HEURISTICS	VIOLATION	RECOMMENDATION	SEVERITY
1. Visibility of system status Designs should keep users informed about what is going on, through appropriate, timely feedback.	The app fails to show parents and users the number of milestones in checklist and how many left to be answered	Add a progress bar on top to show progress when users access milestone checklist screen Show the total number of milestones under each category and how many were checked	3
1. Visibility of system status	Email provider feature is broken. The user is not getting a warning notification of this. And they are not getting a notification that email failed to send	Fix the issue in this feature with dev team. If feature is not ready to be released, mask it in the upcoming release. Make sure to add a notification of status, if feature to be fixed and released	4
2. Match between system and the real world The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon.	There's inconsistency in using pronouns for children. The app alternates between "he" and "she". Can be confusing to users.	Use child's name in text, and use pronouns provided by users	2
3. User control and freedom Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted state.	Users can not undo their selection in milestone checklist.	Add a "clear" option next to milestones checkboxes Provide an option to review changes before exiting	3
4. Consistency and standards Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.	The wording of menu items are confusing (e.g. core principles). order of menu items keeps changing when users navigate through menu	Change the wording used and test them with users Define MVP requirements (this is hasn't be clearly defined before starting the replication of CDC. This is critically needed before changing the navigation items)	4
5. Error prevention	Refer to "**Visibility of system status** " Required fields in registration/ login screens have asterisks (*). if user misses a field, a "fields with * are required" pops up. this can be confusing to the users who may get frustrated if they kept failing to answer required questions	Provide a clean and simple one-line description under missing fields	3
6. Recognition rather than recall Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.	All navigation options are listed in the navigation hub. There is no dedicated homepage that users land on to check their recent activities and reminders to complete tasks To navigate to a new location, users have to first go back to the hub and then use one of the options listed there. This would limit users ability to accomplish multiple tasks in the app. This navigation will need a lot of recall that can be relatively difficult and annoying.	Change and test a navigation style that will help users perform different tasks in a hierarchy that matches their mental model.	4
7. Flexibility and efficiency of use Shortcuts — hidden from novice users — may speed up the interaction for the expert user.	The application lacks the hierarchy needed to allow users to approach tasks in multiple ways that will suit their mental model and working style.	It is critical to decide on navigation style and define application feature before proceeding with redesigning the application. Create user flows (based on user stories) and a hierarchical sitemap for improved recognition and discoverability	4
8. Aesthetic and minimalist design Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.	The application includes menu items (labels) that are disabled. This can make users wonder why these items are not available and how to enable them. In return, this will increase the cognitive load on users and increase the chances that they will	Define MVP features and include only the ones that are ready to be used. Features that are not ready to be released should be masked or scheduled for later releases	4
9. Help users recognize, diagnose, and recover from errors	Refer to "error prevention"	Address error cases and edge cases in user flows Error messages should be expressed in plain, easy to understand language, and they should suggest a solution	3
10. Help and documentation	n/a	Design an engaging onboarding flow Show users their progress Use tooltips when needed	2