More Content Visit Roadmaps	: Code Revi	ews	Find the detailed version of this checklist With details on how to implement these
Other Best Practices			https://roadmap.sh
		L	
Team Wide Pra	ctices		
			Evervone
Document and standardize the standardize th	ne code review process.		
Ensure that the purpose of co	ode reviews is clear to everyone.		
Ensure that "Definition of Dor	ne" is documented and clear to ever	yone	
Encourage team members to	participate in code reviews.		
Define a process for conflict i	resolution in code reviews.		
Have a definitive style guide f	or style preferences.		
Use automation to speed up	the code reviews (linting, sniffing etc)	
Set clear expectations for co	de review turnaround times.		
Provide adequate time for co	de reviews and ensure that it is a pri	ority.	
Use code reviews as an oppo	ortunity for knowledge sharing and le	arning.	
Encourage reviewing code in	unknown-areas for cross-functional	knowledge.	
Constantly monitor and impro	ove code review process.		
Recognition and rewards for	those with track record of quality fee	dback.	
Encourage communication/c	ollaboration; avoid treating code rev	iews as a one-way proc	ess.
Hold regular code review sea	ssions to discuss broader trends or is	sues that arise during t	the review process.
Encourage authors to seek for	eedback during development before	submitting a formal cod	de review.

Follow the coding standards and any other team guidelines.	
Stay consistent with the overall project design and architecture.	
$\lambda \lambda / rite a failing test if the change is far a bug fix$	
white a failing test if the change is for a bug lix.	
Break down complex tasks into smaller easily manageable PRs.	
Consider the impact of the change on other parts of the system	
Consider the impact of the change of other parts of the system.	
Take notes on any questions or concerns about the change to discuss them during the review.	
Write the automated tests	
while the automated tests.	
Write the documentation for the feature or changes if required.	
Undate any documentation that may have made obsolete through the changes	
opulle any about that may have made obbolete through the bhangeb.	
After the Development (Author)	
	Author
	Aution
Review your code before submitting for review.	
Ensure that the changes are complete and ready for review, including all necessary tests and documentati	on.
Verify that the code change has been properly tested in a development environment.	
Double-check that the code adheres to the project's coding standards and best practices.	
Identify any potential performance, security, or scalability concerns and note them for discussion during the	e review.
Make sure to add proper title, description, any screenshots, relevant links, configuration changes etc in the	PR.
Approach the review process with an open mind, and be willing to learn from and collaborate with other tea	am members.
Refere Reviewing (Reviewer)	
Understand the requirements and the context in which change was made.	Keviewer
Pood on the requirements preserve a list of items that should have be a set of the set	
Dused on the requirements, prepare a list of items that should have been covered in the changes.	
Ensure that you understand the codebase and its architecture.	
Review any documentation or design energifications related to the shange	
neview any accumentation of design specifications related to the change.	
Make list of any potential risks or issues that could arise with the change.	
Approach the process with an open mind: he willing to provide constructive feedback and collaborate to im	prove code quality
Approach the process with an open mind, be winnig to provide constructive reeaback and conaborate to m	iprove code quality
Consider the overall quality of the code, including readability, maintainability, and scalability.	
Determine the appropriate level of review needed based on the scope and impact of the code change	
Determine the uppropriate lever of review needed based on the scope and impact of the code change.	
Be willing to collaborate with the author to resolve any issues or concerns that arise during the review proc	Cess.
During Code Review (Reviewer)	
	Reviewer
Be respectful and professional in your feedback, avoiding personal attacks or derogatory comments.	
Duranida ala an and action able foodbook including an action actions for increased and and and and are to be a	
Provide clear and actionable reedback, including specific suggestions for improvement and explanations of	of any concerns.
Identify any potential performance, security, or scalability concerns, and discuss them with the author.	
Driaritize your feedback feedback feedback and the meeting and important isource first	
Prioritize your feedback, focusing on the most important issues first.	
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e	dge cases.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices.	dge cases.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices.	dge cases.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated.	dge cases.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre	edge cases. eferences
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard	edge cases. eferences
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection.	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection.	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind.	edge cases. eferences s
 Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and expression in the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal produce comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. 	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews.	edge cases. eferences s
 Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and end that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal produce comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team 	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author)	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author)	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author)	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and end of the end of the project's coding standards and best practices. Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal provements to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised.	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and est Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and end to be the service of the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal provide comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed.	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and est Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes	edge cases. eferences s morale.
 Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and elements that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal proceed to comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. 	edge cases. s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and est Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes.	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes.	edge cases. eferences s
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the undated code for a second review if needed	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed.	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes.	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed.	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer)	edge cases. eferences s morale.
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and ends in the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal provide exploring the standard seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer)	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Besolve conflicting opinions in a timely moment don't lat a PB sit around due to discoreement	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and end is the the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal provide explanations in provements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments alfected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if it's not artitical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Beview the updated code and ensure that the valuessed by the author.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that they all pass.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that they all pass. Address any questions or concerns that the suggested changes have been implemented as expected. Run the tests again and ensure that they all pass.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that they all pass. Address any questions or concerns that the author may have.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that they all pass. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pri Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments offected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author: Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the yall pass. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticiam, to reinforce good practices and boost team Atter the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that they all pass. Address any questions or concerns that the authors. Be open to feedback has been addressed by the author: Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. After Approval (Author / Reviewer)	edge cases. s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "NIt" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. Rut the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author: Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the yall pass. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. Atter Approval (Author / Reviewer)	edge cases. eferences s morale. Author Reviewer
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team Atter the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. Atter the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the auggested changes have been implemented as expected. Run the tests again and ensure that the auggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the auggested changes have been implemented as expected. Run the tests again and ensure that the auggested changes have been implemented as expected. Run the tests again and ensure that the auggested changes have been implemented as expected. Run the tests again and ensure that the auggested changes have been implemented as expe	edge cases. eferences s morale. Author Reviewer
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the yall pass. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. After Approval (Author / Reviewer) Merge the approved code change into the main/feature branch. Verify that the sode change into the main/feature branch.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if its not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the uthor may have. Be open to feedback from the author and be willing to make adjustments to your feedback if neccess	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the moet important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave aomments to suggest improvements, but prefix it with "Nit" if its not aritical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive ariticism, to reinforce good practices and boost team Atter the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner, don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if its not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. Run the tests again and ensure that the augested changes have been implemented as expected. Run the tests again and ensure that the augested changes have been implemented as expected. Run the tests again and ensure that the all pass. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. After Approval (Author / Reviewer) Merge the approved code change into the main/feature branch. Verify that the code change is functioning as expected in the production environment. Monitor the performance and functionality of the code change and address any issues that arise. Celebrate the sugcestif complete into the code channee in dadress any issues that arise.	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and et Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pre Leave comments to suggest improvements, but prefix it with "Nit" if its not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost learn After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes. Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the suggested changes have been implemented as expected. Run the tests again and ensure that the author may have. Be open to feedback if necessary. Address any questions or concerns that the author may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. Atter Approval (Author / Reviewer) Merge the approved code change into the main/feature branch. Verify that the	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and e Ensure that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation or code comments affected by the changes. Submit the updated code for a second review if needed. After the Code Review (Reviewer) Resolve conflicting opinions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the outhor. Review the updated code and ensure that the auggested changes have been implemented as expected. Ruther tests again and ensure that the auggested changes have been implemented as expected. Ruther of genes and questions are addressed by the outhor. Review the updated code change into the auditor may have. Be open to feedback from the author and be willing to make adjustments to your feedback if necessary. After Approval (Author / Reviewer) Merge the approved code change into the main/feature branch. Verify that the code change is functioning as expected in the production environment. Monitor the performance and functionality of the code ch	edge cases. eferences s morale. Author
Prioritize your feedback, focusing on the most important issues first. Review any tests included with the code change to verify that they adequately cover the functionality and effective that the code change adheres to the project's coding standards and best practices. Ensure that the relevant documentation has been updated. Team wide styleguide is the absolute authority styling. Verify changes against those instead of personal pro- Leave comments to suggest improvements, but prefix it with "Nit" if it's not critical to meeting the standard Seek continuous improvement, not perfection. Keep the short-term and long-term considerations in mind. Consider using pair programming as an alternative or supplement to code reviews. Provide positive feedback in addition to constructive criticism, to reinforce good practices and boost team After the Code Review (Author) Address all the feedback received, including any concerns or questions raised. Implement the suggested changes and provide explanations where needed. Run the tests and ensure that they all pass after making changes Update any documentation ar code comments affected by the changes. Seek feedback from other team members if you are unsure about the changes. Submit the updated code for a second review if needed. Run the tests again and ensure that the yall pass. Address any questions in a timely manner; don't let a PR sit around due to disagreement. Verify that all the feedback has been addressed by the author. Review the updated code and ensure that the yall pass. Address any questions or concerns that the upiling to make adjustments to your feedback if necessary. Atter Approval (Author / Reviewer) Merge the approved code change into the main/feature branch. Verify that the code change is functioning as expected in the production environment. Monitor the performance and functionality of the code change and address any issues that arise. Celebrate the successful completion of the code change and address any issues that arise. Celebrate the successful completion	edge cases. eferences s morale. Author

Keep Shipping