **Compassionate Listening:** Above all, keep it politely respectful. Open source is made up of collaborators from all over the world. Context gets lost across languages, cultures, geographies, and time zones. In addition, written communication makes it harder to convey a tone or mood. Assume good intentions in these conversations. It's fine to politely push back on an idea, ask for more context, or further clarify your position. Just try to leave the internet a better place than when you found it.

◇ **Issues Solution Request:** Anybody in our project can raise an issue to be solved in one of our processes or materials for fixing so that a responsible people will take it forward to fix the problem after verifying it. It also is used to keep track of agendas, tasks, schedules, and the enhancement for our code base.

◇ **Sub-Project Pull Request:** The publication of changes done to a collective procedure and there is a request to merge the changes to another branch of the enterprise to be judged based on our guidelines. Interested collaborators can see the changes and suggest modifications, if the changes are acceptable, they can be merged, branched, and(or) forked.