Sprint Retrospective

Individual

Grant- During this sprint I feel as though I kept track of team progress well as the scrum master, making sure everyone understood their spring user stories and didn't face any major issues. The team may have overshot the first sprint criteria as my allocated task of building a login system using the database could not be completed in time as there were delays in setting up the database, thus the velocity of the team was slightly less than anticipated.

Kyle - Keeping up to date with team members worked well with the daily sprint standup, and it was easy to know what was happening in the project as the sprint progressed, and to be able to address any concerns as they appeared.
The amount of work that was required ended up being more than was expected, due to the inclusion of reading documentation, and extra tools needed to bridge between different elements of the project.

Christian - The main lesson I learnt from this sprint review is being more concise in the way I approach sprints. From the planning phase to the implementation phase, it is more valuable to spend time reflecting and considering the way in which to approach the given tasks and creating a clear outline in my head. This way it will make subsequent tasks more efficient and may improve the final product.

Cameron - During the sprint I researched into effective methods of creating a web server using node js and a platform to host the server. I learnt that it is important to keep track of each other's progress and communicate if you are having any sort of difficulty.

Lincoln - During this sprint, I feel as though I kept track of the website progress and database creation, taking it upon myself to set up the heroku app to get our proof of concept code working. I learnt that it is important to keep on top of tasks and make sure that every little detail is entered and that it covers literally everything in the project, including things that haven't been done or even discussed yet.

Overall

The team worked well during the sprint with strong communication and regular meetings, where any issues or problems were discussed and solved as a team. During the sprint review it was discovered that the client would like to be more involved in the project, wishing to be consulted when making changes such as using a PHP module to run HTML files on Heroku.

The completion of the sprint review showed that the client would prefer a switch from PHP to an extension of a Node JS build pack that allows for HTML files to be run. The client was happy with the UI, which the team feels positive about as well. Although integration testing is implemented by the team, the client would like to see more reporting on the testing and proof

of the definition of done concept, thus we will advise the client of more integration testing that was completed, as well as updating the project plan for the testing of each user story so that it is considered complete.

For the upcoming sprint, the sprint review offered some helpful insight. Firstly, the team will spend more time during sprint planning to make the user stories more specific. An individual definition of done would be preferable - with each user story having its own acceptance criteria. Not only will having this provide a checklist of what features really entail, it will also serve as a visual representation of tasks to complete within each sprint and ensure that no features are missed.
It is also beneficial to use the risk register when necessary. When new issues arise, always refer to the risk register for strategies to mitigate or avoid the risk. If we identify a new potential risk, we should raise it as soon as we become aware of it with the team and work quickly to develop new strategies to avoid the risk if it does become an issue.

Problems:
- Definition of done was not adequate
  - Did not include enough detail
  - In future should have detailed explanations of testing
- Sprint backlog and product backlog should include definition of done on all stories
  - Include what to look for and how to test
  - Should include quality testing as well
- Team roles should be more specific
- More communication on what needs to be done and what has been done
- The switch to php was not liked by the client, use javascript buildpack.
- Make risk register more clear
  - R05 was not adequate
  - R04 used an inappropriate word