**analysing**

base on the project, the aim is to address the efficiency of knowledge and experience transfer. To this end, we gonna need to design, build, and implement a system for storing and retrieving Failure Reports, Analysis Results, and Corrective Actions. This system needs to be able to accept user input, store data, allow users to query and retrieve data, and also support data analysis to a certain extent to reveal repeating failures, bottleneck failures that are most likely to impact the project schedule, failures that are most likely to impact the project budget, and failures that lead to cascading further failures. Considering these requirements, there r some personlly suggestion:

Backend: Python + Django.(easy to pick up eventhough ive never learned before tho)

Database: PostgreSQL or MySQL(from cits1402). Both are powerful, reliable relational database systems that are very suitable for storing and querying large amounts of data. If your project needs to perform complex queries or data analysis, PostgreSQL might be a better choice because it supports more advanced query and analysis features.

Frontend: React. React is a powerful JavaScript frontend library that can be used to create componentized user interfaces. It is very suitable for creating modern, interactive web applications.

additional inforamation abt another option with frontend

React or Vue.js

(react is not a language is part of js, so we cant compare with html css.)

**Learning：**

Vue's template syntax and component structure are relatively intuitive and more friendly to beginners.

Vue.js's documentation is often considered easier to read and easier for beginners to understand.

React leans more towards JavaScript, using JSX syntax, and requires a certain level of understanding of JavaScript to use effectively.

**Flexibility**:

React follows the philosophy of being as simple as possible, providing only a solution for the view layer, while other things like routing and state management need third-party libraries. This offers you greater flexibility but also means you have more decisions to make. Vue.js, on the other hand, provides a more complete solution, including routing and state management, which might be easier to get started with for simpler projects.

**Community and Ecosystem:**

React, being around longer, has a larger community and more third-party libraries and tools. This means that if you encounter problems during development, you're more likely to find solutions online.

personally:

In this project, the choice of frontend might not be that important, as the main functionality of the system is data collection, storage, and querying, which are primarily backend tasks. and i dont think we have enough time to deal with frontend tbh. so if we good at js, go for react, otherwirse think abt others,