Individual Work Log

PROJECT NAME:	**Genealogy Application / Website**		
STUDENT NAME:	**Marco Giacoppo**		
STUDENT ID:	**104071453**	**WEEK # (& dates covered):**	**Week 8 (14/04/2025 – 18/04/2025)**

TASKS	STATUS	TIME SPENT	ACTION ITEM/NOTE
Meeting with supervisor	Completed	30 minutes	
Attended the lecture	Completed	1 hour	
Edit relationship panel to add relationship	Completed	7 hours	Edited the sidepanel so we can add relationships using dropdown initial relation from add person button/
Debugging relationship between all different people	Completed	5 hours 30 minutes	This took me nearly 6 hours to do because it was a very complicated logic. There were so many bugs we found which I will explain later.
Worked on side-by-side node connections	Completed	4 hours	Worked on it for so long because didn't do any research on the library that we used. Turns out it was a simple mistake I did.
Research on Dagre.js	Completed	2 hours	Decided to do further research on how to

			implement 4 nodes instead of 2.
Fixing the card positioning and integrating new tree structure	On going	10 hours	Needed to fix this since the side-by-side node is manually coded. So far, the parent-child connection is looking good, but the spouse lines are not working properly.
TOTAL WEEKLY TIME SPENT		**30 hours**	

TASKS PLANNED FOR NEXT WEEK	**EXPECTED COMPLETION**
Begin to polish all the code structures	End of week 10
Try to find a better way to position all the cards	End of week 10
Add all small features so we can all focus on the bigger tasks	End of week 9

Summary/weekly reflection for Week #:

-Successfully added the edit relationship function on the sidepanel. This was very confusing since there were so many nested logic inside it.

*A few of them are **Bidirectional Relationships, Spouse Inference Logic, Order of Entry Matters, Avoiding Duplicates, and Sibling Inference.** All these were the reason why it took so long.*

1. A 'parent' relationship from A to B implies B is a 'child' of A. But if we only store one direction, we need to infer the reverse manually.
2. If one parent is linked to a child, and that parent has a spouse, we're expected to infer the spouse is also a parent, but only if the context makes sense. If that spouse is added later, the original inference logic wont pick it up unless we scan and merge all relations.
*3. We can add relationships in any order (child-dad, then mom – dad). But unless my logic re-evaluates **everything** each time, some connections will be missed.*
4. I have to make sure that I don't show 'Dad' twice if he's both explicitly and implicitly related. SQL UNION helps with that, but only if fields match perfectly.

5. Once you know two people share parents, you can call them siblings. But again, we're not just matching one field, we're doing relationship triangulation.

-These screenshots below will show the progress on the relationships between people and their respective card positionings.

This screenshot shows that the card is working totally fine if the connection is merely parent-

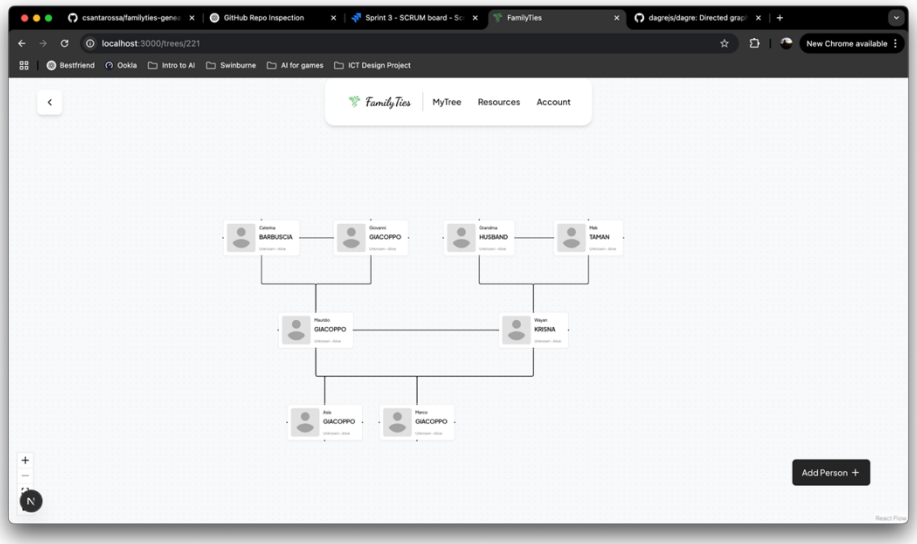

Figure 1: First look after programming the logics.

This figure shows that I found the first bug. Whenever users add a new sibling to their parents (in this case im adding my uncle from my mom). The card stacks on top of each other because of the snap-positioning wasn't working properly.

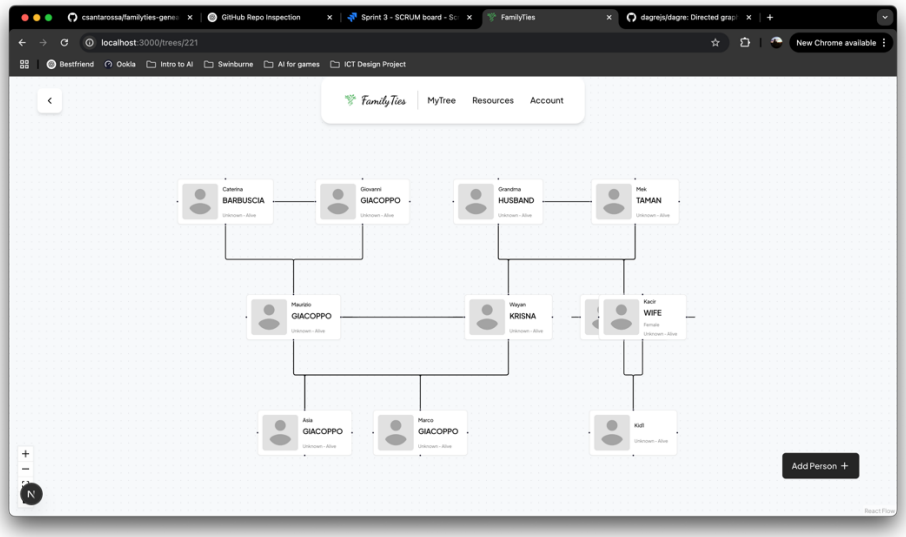

Figure 2: First bug found.

Here we can see that I'm testing something to make sure it's working. I added two 'new' parent as a parent of my uncles spouse. As we can see on *Figure 4,* turns out my logic works AS LONG AS every spouse relationship has got their own parents.

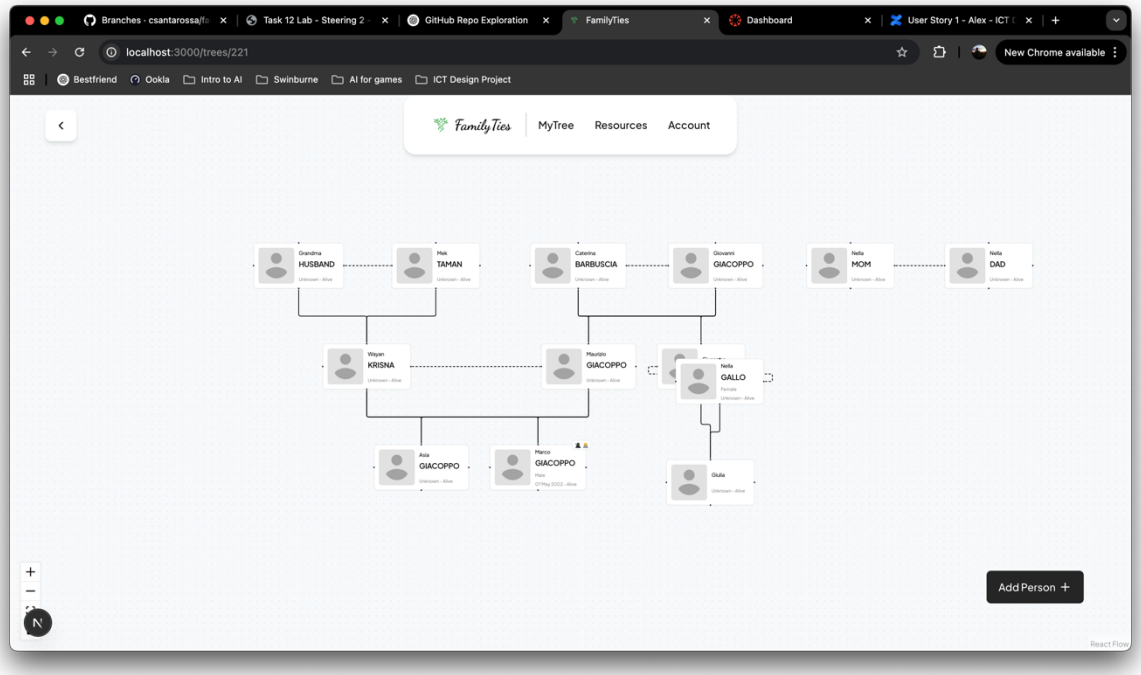

Figure 3: Did some testing.

This wasn't my goal but I was getting somewhere.

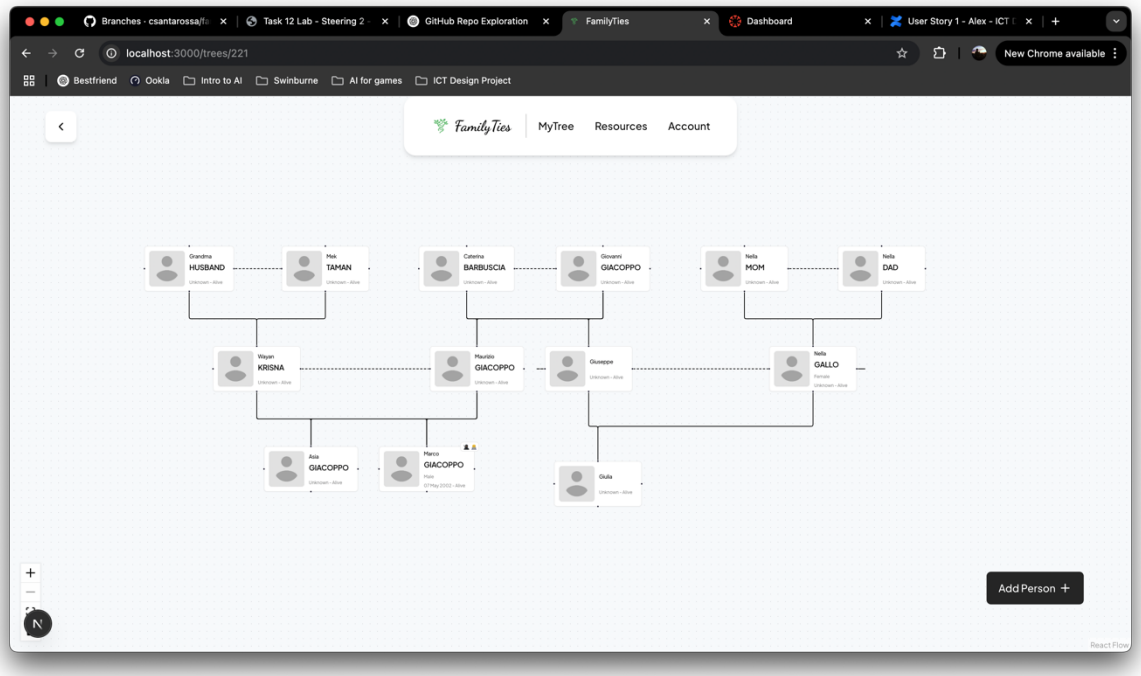

Figure 4: Halfway working

After modifying more codes, I managed to get the spouse to be placed next to it's spouse (as long as there's no other siblings present). This created an error where if there was a sibling, the spouse will be placed very far apart.

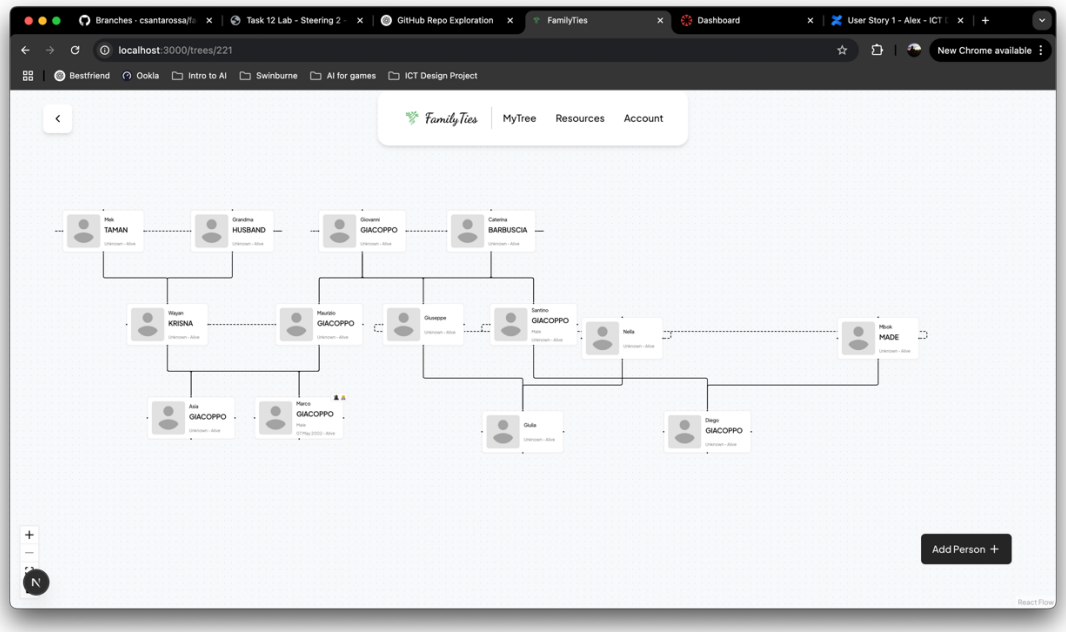

Figure 5: Modified the logic so that even though there's no parent, spouse is next to each other.

Finally, I forced the program to follow all the conditional logics I gave them and managed to show this working tree structure. There are still a lot of tests I need to do to ensure that this is fully working. For now, this is what I have.

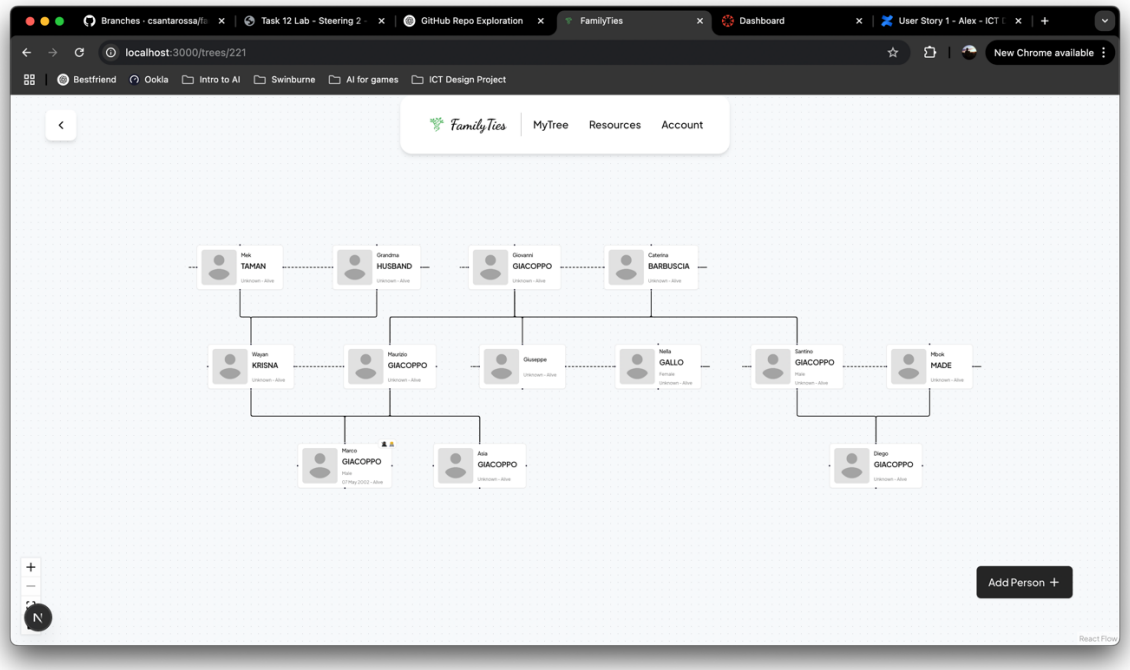

Figure 6: Final design model so far.