# Project Workshop – CPRG214 ASP.NET

Team 1:

Kehinde

Hazem

Terry

Darya

**Write an informal reflective journal of around half a page that discusses what you learned from the modules during this term, how your team-members worked together (were there communication/scheduling/etc. issues that had to be resolved?), what things worked best, and what you would do differently in the future. Don’t mention any names, just think about the processes your team used, how well they worked and how they could be improved for your next project. This should be in a Word/OpenOffice/RTF document which will be added to the zip file that you hand in with your second presentation.**

During this term we have learned a lot about the .NET Framework: how to develop windows form applications and web based applications, using ASP.NET. In particular, we have explored how to write good structured code, creating data sources and data object sources and binding them to controls, and the general workflow of developing and deploying a web application.

Some of the challenges our group has encountered during the Project Workshop included differences in schedules and some difficulty with concurrent work. We have resolved the scheduling issue by ensuring that we plan out our work while in class and divide up the tasks to finish them individually, working in parallel. The issue of version control and concurrent work has been resolved by making one of the team members a master keeper of the final code. Given the size of our application, the solution worked very well, but in the future, especially for larger projects, it would benefit us to use GitHub or any other open source, web-based Git repository hosting service.

One of ways we could have improved our workflow would have been spending more time reviewing the program as we write it. Ongoing review would ensure there are no surprises from anyone’s work, which was a possibility, given our parallel workflow. We could have also benefitted from a more detailed planning stage and ensuring we have solid knowledge of the program requirements. That would have saved us time and ensured that no extra time is spent on unnecessary features that are not required by the client.