Challenges Faced:

With the different strengths and weaknesses and big challenge was to properly assign everyone to different tasks while also allowing everyone to work on both what they are strong at as well as what they struggle with, thus allowing us all to learn accordingly. We overcame this by using pair programming in which we would task a member who may be still be inexperienced or struggling with one task with a member who was strong in the task, thus allowing the deadline to be met but also so that the member who is struggling to learn more since they have someone by them who can help and teach them as they go.

Big issue the team faced was properly setting out a timeline and having set deadlines to adhere to. This was an issue mainly because none of us has had any experience with creating an application yet so we are not sure how long each task is supposed to or might take. We overcame this by putting in a lot of time early on to counter this inexperience and make sure we meet our set deadlines.

Our biggest challenge faced was the fact that none of us has actually implemented a fully functional application yet, thus we are all fairly new to it. This is compounded by the fact that some of our course we took first semester have faded slightly given the nature and speed of this program. This meant that we had to spend a lot of time refreshing ourselves on these earlier skills we picked up such as css, html, and webpage building.

Another difficult challenge we faced was the fact that none of us has had any experience with php scripting language. Luckily we had a very hard working group and people spent the time going through the php slides which gave us the knowledge to properly implement php into our application.

What our team Learned:

Technical:

As a team we learned many technical skills, we learned how to use php scripting language to query a database and then use the information to populate an html webpage.   
  
We also learned how to create and populate a database from scratch in which we could query it with the php mentioned earlier.

Another important skill we picked up is how to properly use gitlab and use proper version control management to develop our application, and how to properly merge different branches and work on separate parts of the application without affecting our “working” product.

As a group we also learned the importance of proper commits and how the messages require enough information to properly tell another person what was changed and why.

Soft Skills:

Communication is KEY! As a team of very different personalities it was extremely important for us to communicate properly with each other in a respectful and courteous way otherwise we would end up hindering our ability to get things done.

Flexibility was another huge soft skill that we learned. This is because we all have very different ways of thinking, this means that while one person may do something one way and another did it a completely different way it doesn’t necessarily mean that either was better but that they were just different. This subsequently meant that we needed the flexibility to adjust our code to another’s way of thinking and coding without making a deal out of it and adapting accordingly.

Teamwork, responsibility, and work ethic were all required skills that we seemed to have also picked up. To be a good teammate you need both responsibility and work ethic, because without those you cannot possibly be a good team member. In order to exhibit good teamwork everyone needs to understand and be on somewhat the same page that requires the members to be responsible and understand that they are responsible to not just themselves but also the team. This then needs the proper work ethic to make sure you and the others are on the same page and willing to all be equally responsible to each other.

What we would do differently next time?

If we could do anything differently next time, it would be having a more fleshed out requirements and SPECS document. We ran into slight trouble not fully understanding what exactly we were building, and what features are supposed to be there or whether or not to add newer ones which were not on the original SPECS, but were clearly something that should’ve been. This would’ve allowed us to have a more accurate timeline that would’ve subsequently made it easier to follow and wouldn’t have requirement as much adjustment as it did.