

# Retrospective

## Main Criteria

- Has the retrospective identified genuine problems with the team's policies and practices?
- What strategies has the team decided on to address any identified problems?
- Are those strategies likely to work?

What went well?	<ol style="list-style-type: none"><li>1. As a team we created a successful sprint backlog which allowed everyone to digest and comprehend the requirements of this project. From this, we were able to constantly review our required workflow and arrange accordingly.</li><li>2. The efficient allocation of tasks allowed all team members to participate and contribute towards the coding of the product backlog and tasks interfaces. This prevented team members from requiring to learn a whole new language from scratch and instead focus on their more suitable ones.</li><li>3. An online MySQL database was successfully set up thanks to the quick upskilling of the team, allowing all users of the application to access saved data. All backend team members were able to access and make meaningful changes to the database.</li><li>4. Effective communication amongst group members ensured no conflicts within git between the team. For example, a simple heads up towards all other group members when members would be committing/pushing would suffice and any major code will be screened before deployment.</li></ol>
What could have been done better?	<ol style="list-style-type: none"><li>1. A lack of experience with the application had caused team members to have a hard time adjusting to time logging on jira. This resulted in either members being unable to log their time completely or having to provide estimates.</li></ol>

	<ol style="list-style-type: none"> <li>2. Upon starting the sprint, team members overestimated their capabilities due to over ambitiousness resulting in some tasks not being fully completed. This aspect should be heavily reviewed to ensure that this issue doesn't repeat.</li> <li>3. Due to team members' inexperience with certain languages or frameworks, progress slowed down at times due to extensive code reviews and online research.</li> </ol>
What will we try next?	<ol style="list-style-type: none"> <li>1. Now that we understand our capabilities better during our next sprint we will dedicate more time and effort onto sprint planning to ensure tasks are achievable.</li> <li>2. Better communication on who is doing what task and assigning these tasks first in the project management tool during sprint planning.</li> <li>3. Breaking down the epics and user stories into tasks before we begin the development phase to allow for better understanding, as well as more room for backlog refinement during the sprint.</li> <li>4. Stronger emphasis and higher expectation on time logging throughout the entire sprint now that the team has some experience working with the project management tool.</li> </ol>
What questions do we have?	<ol style="list-style-type: none"> <li>1. What should we do if we forget to add a user story to the sprint backlog?</li> </ol>