Progress Report

Katie Hoskins, Tanya Peacock, Marcus Stange

In our previous diagrams, we made a strong explanation of our entire project, but there were a few things that needed to be changed throughout the process. When it comes to the database, the biggest change that we made was adding a “Schools” table. This was incredibly necessary in order to assign users to schools. Without this table, this would have been much harder. The user is assigned a school_id foreign key that matches a school_id on the “Schools” table. From there, we can gather the information needed from the schools table that correlates to the user. Without this table, we would have to add an array of schools to the specific page we want to use it for, and from there gather all the information. This would be much harder and make the code much more difficult to read as well. The “Schools” table contains the school, school_id, principal, pEmail, assistantPrincipal, aEmail, assistantPrincipalTwo, aEmailTwo, assistantPrincipalThree, and aEmailThree. This holds the information for the specific schools and who runs them. This will specifically be for contact information. Another big change that was made is that we completely removed the “Register” page. We found this page to be unnecessary since the “Sign In” page takes the same information as the “Registration” page would have. Instead, now we just check if the email already exists in the database. If it does not, the email is added to the database, but if it does, we just take the user information from the signed in email and connect it to the page. Another change that we have made is that we are adding the events that are only happening today to the main dashboard page. All of the information is not immediately visible to the user. The user can click on the event, though, and a modal will pop up. In the modal, it contains the information from the event in the database. All of the information is added to that modal and is not editable. For the “Events” page, we have now gotten a real calendar implementation. When we first started working on the page, it was just a text box that allowed the user to input the correct information. On the uneditable events page, we were going to add a list like we have on the dashboard to allow users to see what events are coming up. Now, we have the calendar implementation there as well. This shows users which events are coming up for the month. They can also change months to see events that are happening more in the future. Today's date is highlighted on the calendar so the user will automatically know what is happening today if they do not go to look at the dashboard. In the users page, we have modals that are editable by the administration. Previously, we were just going to use a simple table like we have on our dashboard, but for users. This makes things a lot easier for the user as they are now able to simply click the edit button, and have all the information they need to pop up and edit. A large amount of our changes are based on user experience, and making things easier for the user to see and do.

We are about 75% done with our website. We have completed the Accounts page where the user can view and edit their information, the Dashboard where users can view upcoming events, the Home, Contact, and Sign In pages where users can view things before signing in, and the Users page where administration can view the users and edit it if needed. The main things that need to be revised or completed are the "Events" page and "Teacher Events" page. The only issue that we are having with this is getting the times formatted correctly. The database uses ISO time, but we want to make it compatible with browser time. We are working on this, but we are having a hard time getting the times to get formatted correctly on both pages. The Roster page needs to be finished. The biggest thing that needs to be fixed there is setting the presence for students. This will need to be a dropdown and we need to find out how to implement the dropdown for Supabase. The "Archives" and "Reports" page needs to still be implemented. The biggest problem that we face there is we need to have the “Roster” page completed in order to send the information to the administration. This still needs to be completed, but once we get the “Roster” page working, we will be able to easily implement the other pages that we need. The “Teacher Contact” page is almost complete as well. We want to use a table format for this, but we need to be able to match the school_id to the “Users” table and that is coming to be a hassle that still needs to be completed. Other than that, everything is completed and ready to be published.

There have been a few problems that we have come across in the project. Some relating to the actual code itself, some related to the database, and some related to conflicts in the group. When working with the database, some members of the team have had some disagreements about certain things that need to be done. For example, one member wanted to have an id for each user, but another wanted to keep the “id” as the email. Things like this are easily resolved, but do need some more discussion in the long run. Another problem between group members has been that one member may not understand something, and the other may get frustrated that they do not understand it. This causes some strife between the members, but ultimately gets fixed. When it comes to the database, some problems such as RLS policies have been a smaller issue. They were difficult to implement in the first place, but once they were solved it was much easier to work with. The database also needs to use triggers, which we have not quite figured out yet. In order to automatically add a user from the auth.users table to a public.users table, we need to use triggers, but sometimes they do not work correctly. This is still something that we are working on. In the code base, some things like time formatting has been difficult. Sometimes the formatting works in one place, but not the other. One big issue that we face is communication between the group. One person may make a change, but not say anything to the other people.

This causes some confusion in the project when some code has been changes that other people did not know about. Once we communicate about the changes though, there is no other issues. Also, sometimes a group member may push a change to the GitHub that breaks the website. This is a huge issue when it happens, even if rarely. This goes hand in hand with the communication issue.

Overall, the project has run smoothly other than a few bumps in the road. As we have continued through this project, less problems have arisen within the team and we are learning how to work together. The problems in the code have become less and less as we have learned how to research issues we are facing and reusing code when necessary.