

User

Documentations

ltems/

Tasks to be linked with the associated requirement. The user must be able to tell which requirement he addressed with that particular task. It would be better if we ask the user to have a specific report format, it can be a specific markdown format too.

The user must describe the task, what it does and the function associated with it we can provide the whole codebase to the embeddings and keep it up to date so that it can be analysed and the user can get timely reviews.

The API calls will return all the suggestions to the user it is then the users choice to accept them or not the UI can be like a github merge request with suggestion.

```
typescript
                                                                          Copy code
export const createNewFoodItem = async (req: any, res: Response, next: NextFunction)
       const userData = req.user;
       const currentUser = await user.findOne({ Email: userData.email });
       if (!currentUser) {
           return res.status(404).send("User not found");
       if (!currentUser.Eating_Mess) {
            return res.status(403).send("Manager is not associated with a mess");
       const { name, image, category } = req.body;
        const createdFoodItem = await mealItem.create({
            Name: name,
           Image: image,
           Category: category,
           Mess: currentUser.Eating_Mess
       });
       return res.status(200).send("Inserted");
   } catch (error) {
       console.error(error);
       return res.status(501).send("Internal Server Error");
```

The analysed code can be displayed side by side to the user so that he can view the changes to be made.

Other option is to ask
ChatGPT to produce a
code with all the changes
incorporated and the user
can accept all of them.

User

ChatGPT

Based on the provided 'createNewFoodItem' function and the context of Requirement 1 (*The manager must be able to add food items to his mess only. And those food items must be kept from other managers. Each manager has his separate view."), here are potential areas where bugs or issues could arise:

1. Missing Authentication Check:

• The function assumes that `req.user` contains valid user data, but there is no explicit check for manager authentication. Ensure that the request is coming from an authenticated manager.

```
typescript

// Potential improvement
const userData = req.user;
if (!userData || !userData.email) {
    return res.status(401).send("Unauthorized");
}
```

2. Incomplete Mess Association Check:

• While there's a check for the existence of the manager's mess, the function does not explicitly check if the manager is associated with that mess. Ensure that the manager has the authority to add items to the mess.

```
typescript

// Potential improvement
if (!currentUser || currentUser.Eating_Mess !== userData.messId) {
    return res.status(403).send("Manager is not authorized for this mess");
}
```

3. Input Validation:

* The function assumes that `req.body` contains the necessary properties (`name`, `image`,

We can have a) Frompt on the side Suggestions like this can be displayed to the user with potential improvements. It is up to the user to accpet them.

1. Identify the code lines
2. Ask the user to accept the changes
3. Yes/No

I am still figuring out how to get the line numbers but what we can do is display this as a bug and needs to be fixed before procedding.



6 ChatGPT

In the original code you provided, here are the approximate line numbers where you can insert the suggested authentication check:

```
typescript

// ... (Previous code)

export const createNewFoodItem = async (req: any, res: Response, next: NextFunction)
    try {
        // Line 23: Insert the authentication check here
        let data = req.user;
        if (!data || !data.email) {
            return res.status(401).send("Unauthorized");
        }

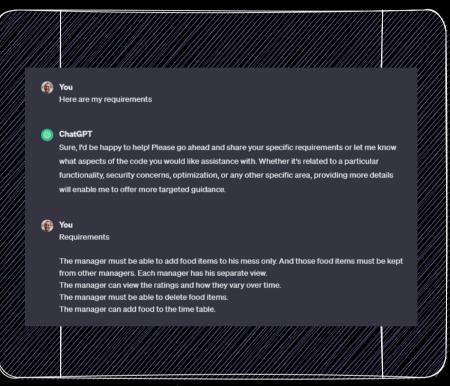
        // Continue with the rest of the function...

        let currUser = await user.findOne({ Email: data.email });
        // ...
    } catch (e) {
        // ...
    }
}
```

Insert the authentication check at the beginning of the 'createNewFoodItem' function, as shown in the snippet above. Adjust line numbers accordingly based on any modifications or additional code in your actual file.



Client



Send Requirements

This will be automated ChatGPT
To assess how closely the 'deleteTimeTableHandler' function eligns with your requirements, let's refer back to Requirement 3:

Based on the report Format we will ask all the questions.

This will provide

Requirement 3: The manager must be able to delete food items

let userMess = currUser.Eating_Mess;

Now, let's review the 'deleteTimeTableHandler' function:

how closely the function aligns with my requirement

typescript export const deleteTimeTableHandler = async (req: any, res: Response, next: NextFunct let data = req.user; try { let currUser = await user.findOne({ Email: data.email }); if (!currUser) { return res.send("User Not Found").status(404); } else {

(You

(You

Give a percentage closeness

Assigning a precise percentage to the closeness of code to a requirem Pierre supply may depend on various factors. However, based on the analysis of the / coseness 'deleteTimeTableHandler' function with respect to Requirement 3, it appears to align quite closely. I would estimate a high percentage of closeness, per page around %0.95%.

The function checks for user authentication, verifies the manager's mess association, and handles the deletion of a specified food item from the time table. It also provides appropriate responses in case of success or errors.

To improve this percentage or achieve a higher level of confidence:

- 1. Testing: Perform comprehensive testing to cover various scenarios, including edge cases and notential error conditions
- 2. Feedback Handling: Consider providing more detailed feedback to the manager in case of errors. helping them understand the reason for any issues.
- 3. Logging: Ensure that the function logs relevant information for troubleshooting and monitoring.

These additional steps would contribute to a more robust implementation and enhance the overall alignment with the requirement. Keep in mind that code quality and correctness are iterative processes, and continuous testing and feedback are essential for improvement



Manager/can/manually

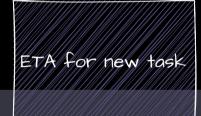
You

are the food items not visible to other managers

ChatGPT

In the provided 'createNewFoodItem' function, there isn't a specific check to ensure that the created food items are associated only with the manager's mess. This means that, as it stands, food items are not inherently restricted to a specific manager's mess, and they could potentially be visible to other managers.

To address this and fulfill your requirement that "the manager can add food items to his mess only." you should add a check to verify that the manager creating the food item is associated with a specific mess. This check can be performed by comparing the manager's mess with the mess associated with the created food item.



(fig You

As per the previous reports now much time with it take?

ChatGPT

While I can provide general estimates based on typical development tasks, the actual time required to implement a feature can vary based on several factors, including the specific details of your codebase, the complexity of the implementation, and any unforeseen challenges that may arise during development. The previous reports provided information on specific tasks and the time taken for those tasks, but estimating the time for a new requirement involves additional

Considering the context from the previous reports and the typical tasks involved in implementing a new feature, here's a rough estimate:

- * New Requirement: The manager can add food to the time table.
- * Estimated Time: Given the context of your application and the previous reports, a rough estimate might be in the range of 4 to 8 hours as mentioned earlier.

Keep in mind that these estimates are speculative, and the actual time required may be more or less depending on the specifics of your implementation. It's recommended to break down the