Anthony Legg - Studio 4 PDR

Link To GitHub Repo
Link To Studio 4 Production Website

Evidence For Submission

What are your personal goals and objectives in addition to the course objectives?

Development of communication, leadership and stress-management skills continues to be a major goal. I think that I have improved in this area from last semester, but there is definitely room for more improvement. I need to take let go more and take things much less personally. Working as part of a group is challenging; everyone has their way of doing things, so I need to be more patient and less reactive.

Would like to get more skills and experience in CI and CD, using services such as GitLab's or Azure DevOps to streamline the project from planning through to deployment. Would also like to apply some of the skills learn from the operations engineering paper in future projects. I think docker would have been hugely beneficial to this project, it would have eliminated some of the configuration issues across multiple development environments.

What has gone well? What are your strengths?

As a team we were able to improve all aspects of the project that we started with. The site has been made more secure, easy for the end user to navigate, consistently styled across all pages, and provides functionality to suit the clients needs. The most important thing is to deliver on the goals of the client; I think that ultimately we delivered on most of these objectives through communicating and working as a team. Every sprint resulted in improvements in the product and the way we worked together.

I think my strengths are honesty, reliability, empathy and self-motivation.

What could have gone better and how?

The big one getting everyone working together rather than individually.

What is not clear or is uncertain?

At this point, I don't actually feel unclear about anything. I learned a tonne from this paper and feel like I have developed a heap of new skills and better ways of working.

Learning Outcome 1

> Select and apply industry-standard tools and processes to solve non-trivial problems in a team environment.

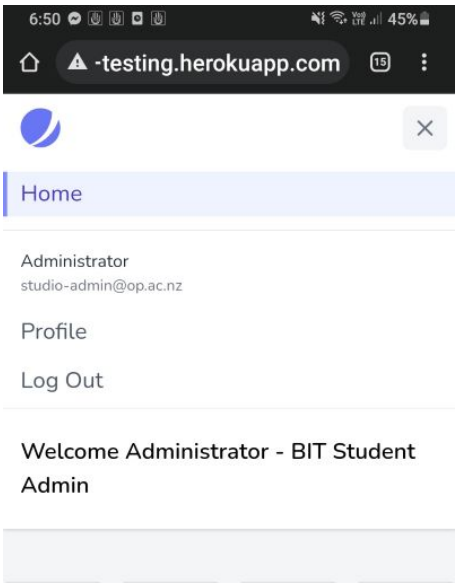
1.1 Contribute to development of new product features

Front End Development

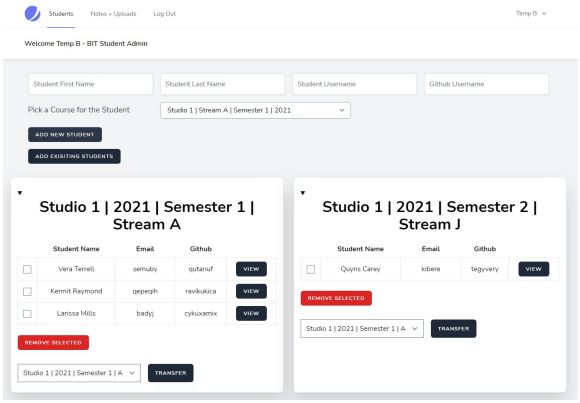
Mitchell's as the developer, had a lot of work to do to resolve important issues such as security, and development of new features. Some of these changes were complex, and time consuming; So that Mitchell was not overloaded with both front and backend development, I picked up some development tasks such as completing UI changes on new pages for new features so that Mitchell could continue developing other product backlog items that had a higher priority. By doing this I contributed to implementation the client requested changes across the site.

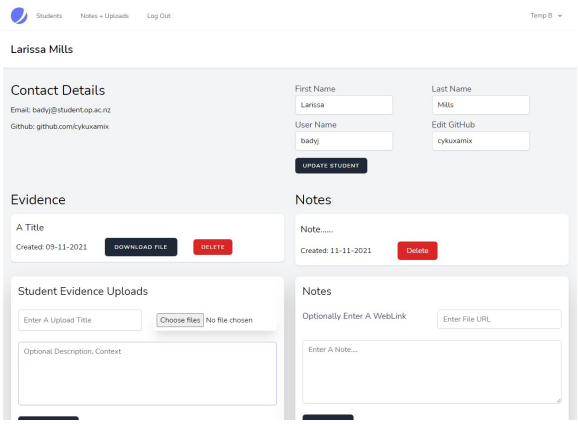
- Sprint 2 - #106(sprint-2-student-controller)
- Sprint 2 - #109(sprint-2-jetstream-templates)
- Sprint 3 - #141(sprint-3-update-homepage)
- Sprint 4 - #184(sprint-4-student-update-form)
- Sprint 4 - #185(sprint-4-student-status)
- Sprint 4 - #196 (sprint-4-student-update-form)
- Sprint 4 - #212 (sprint-4-admin-panel-deployment-fixes)
- Sprint 5 - #241 (sprint-5-homepage-layout-update)

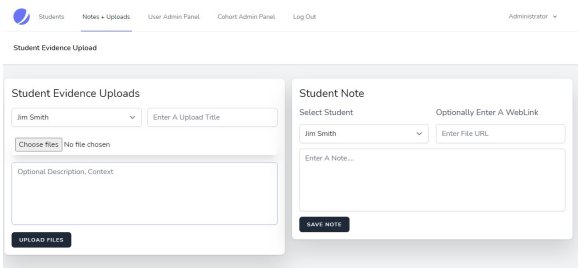
- Sprint 5 - #250 (sprint-5-homepage-layout-update)
- Sprint 6 - #269 (sprint-6-layout-bug-fix)

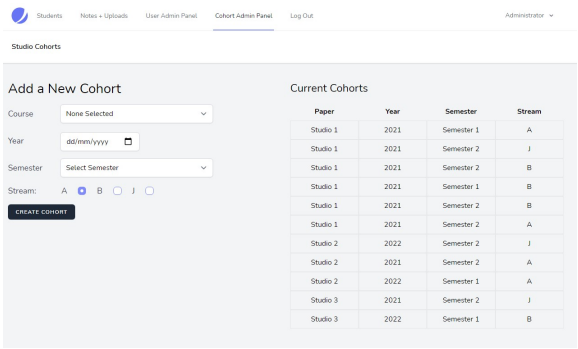


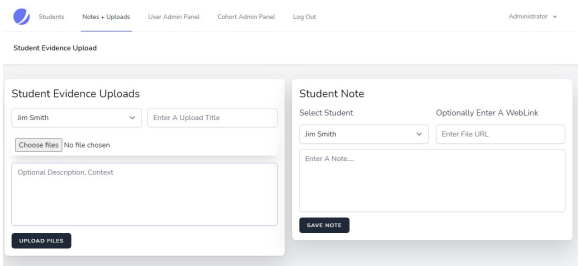
Final UI Changes

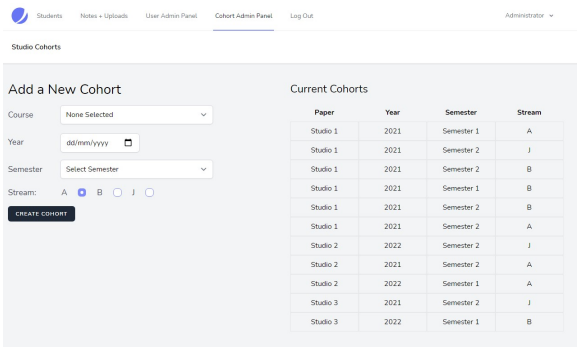


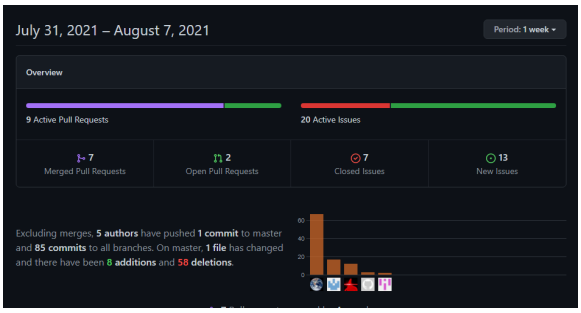


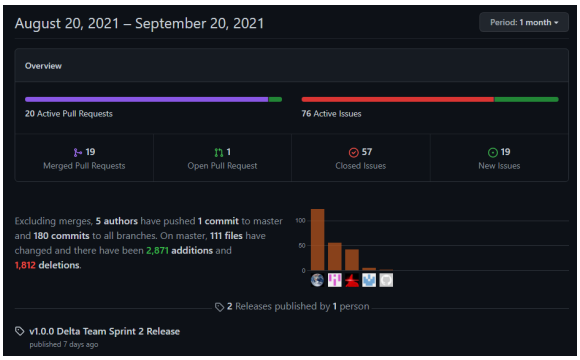


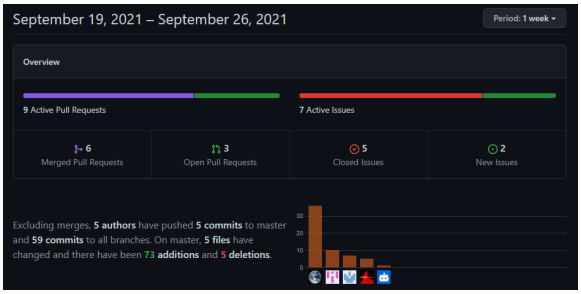


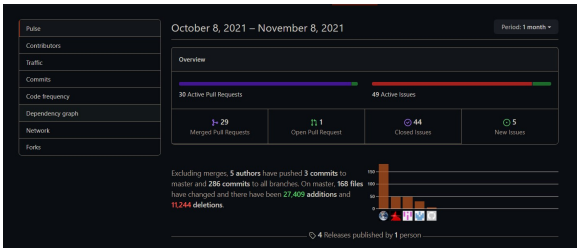






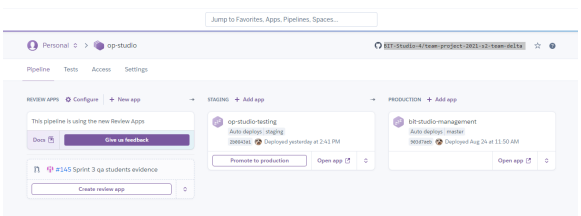


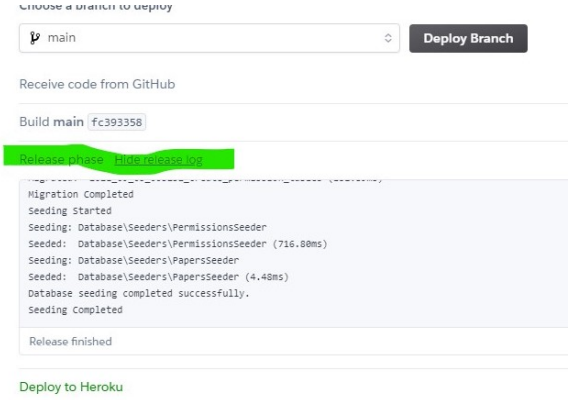




1.2 Contribute to Project Deployment

1. Set-Up the pipeline configuration on Heroku, linked this back to the repo at the start of the semester
2. Configured, tested release script and Procfile to run migrations automatically on both Heroku applications. This previously needed to be done manually each time a pull request was merged to the staging and master branches.





3. During sprint 3, I discovered that Heroku was serving content from a `main` branch, this meant that when we

deployed, code was being pushed to a `master` branch and was not getting served; to resolve this I had to merge `master` into the `main` branch, resolve merge conflicts and recompile CSS and JS files locally before pushing back to the Heroku repository.

- Pull Request #164 - sprint-3-deployment-fix
- Pull Request #168 - sprint-3-deployment-fix

1.4 Demonstrate Improvements in Applying Agile Project Management

1. At the end of sprint 2, we had run behind schedule, and needed to begin prepartation for sprint 3. We made the mistake of deploying to production, without first getting final signoff.
    - At this point much stricter branch rules were put in place to avoid this occurring again.
    - This mistake caused us to think much more critically at our product backlog in the sprint planning stage.
2. **Client feedback** During sprint 3, the team had a review with the product owner to get feedback on changes in the UI layout and colour scheme. Feedback received was immediately acted on rather than being recorded as a user story, and a action plan for the next sprint.
    - this mistake reinforced the need to note the users feedback in the form of user stories
    - plan/propose changes; add to the product backlog, and get further feedback from the client before proceeding with any new feature or change.

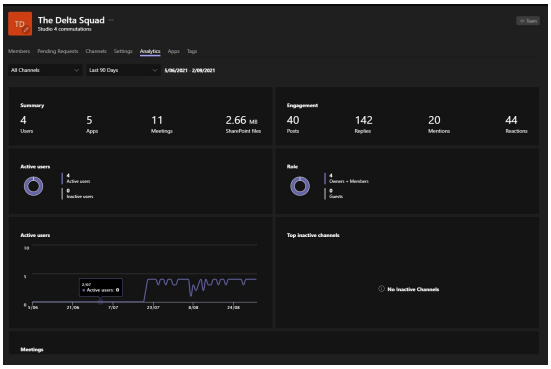
3. **User stories** As a team we regularly got feedback about our user stories, their lack of visibility in the product backlog, and the need to keep these relevant. Initially we had our user stories separate from our product backlog items, instead linking these using a hash tag. By sprint 3, it became clear doing this made our user stories stale and our product back hard to follow, even for us. At this point all the stale user stories were closed, and the current product backlog updated with current user stories. This immediately made the point of each item much clearer and easier to update and get feedback.
4. **Code Reviews** - Sprints 3 & 4 both ran over time. Sprint 4 we took more care in the planning our product backlog and targets for completing allowing enough time at the end to get feedback, and plan for sprint 5. This did not solve the issue, and the sprint still ran over time. One issue that became crystal clear was that the code review process had no definitive end point, and this resulted in some pull requests remaining open and unreviewed for four days in some cases. Despite reminders in class, on teams, during stand up meetings, automated reminders and tagged comments in the pull requests themselves, if the code reviews were not done, nothing would progress. Mitchell more than anyone was negatively affected by this bottleneck, because his changes impacted other backlog items that he needed to move on to, but couldn't until issues were resolved in the PR.

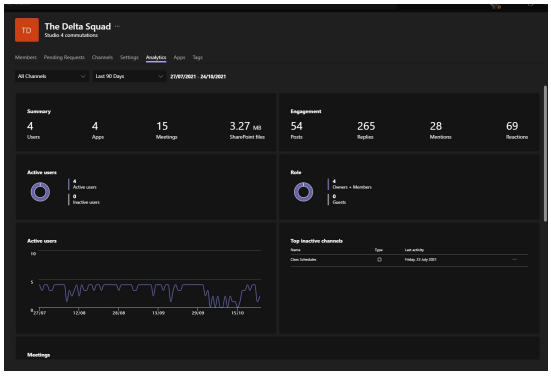
   To eliminate this issue, and place more urgency on completing reviews I proposed some rules around code reviews for sprint 5, trial how this improved our workflow and assess it and the end in the retrospective. The main rule being that pull requests exist for a maximum of 48 hours, if

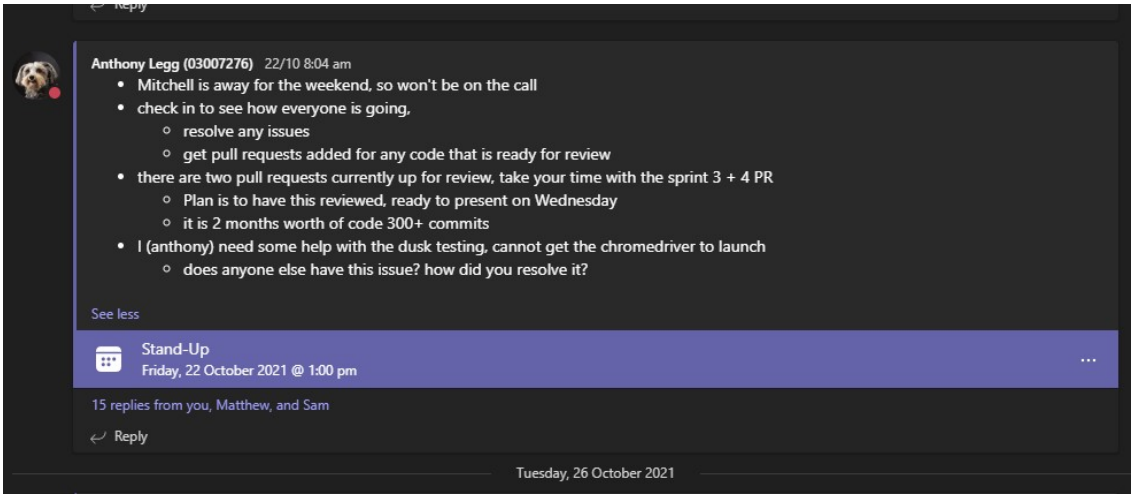
there are no change requests and no test failures the branch would be merged. For the most part, this worked perfectly. Two new issues came up, one was with the test suite failing more often due to the speed of changes, the other with development branches needing updating more often.

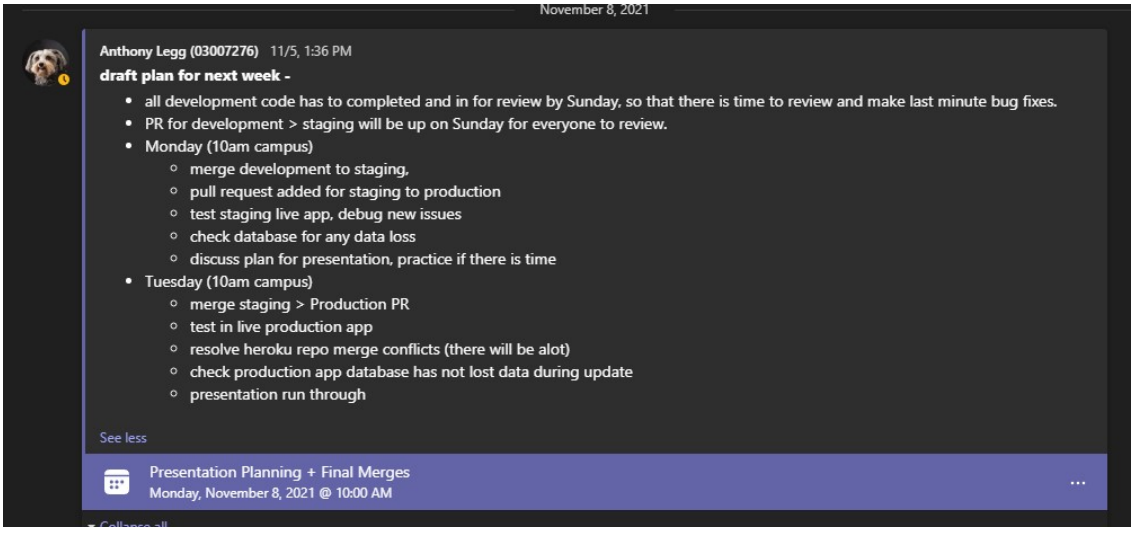
5. Project Discussion Pages share resources, propose ideas/solutions, sprint plans.
6. **Team Engagement**
    - stand up meetings scheduled at regular times each day; consistency made it easy for everyone to plan for and attend these meetings.
    - From sprint 3, I noticed that everyone on the team was putting more into sprint planning and retrospectives.
    - In sprint 5, some bad habits started to re-appear, I needed remind everyone to follow the agreed to in the project board. A number of new backlog items were getting added and developed mid-sprint without

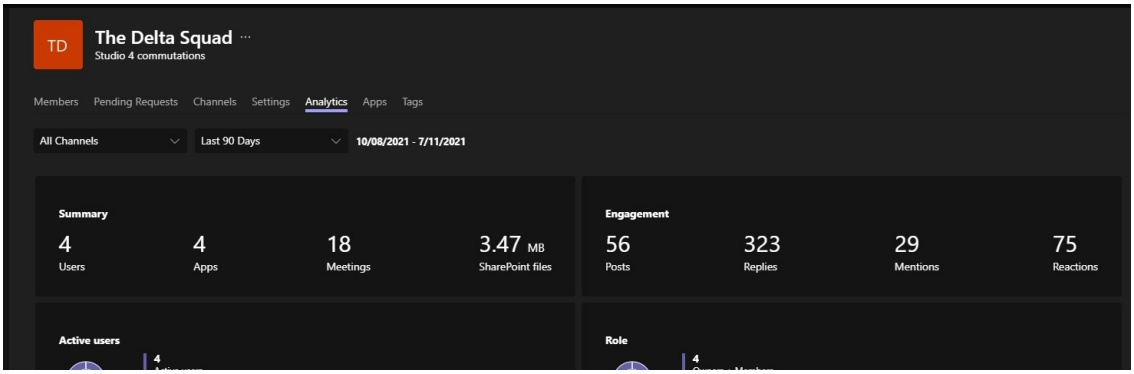
discussion with the rest of the team, or the product owner.

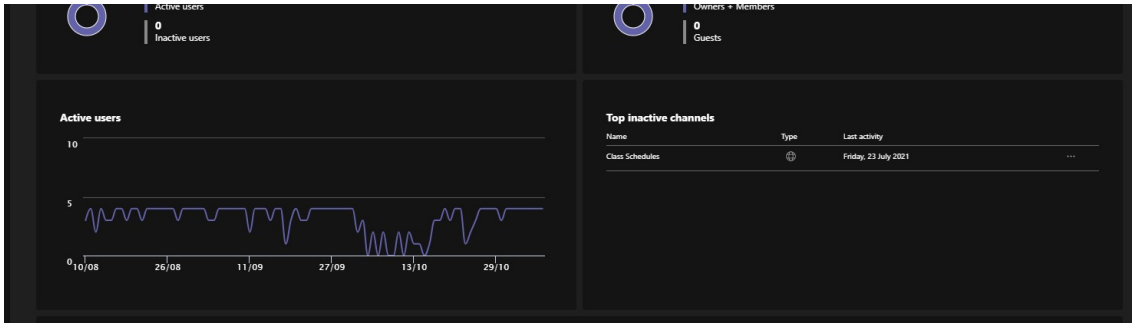










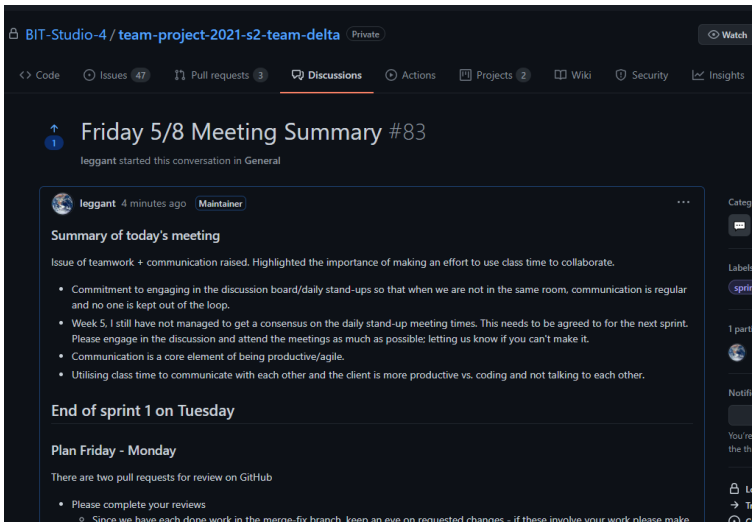


1.5 Use industry-standard communication and project management tools in a professional manner

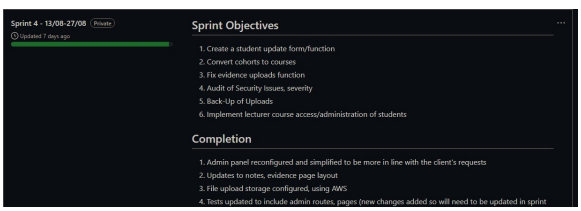
I cannot say that my communication on Teams was always 100% professional. There were several instances that I had to speak up and be brutally honest, because I could see that there was a casual attitude which was impacting how we were functioning as a team. For instance, our product backlog could be altered at anytime without discussion, that the sprint could be extended beyond two weeks if we were running late, that we could work separately without affecting others, that stand-ups were optional and that code reviews were ok to leave hanging for four days.

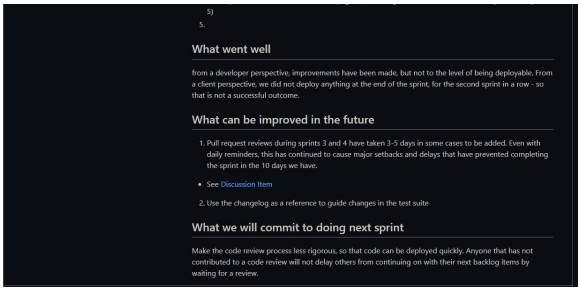
1. Project Release Executive Summaries
    - Sprint 5 Pre-Release
    - Sprint 4 Pre-Release
    - Sprint 3 Pre-Release
2. GitHub Discussion Boards
    - Stand-Up Meeting Summaries provided on the teams channel and GH discussion boards during the first half of the semester. I found that attendance at stand-up

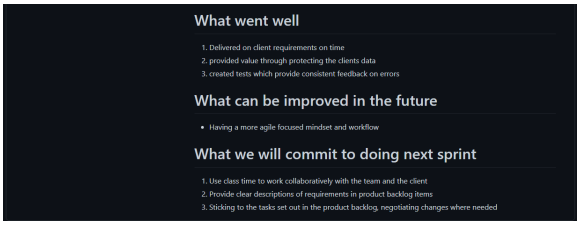
meetings got a lot more consistent, and that writing these notes had no longer had any impact as no one needed to read them.
    - Guides and Resources
    - Security Audit
    - Project Releases
    - Workflow Changes seek feedback on proposed changes resulting from sprint retrospectives
3. Teams Channel - Stand Up Meetings + Agendas
    - In place of meeting summaries, I found it more useful to set an agenda for our stand-ups in addition to checking in with everyone on how they were tracking, resolving issues, updating the project board.

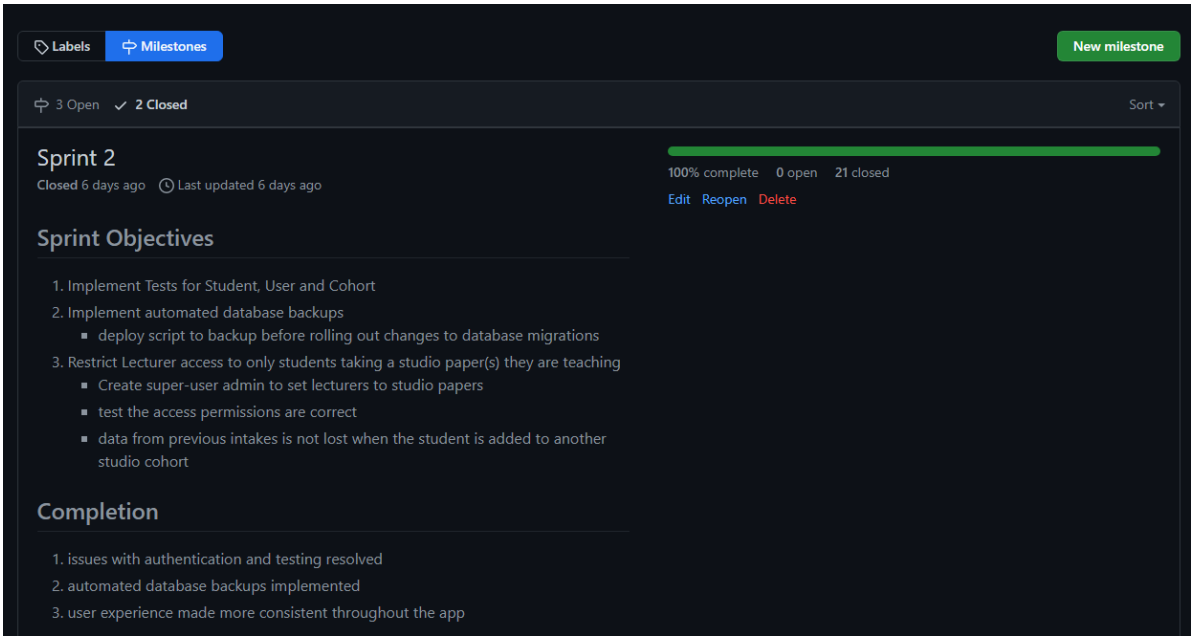


4. Repo Changelog - Project Transparency + Documentation
   Used in project release documentation
5. Repo Commitlog- For QA Reference, Test Maintenance and Development




5)
5.
What went well
from a developer perspective, improvements have been made, but not to the level of being deployable. From a client perspective, we did not deploy anything at the end of the sprint, for the second sprint in a row - so that is not a successful outcome.
What can be improved in the future
1. Pull request reviews during sprints 3 and 4 have taken 3-5 days in some cases to be added. Even with daily reminders, this has continued to cause major setbacks and delays that have prevented completing the sprint in the 10 days we have.
• See Discussion Item
2. Use the changelog as a reference to guide changes in the test suite
What we will commit to doing next sprint
Make the code review process less rigorous, so that code can be deployed quickly. Anyone that has not contributed to a code review will not delay others from continuing on with their next backlog items by waiting for a review.


What went well
1. Delivered on client requirements on time
2. provided value through protecting the clients data
3. created tests which provide consistent feedback on errors
What can be improved in the future
• Having a more agile focused mindset and workflow
What we will commit to doing next sprint
1. Use class time to work collaboratively with the team and the client
2. Provide clear descriptions of requirements in product backlog items
3. Sticking to the tasks set out in the product backlog, negotiating changes where needed


Labels
Milestones
New milestone
3 Open
2 Closed
Sort
Sprint 2
Closed 6 days ago
Last updated 6 days ago
100% complete
0 open
21 closed
Edit
Reopen
Delete
Sprint Objectives
1. Implement Tests for Student, User and Cohort
2. Implement automated database backups
▪ deploy script to backup before rolling out changes to database migrations
3. Restrict Lecturer access to only students taking a studio paper(s) they are teaching
▪ Create super-user admin to set lecturers to studio papers
▪ test the access permissions are correct
▪ data from previous intakes is not lost when the student is added to another studio cohort
Completion
1. issues with authentication and testing resolved
2. automated database backups implemented
3. user experience made more consistent throughout the app

Learning Outcome 2

> Analyse and manage development challenges to create production-quality outputs.

2.1 Contribute to Automated Test Suite

GitHub Dusk Testing

Work began on this in sprint 1, was functional at the end of sprint 3 but was not able to be reliably utilised until sprint 5, when the dusk tests were functioning correctly both locally and in the GH actions container. This is now configured as a branch guard on the master and staging branches. Every commit and pull request on the repository is run tested.

1. Sprint 1 #57 - sprint-1-laravel-deployment-testing
2. Sprint 2 #104 - sprint-2-github-deployment-actions
3. Sprint 2 #121 - sprint-2-github-deployment-actions
4. Sprint 3 #144 - sprint-3-github-dusk-action
5. #145 - Code Review
6. Sprint 3 #168 - sprint-3-deployment-fix
7. Sprint 4 #182 - sprint-4-github-action-updates
8. Sprint 4 #191 - sprint-4-auto-changelog
9. Sprint 4 #218 - sprint-4-github-action-updates
10. Sprint 5 #235 - logout security testing

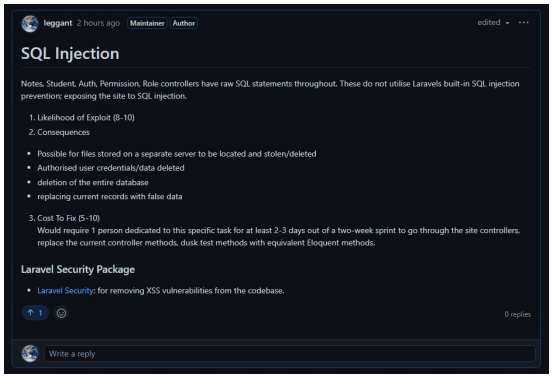
Code Linter Test

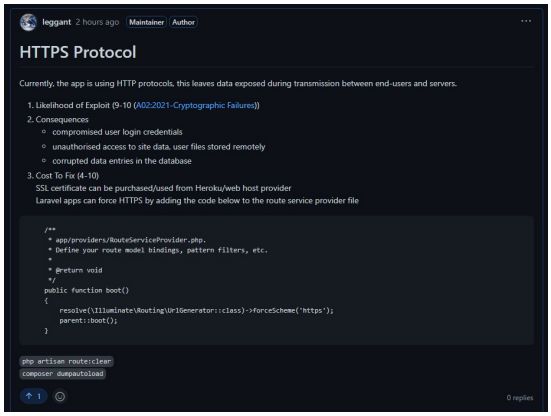
Configured a code linter check GitHub action to run on every commit and set this as a branch rule/guard on production, staging and development branches.

```
Time: 3 secs    Memory: 4.0 MiB Cache: No

FAILURES!
Files: 225, Failures: 1

There was 1 errors:
1. /github/workspace/app/Http/Controllers/StudentController.php:138
    135|            'first_name' => 'alpha|max:25|min:3',
    136|            'last_name' => 'alpha|max:25|min:3',
    137|            'username' => ['required', 'max:10', 'alpha_num', Rule::unique('student')->ignore($id)],
  > 138|            'github' => 'alpha_dash| 'github' => 'alpha_dash|Rule::unique('students', 'github')->ignore($upstudent->id)|nullable|max:15',|nullable|max:15',
    139|            'cohort_id' => 'nullable|integer',
    140|        ];
    141|
 unexpected 'github' (T_STRING), expecting ']' in line 138
```

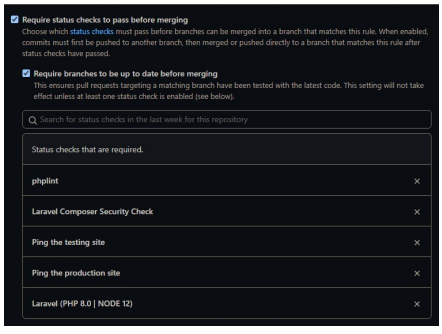
2.2 Contribute to Project Security

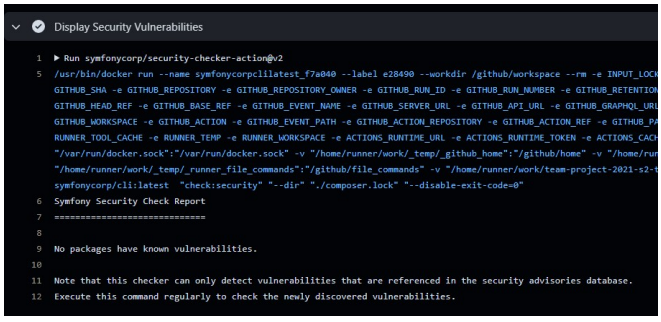
- Security Audit





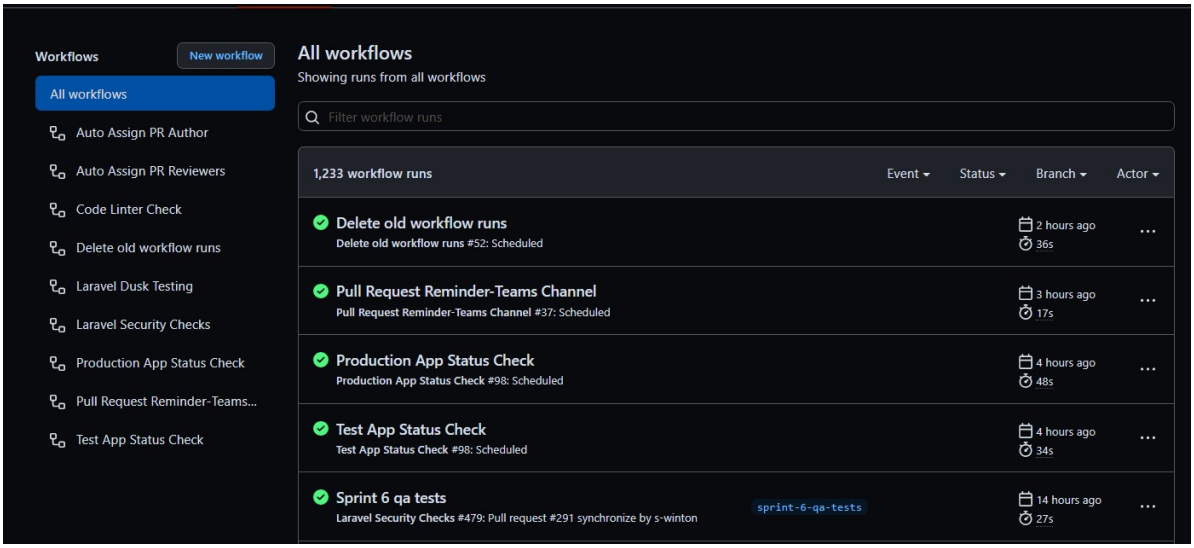
- Student Evidence Uploads #127
- Student Evidence Upload Backup #128
- Evidence Security: Test S3 Bucket
- Code Review #75 - Publish Backup VM details in Readme
- Code Review #211 - private keys published to repo
- Composer Package Security Check GH Action - Set this as a branch rule/guard on production and staging branches.

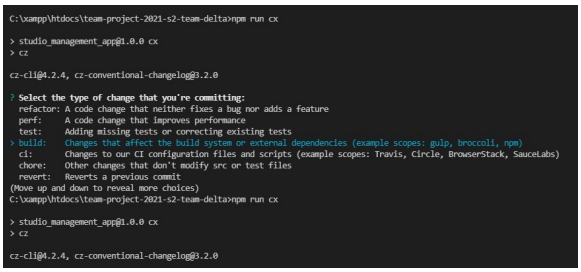


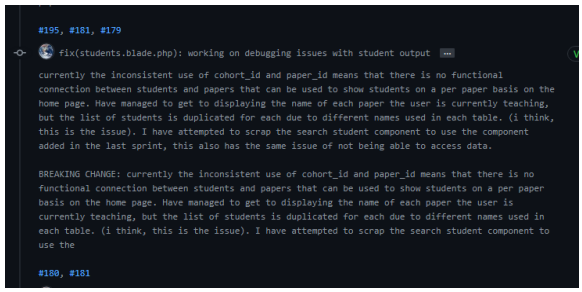


2.3 Contribution to Project Automation

In addition to the Heroku release script, automated dusk and linter tests, I also created the 7 other GH actions in the repository, prettier code formatter, changelog generator, commitizen git commit linter, and the project kanban board (template) automation, making a template version that could be copied with automation settings at the start of each sprint.

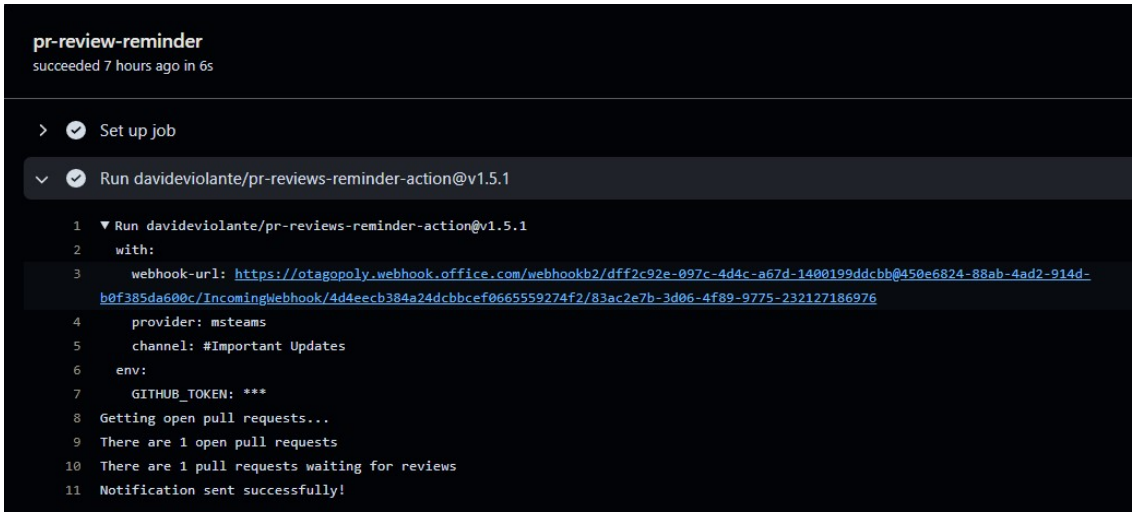


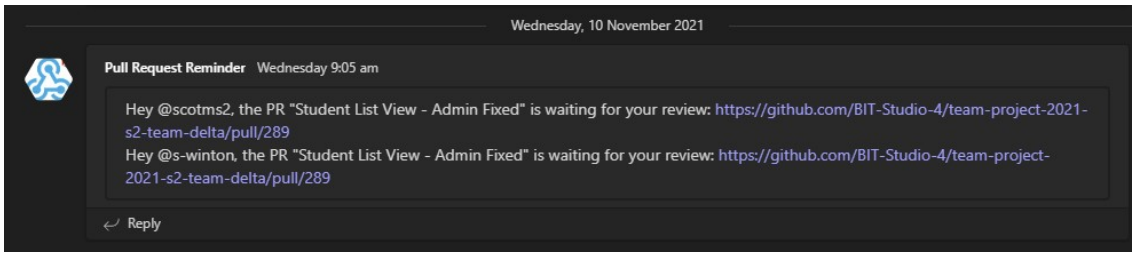




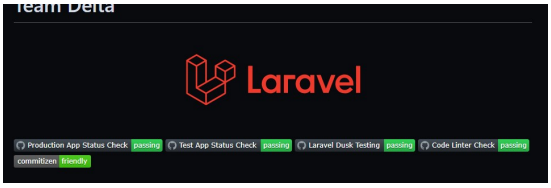
1. Automated assignment of pull request reviewers and assignees - this was to stop this being forgotten about.

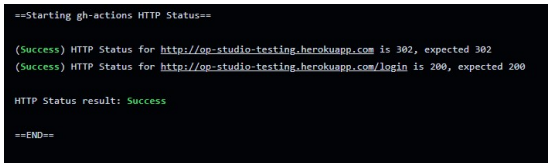
2. Configured GitHub Action to send reminders to PR reviewers if there are reviews that have not been completed. This is configured to send reminders to the project's teams channel using a web hook.





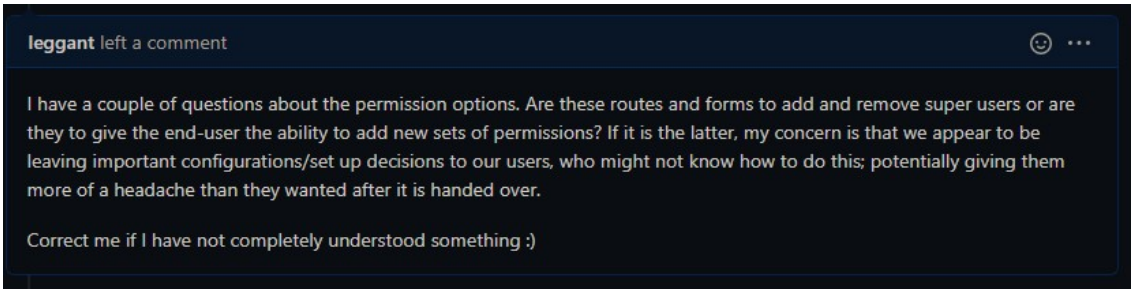
3. Configured Prettier formatter, to scan and format project code locally. This included writing and testing the configuration and ignore files to ensure this automation did not format any code that was not written by the project team.
4. Actions added that check the HTTP response from the live test the production apps if the live test app is down, a branch guard blocks merging to production until the issue causing the error is resolved.

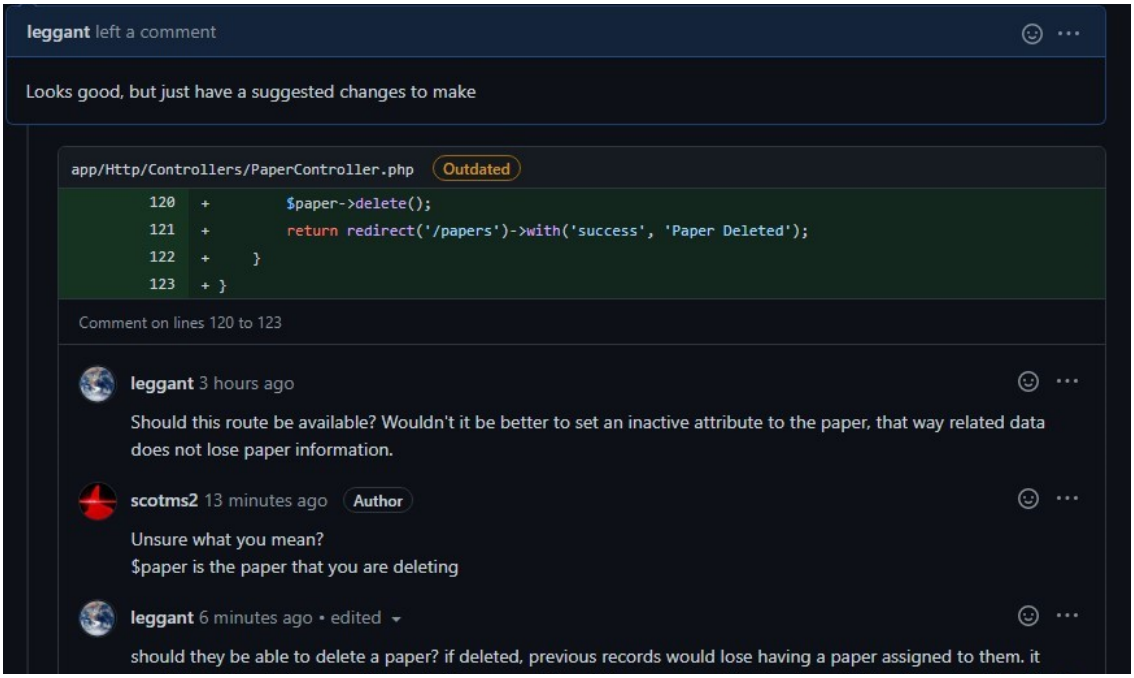




2.4 Participate in Code/Solution Review to Ensure High-Quality Outputs

1. Pull Request #75 - Review 1 + Review 2
2. Pull Request #134 - Review 1 + Review 2
3. Pull Request #198 - Review 1 + Review 2
4. Pull Request #211 - Review 1 + Review 2
5. #243 - dusk test suite
6. Full Code Review List





would be better to have an attribute in the paper stating if they are currently being taught. Something to raise with the client later today :)

Reply...

Resolve conversation

Sprint 3 Dashboard Update #141

Merged leggant merged 5 commits into staging from sprint-3-update-homepage 13 hours ago

Conversation 3 Commits 5 Checks 4 Files changed 11

leggant commented 3 days ago • edited

Changes Close

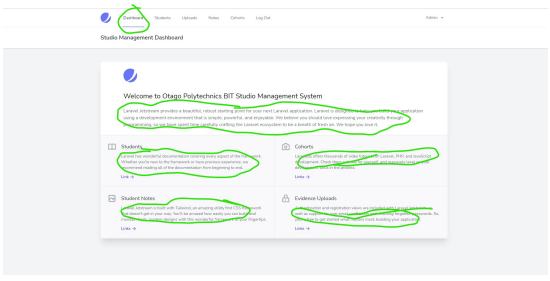
- Closes Create Resource Controllers- Transfer Functions #5
- Closes Update Routes File #19
- Closes Dashboard Route Update #137
- Closes API Controller - Resource Controllers #21

Changes made:

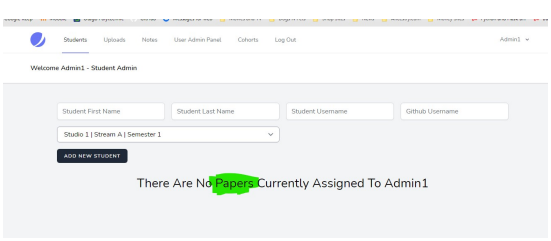
1. altered the dashboard; removed default Laravel content so this is not present when the dashboard is re-activated in future.
2. changed the home route to the student index page; commented out the Dashboard route in the nav and web route file

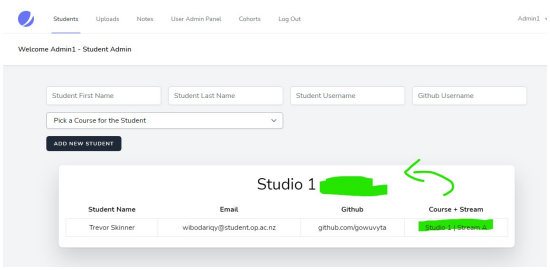
Outside of scope

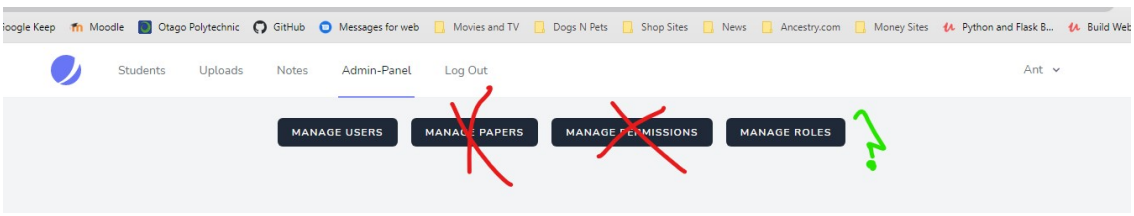
3. Added a Note resource controller, transferred functions from api and pages controller
4. updated all other controllers with methods from the api and pages controller
5. changed the web routes to use these controller methods
6. changed the form routes in cohort, note and evidence pages



Paper	Year	Semester	Stream
Studio 1	2023-01-01	Semester 2	J
Studio 1	2023-01-01	Semester 1	B
Studio 1	2022-01-01	Semester 1	A
Studio 1	2023-01-01	Semester 2	J
Studio 2	2023-01-01	Semester 1	A
Studio 2	2023-01-01	Semester 1	A
Studio 2	2022-01-01	Semester 2	A
Studio 2	2022-01-01	Semester 1	B
Studio 2	2022-01-01	Semester 1	B







Respond to feedback to produce high quality outputs

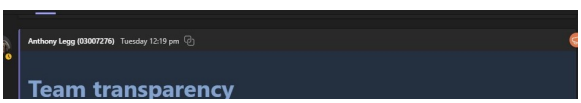
1. Sprint 4 - Auto Changelog Script was added in response to feedback given on our teams transparency and progress. The changelog output was included in the project release information and a commitlog generused in
2. Dashboard Route Update + Pull Request #141
3. Student Status + Pull Request #185

Project Feedback

As mentioned above, feedback on our user stories, particularly during the first three sprints was that they were not visible, lacked connection with the product backlog items, and were not kept up to date with feedback from the client. By separating our user stories and product backlog tickets our user stories became stale, our product back hard to follow, and project work lacked clearly defined goals. This feedback pushed us to close all the stale user stories, and refresh the current product backlog with current user stories. This immediately made the point of each item much clearer and easier to update and get feedback.

User Experience/User Stories

- User Story: Set Up #209
- User Story: No Guessing Games #208
- User Story: Less Clicks #207

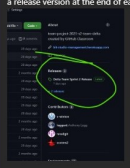


Team Transparancy

Sorry guys, one thing I keep forgetting to bring up from the scrum meeting last week. Elise and Martin highlighted that there is not enough transparency from each team back to the product owner (Elise) or the boss (Martin). There are several ways we can resolve this; through using a section on the repo discussion board, through progress updates on the product backlog ticket, tagging the product owner on items, adding a detailed summary of the changes made in the sprint; I think another useful and automated way we could do this is through using a changelog markdown file in the repo. This can be automatically generated, but relies on the information provided in pull requests, commit messages etc.

Progress updates in the ticket are probably going to be the most useful; they will get you to go back to the plan, check off completed items and remind you of what needs to be done next. This can then get translated over to the pull request, changelog etc. Updates don't need to be huge; done regularly they won't be difficult to add to.

In addition to the above, I will also be adding a release with a summary at the end of each sprint; these are more for Martin, so that he can see a release version at the end of each sprint



Let me know what you think, how you want to tackle this. It's late in the game, so we do need to get this into our workflow starting asap.

See less

Reply

Keep a Changelog
Keep a Changelog
keepachangelog.com

Reply

Today

Matthew Revell (1000073025) 1:39 pm
Just noticed that the email input box is missing from the add student form. The username/id is being displayed as the email address.

2 replies from you

Matthew Revell (1000073025) 1:54 pm
it's fine just needed to change the test for adding a student

Anthony Legg (03007276) 1:58 pm
yes..... we really need to address this workflow issue aye. You aren't getting enough information to keep the test suite up to date.

everything changes across multiple branches each sprint, its not something you can easily track on your own.'

Matthew Revell (1000073025) 2:03 pm
I do expect to modify or remove tests to fit the app after each sprint but having to discover some for myself - maybe the changelog could help with that too.

Anthony Legg (03007276) 2:04 pm
actually that's a really good solution!!

2:07 pm
👍 1
but does rely on quality commits - something that should be helped along with everyone using the commitizen package that is getting added in one of my current PRs.

Would you need to full run of commits or just the pull request merges?

changelog can include both but it gets ALOT of output