



Monthly Takeaways

Krishnaraj Thadesar

Associate Software Developer Intern

5th September 2024

Agenda

- Introduction
- Project Details
- Description of Core Work
- Software and Applications used
- Challenges Faced
- Takeaways and Learnings

About me and my Team

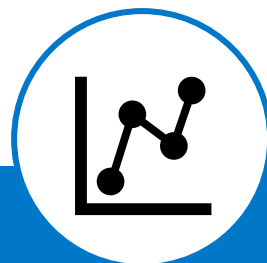
Me	My Team	Cruise RMS
<ul style="list-style-type: none">• Part of the Cruise RMS team.• Currently undergoing an Angular Training• I intend to also work in making the UI as soon as we finish training.	<ul style="list-style-type: none">• My team consists of Arun Pratap Singh, Harekishan Shivnani, Sunanda DuttaSen, Rajendra Mulgali and Omkar Chogale• My Mentor and manager is Mr. Harekishan Shivnani.	<ul style="list-style-type: none">• We are developing a new Product for IDEaS• It involves Forecasting revenue for Cruise Companies.

Core Work Ongoing



Process 2 new Files

Out of Order and
Historical Pricing files
need to be processed



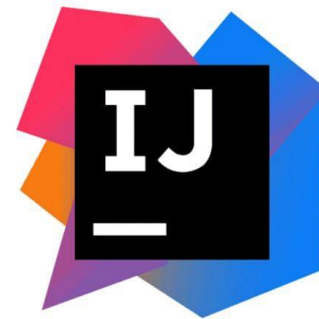
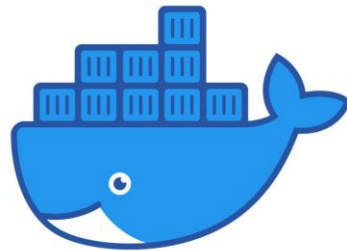
Analyze Files

Perform basic analysis on
the file, and try to find
any potential errors or
edge cases it may have



Angular Training

Attend and learn Angular
for building frontend.



Software and Applications Used



Challenges Faced



Keeping up with Code Quality



Build failures in github pipeline

Technical Learnings

1. All of your code logic for any function must reside within the height of your monitor, and you mustn't have to scroll. If you do, write functions.
2. Try to reduce any duplication of code.
3. Try to induce the feeling of being a developer in your team, rather than a temporary helper.
4. It is better to do one thing with accountability than 4 things that you aren't sure about. Reliability > versatility.
5. Do not love your code.
6. Do not say something doesn't work, without *completely* making sure it doesn't.
7. Outline your test cases before you write them, name the functions first.
8. Merge with main every now and then if your PRs last long.

Technical Learnings

9. Every now and then, make sure to check everything is up and running.
10. Make small Pull requests. Even smaller commits. Try to catch it up with your stories and subtasks in stories.
11. Simplify things – Complex is fun and exciting. It has a sense of awe. Simple code is understandable and therefore maintainable.
12. Breaking things down and doing things step by step is always better than combining a few things together. Its okay if work happens slower than expected, as long as it happens right.
13. Write test cases as soon as you finish writing a file.
14. Support using a pre-tested pre-existing famous library instead of writing any logic by yourself, only to have more things to test.
15. Try to reduce the usage of if-statements wherever you can. More branches.

Personal Learnings

1. Being a good listener goes a longer way than we give it credit.
2. Before you ask request someone anything professionally, make sure they haven't already done it.
3. When asked *anything*, be honest about it. Even if the answer sounds disappointing.

$$I = I_0 \sin \omega t$$

Thank You

Continued gratitude to Mr. Arun Pratap Singh and Mr. Harekishan Shivnani for their over the clock availability for any and all help I need

info@ideas.com | ideas.com

