

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.

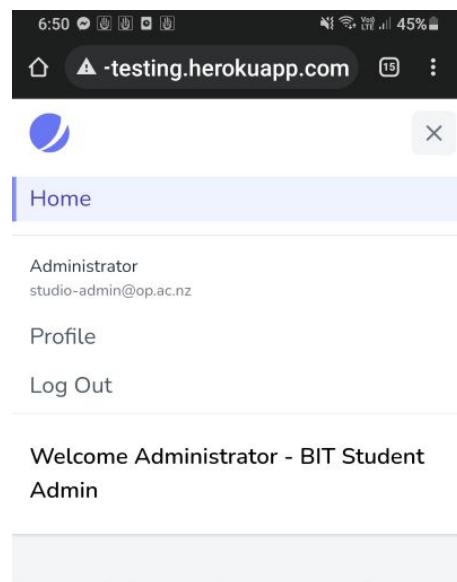
1.1 Contribute to development of new product features

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

Contact Details

Email: bady@student.op.ac.nz
GitHub: github.com/cykuamix

Evidence

A Title
Created: 09-11-2021 DOWNLOAD FILE DELETE

Notes

Note.....
Created: 11-11-2021 Delete

Add a New Cohort

Course: None Selected
Year: dd/mm/yyyy
Semester: Select Semester
Stream: A B C D
CREATE COHORT

Current Cohorts

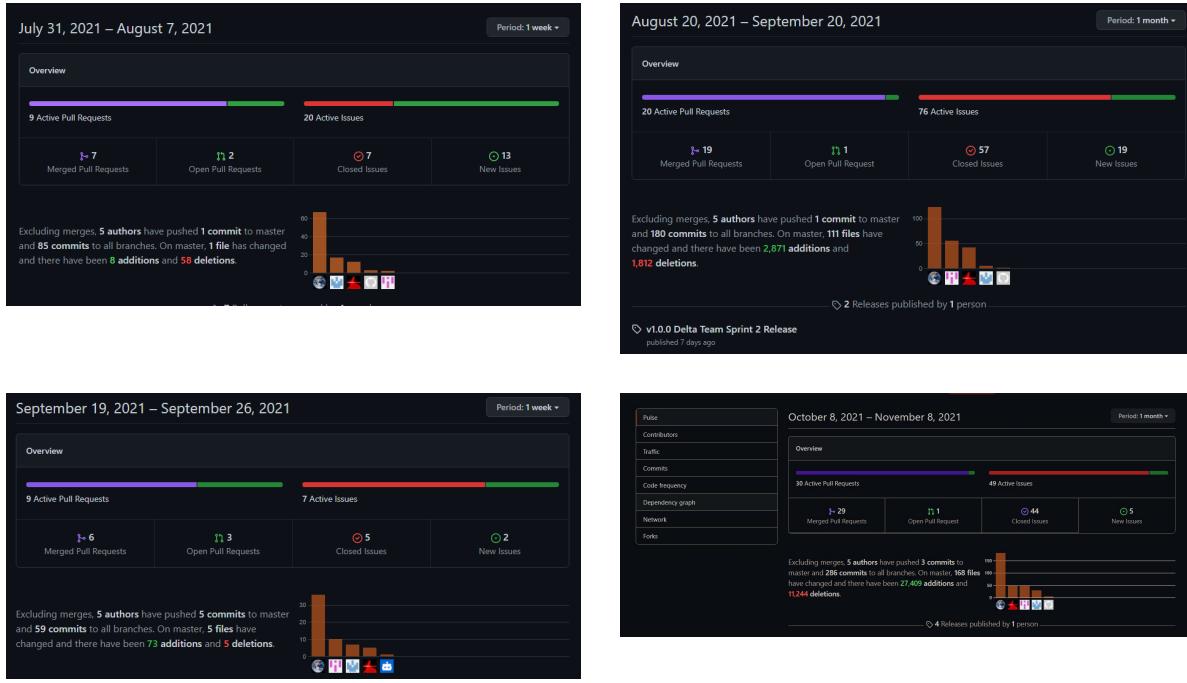
| Paper | Year | Semester | Stream |
|----------|------|------------|--------|
| Studio 1 | 2021 | Semester 1 | A |
| Studio 1 | 2021 | Semester 2 | J |
| Studio 1 | 2021 | Semester 2 | B |
| Studio 1 | 2021 | Semester 1 | B |
| Studio 1 | 2021 | Semester 2 | B |
| Studio 1 | 2021 | Semester 2 | A |
| Studio 2 | 2021 | Semester 2 | J |
| Studio 2 | 2021 | Semester 2 | A |
| Studio 2 | 2022 | Semester 1 | A |
| Studio 3 | 2021 | Semester 2 | J |
| Studio 3 | 2022 | Semester 1 | B |

Add a New Cohort

Course: None Selected
Year: dd/mm/yyyy
Semester: Select Semester
Stream: A B C D
CREATE COHORT

Current Cohorts

| Paper | Year | Semester | Stream |
|----------|------|------------|--------|
| Studio 1 | 2021 | Semester 1 | A |
| Studio 1 | 2021 | Semester 2 | J |
| Studio 1 | 2021 | Semester 2 | B |
| Studio 1 | 2021 | Semester 1 | B |
| Studio 1 | 2021 | Semester 2 | B |
| Studio 1 | 2021 | Semester 2 | A |
| Studio 2 | 2021 | Semester 2 | J |
| Studio 2 | 2021 | Semester 2 | A |
| Studio 2 | 2022 | Semester 1 | A |
| Studio 3 | 2021 | Semester 2 | J |
| Studio 3 | 2022 | Semester 1 | B |



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.

The figure contains two screenshots of the Heroku web interface.

- Pipeline Configuration:** Shows the 'Pipeline' tab with three stages: REVIEW APPS, STAGING, and PRODUCTION. The REVIEW APPS stage has a 'Give it a try!' button. The STAGING stage shows an app named 'opp-studio-testing' with a 'Deploy' button. The PRODUCTION stage shows an app named 'opp-studio-management' with a 'Deploy' button. A note says 'This pipeline is using the new Review Apps Dev'.
- Deployment Log:** Shows the 'Deploy main' branch. It includes a 'Receive code from GitHub' section, a 'Build main fc393358' section, and a 'Release phase Hide release log' section. The log output shows migration and seeding completed successfully. A 'Release finished' message and a 'Deploy to Heroku' button are also present.

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 preparation 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.

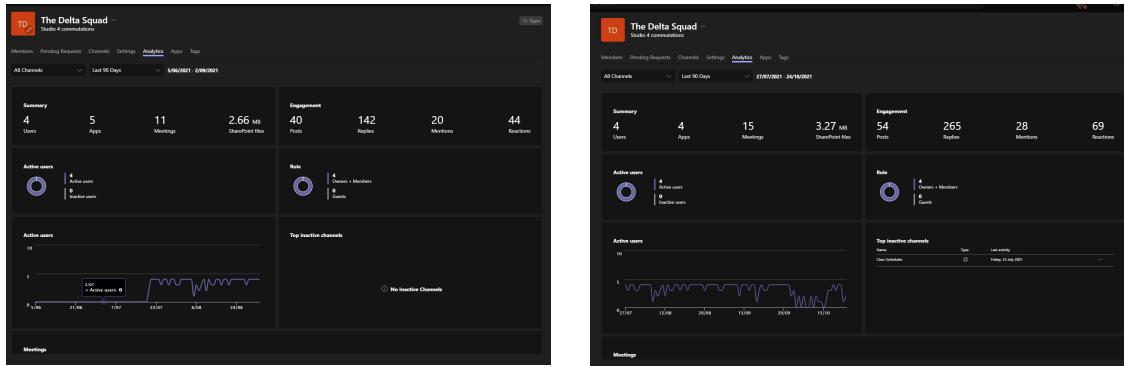
The screenshot shows a GitHub discussion thread. The title is "Delivery + Code Reviews of Pull Requests #217". It is marked as "Unanswered" and was asked by "leggant" 17 seconds ago. The post is labeled "Maintainer". The content discusses pull request reviews taking 3-5 days during sprints 3 and 4, causing delays. It proposes a new approach where reviews are limited to 48 hours, and if a PR has at least one review, it can be merged even if there are remaining changes. It also suggests a max of 24 hours + min of 2 reviews for the development >> staging branch. A comment from @scotms2 and @revellgit (@s-winton) expresses support for this approach. The post has 1 upvote and 0 comments.

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.



Anthony Legg (03007276) 22/10 8:04 am

- Mitchell is away for the weekend, so won't be on the call
- check in to see how everyone is going.
 - resolve any issues
 - get pull requests added for any code that is ready for review
- there are two pull requests currently up for review, take your time with the sprint 3 + 4 PR
 - Plan is to have this reviewed, ready to present on Wednesday
 - it is 2 months worth of code 300+ commits
- I (anthony) need some help with the dusk testing, cannot get the chromedriver to launch
 - does anyone else have this issue? how did you resolve it?

[See less](#)

Stand-Up
Friday, 22 October 2021 @ 1:00 pm

15 replies from you, Matthew, and Sam

Reply

Tuesday, 26 October 2021

Anthony Legg (03007276) 11/5, 1:36 PM

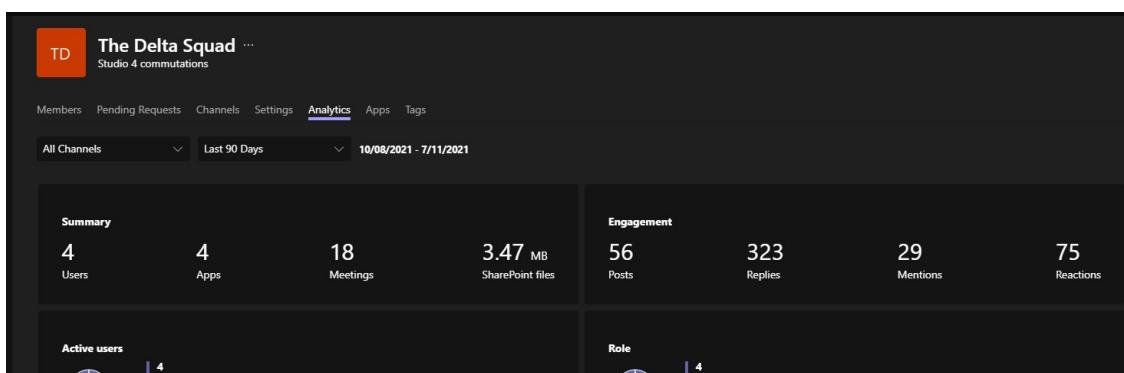
draft plan for next week -

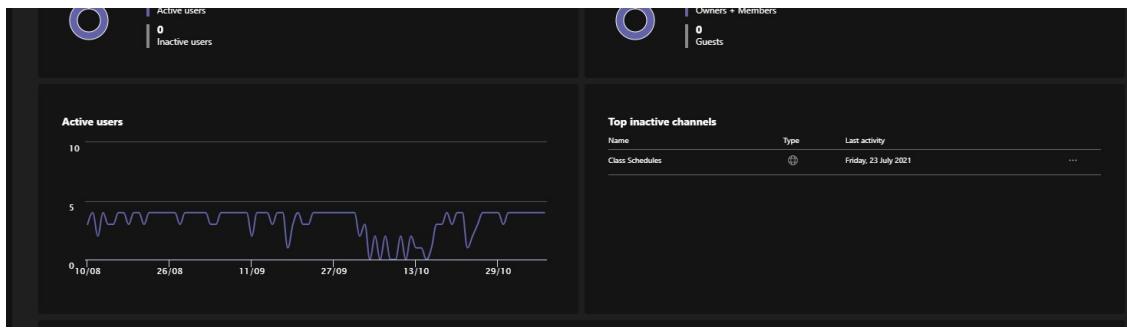
- all development code has to completed and in for review by Sunday, so that there is time to review and make last minute bug fixes.
- PR for development > staging will be up on Sunday for everyone to review.
- Monday (10am campus)
 - merge development to staging,
 - pull request added for staging to production
 - test staging live app, debug new issues
 - check database for any data loss
 - discuss plan for presentation, practice if there is time
- Tuesday (10am campus)
 - merge staging > Production PR
 - test in live production app
 - resolve heroku repo merge conflicts (there will be alot)
 - check production app database has not lost data during update
 - presentation run through

[See less](#)

Presentation Planning + Final Merges
Monday, November 8, 2021 @ 10:00 AM

...
Collapse all





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.

The screenshot shows a GitHub repository named 'BIT-Studio-4 / team-project-2021-s2-team-delta' with a private status. The 'Discussions' tab is selected. A post titled 'Friday 5/8 Meeting Summary #83' by 'leggant' (4 minutes ago) is displayed. The post contains a summary of the meeting, noting the importance of teamwork and communication. It lists several items for the next sprint, including committing to engagement in daily stand-ups, getting a consensus on meeting times, and utilizing class time for communication. Below the summary, there's a section for the 'End of sprint 1 on Tuesday' and a 'Plan Friday - Monday' section, which mentions two pull requests for review. The sidebar on the right shows repository statistics: 1 part, 1 note, and 0 files.

4. [Repo Changelog - Project Transparency + Documentation](#)

Used in project release documentation

5. [Repo Commitlog- For QA Reference, Test Maintenance and Development](#)

The image displays two GitHub repository screenshots. The left repository, 'Sprint 4 - 13/08-27/08 (Private)', has an 'Updated 7 days ago' status. Its 'Sprint Objectives' section lists tasks such as creating a student update form/function, fixing evidence uploads, and back-up of uploads. The 'Completion' section lists updates to admin routes, evidence page layout, file upload storage configuration, and tests updated to include admin routes. The right repository, 'Sprint 1 - 20/07 - 10/08 (Private)', has an 'Updated 12 days ago' status. Its 'Sprint Objectives' section includes creating a working deployment, securing data access, developing tests for students and navigation, and ensuring project accessibility. The 'On Completion' section details the deployment requirements, data access, test development, and site navigation, along with a note about retaining access through user credentials.

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 less time to work collaboratively with the team and the client
 2. Provide clear descriptions of requirements in product backlog items
 3. Stick to the tasks set out in the product backlog, registering changes where needed

Milestones

3 Open 2 Closed

Sprint 2
Closed 6 days ago Last updated 6 days ago

New milestone

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.

```

15 Time: 3 secs    Memory: 4.0 MiB Cache: No
16
17 FAILURES!
18 Files: 225, Failures: 1
19
20 There was 1 errors:
21  1. /github/workspace/app/Http/Controllers/StudentController.php:138
22    135|         'first_name' => 'alpha|max:25|min:3',
23    136|         'last_name' => 'alpha|max:25|min:3',
24    137|         'username' => ['required', 'max:10', 'alpha_num', Rule::unique('student')->ignore($id)],
25 > 138|         'github' => 'alpha_dash|github' => 'alpha_dash|Rule::unique('students', 'github')->ignore($upstudent->id)|nullable|max:15',|nullable|max:15',
26    139|         'cohort_id' => 'nullable|integer',
27    140|     ];
28    141|
29 unexpected 'github' (T_STRING), expecting '[' in line 138

```

2.2 Contribute to Project Security

- [Security Audit](#)

The image shows two GitHub pull request screenshots. The left one is titled "SQL Injection" and the right one is titled "HTTPS Protocol". Both pull requests have been updated 2 hours ago by the maintainer. The "SQL Injection" pull request discusses raw SQL statements and Laravel Security Package. The "HTTPS Protocol" pull request discusses SSL certificates and route provider code.

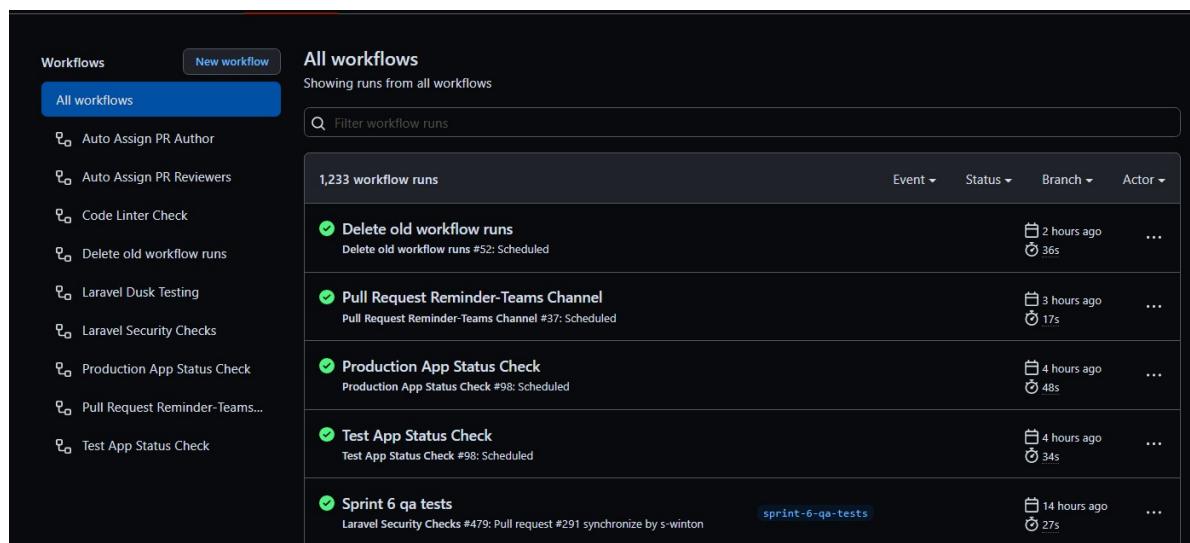
- [Student Evidence Uploads #127](#)
- [Student Evidence Upload Backup #128](#)
- [Evidence Security: Test S3 Bucket](#)
- [Code Review #75 - Publish Backup VM details in Readme](#)
- [Code Review #211 - private keys published to repo](#)
- [Composer Package Security Check GH Action](#) - Set this as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

The image shows a GitHub repository settings page for "Status checks that are required". It includes a search bar for status checks from the last week and a list of required checks: phplint, Laravel Composer Security Check, Ping the testing site, Ping the production site, and Laravel (PHP 8.0 | NODE 12).

The image shows a terminal window with the command "Run symfony/corp/security-checker-action@v2" and its output. The output includes Docker run commands, environment variables, and a note about the checker's limitations. The final part of the output is a "Symfony Security Check Report" which states "No packages have known vulnerabilities".

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.



The screenshot shows the GitHub Workflows interface. On the left, there's a sidebar with various workflow categories like 'Auto Assign PR Author', 'Code Linter Check', and 'Production App Status Check'. The main area is titled 'All workflows' and shows a list of '1,233 workflow runs'. Each run is listed with a green checkmark, the name of the workflow, the event it triggered, its status, the branch it ran on, and the time it was run. For example, a 'Delete old workflow runs' run was scheduled 2 hours ago and completed 36s ago. Another run, 'Pull Request Reminder-Teams Channel', was scheduled 3 hours ago and completed 17s ago.

```
C:\xampp\htdocs\team-project-2021-s2-team-delta>npm run cx
> studio_management_app@1.0.0 cx
> cz
cz-cli@4.2.4, cz-conventional-changelog@3.2.0
) Select the type of change that you're committing:
refactor: A code change that improves performance
perf: A code change that improves performance
test: Adding missing tests or correcting existing tests
> build: Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)
ci: Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, Saucelabs)
chore: Other changes that don't modify src or test files
fix: Bumps package versions
(Move up and down to reveal more choices)
C:\xampp\htdocs\team-project-2021-s2-team-delta>npm run cx
> studio_management_app@1.0.0 cx
> cz
cz-cli@4.2.4, cz-conventional-changelog@3.2.0
```

```
#195, #181, #179
-> fix(students.blade.php): working on debugging issues with student output ...
currently the inconsistent use of cohort_id and paper_id means that there is no functional connection between students and papers that can be used to show students on a per paper basis on the home page. Have managed to get to displaying the name of each paper the user is currently teaching, but the list of students is duplicated for each due to different names used in each table. (I think, this is the issue). I have attempted to scrap the search student component to use the component added in the last sprint, this also has the same issue of not being able to access data.

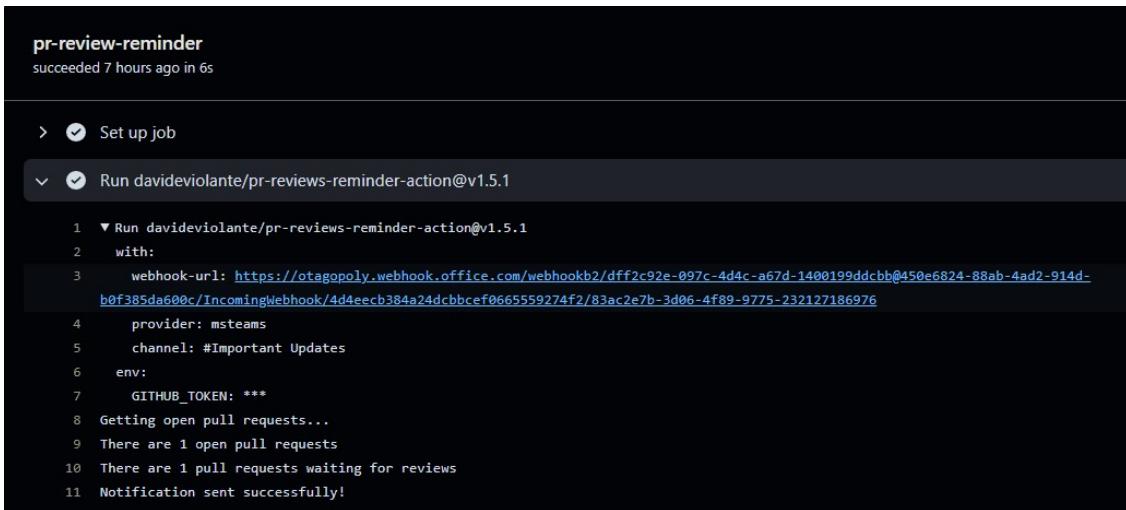
BREAKING CHANGE: currently the inconsistent use of cohort_id and paper_id means that there is no functional connection between students and papers that can be used to show students on a per paper basis on the home page. Have managed to get to displaying the name of each paper the user is currently teaching, but the list of students is duplicated for each due to different names used in each table. (I think, this is the issue). I have attempted to scrap the search student component to use the

#180, #181
```

1. Automated assignment of pull request [reviewers](#) and [assignees](#) - this was to stop this being forgotten about.



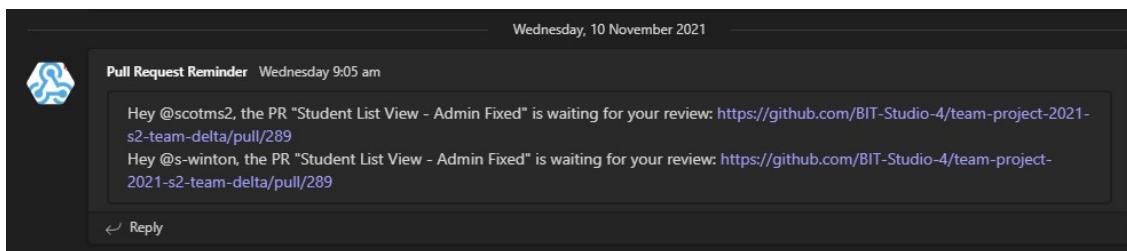
2. Configured GitHub Action to send [reminders to PR reviewers](#) if there are reviews that have not been completed. This is configured to send reminders to the project's teams channel using a web hook.



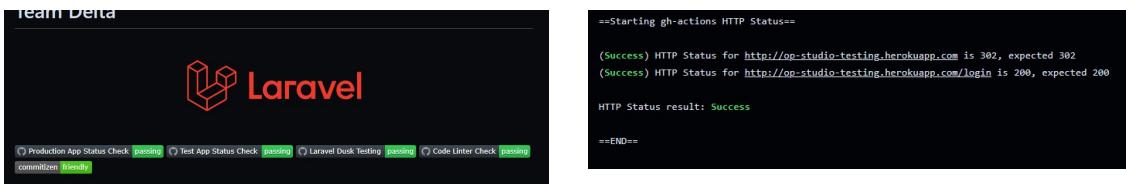
The screenshot shows a GitHub Actions run titled "pr-review-reminder" that succeeded 7 hours ago. It includes a "Set up job" step and a "Run davideviolante/pr-reviews-reminder-action@v1.5.1" step. The log output shows the action running, connecting to a webhook, and sending notifications about open pull requests and reviews.

```
pr-review-reminder
succeeded 7 hours ago in 6s

> Set up job
❯ Run davideviolante/pr-reviews-reminder-action@v1.5.1
  1 ▶ Run davideviolante/pr-reviews-reminder-action@v1.5.1
  2   with:
  3     webhook-url: https://otagopoly.webhook.office.com/webhookb2/dff2c92e-097c-4d4c-a67d-1400199ddccb@450e6824-88ab-4ad2-914d-
  4       b0f385da600c/IncomingWebhook/4d4eeb384a24dcbbcef066559274f2/83ac2e7b-3d06-4f89-9775-232127186976
  5     provider: msteams
  6     channel: #Important Updates
  7     env:
  8       GITHUB_TOKEN: ***
  9     Getting open pull requests...
 10    There are 1 open pull requests
 11    There are 1 pull requests waiting for reviews
 12    Notification sent successfully!
```



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](#)

leggant left a comment

I have a couple of questions about the permission options. Are these routes and forms to add and remove super users or are they to give the end-user the ability to add new sets of permissions? If it is the latter, my concern is that we appear to be leaving important configurations/set up decisions to our users, who might not know how to do this; potentially giving them more of a headache than they wanted after it is handed over.

Correct me if I have not completely understood something :)

leggant left a comment

Looks good, but just have a suggested changes to make

```
app/Http/Controllers/PaperController.php Outdated
120 +     $paper->delete();
121 +     return redirect('/papers')->with('success', 'Paper Deleted');
122 +
123 + }
```

Comment on lines 120 to 123

leggant 3 hours ago

Should this route be available? Wouldn't it be better to set an inactive attribute to the paper, that way related data does not lose paper information.

scotms2 13 minutes ago Author

Unsure what you mean?
\$paper is the paper that you are deleting

leggant 6 minutes ago • edited

should they be able to delete a paper? if deleted, previous records would lose having a paper assigned to them. it

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 legant merged 5 commits into [staging](#) from [sprint-3-update-homepage](#) 13 hours ago

Conversation 3 Commits 5 Checks 4 Files changed 11

legant commented 3 days ago • edited

Changes Close

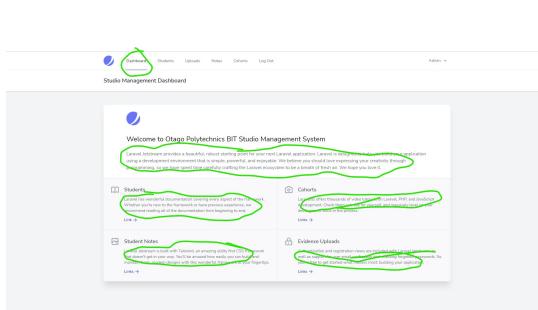
- Closes [Create Resource Controllers- Transfer Functions #5](#)
- Closes [Update Routes File #19](#)
- Closes [Dashboard Route Update #137](#)
- Closes [API Controller - Resource Controllers #21](#)

Changes made:

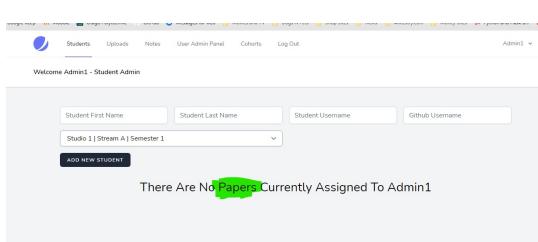
- altered the dashboard; removed default Laravel content so this is not present when the dashboard is re-activated in future.
- changed the home route to the student index page; commented out the Dashboard route in the nav and web route file

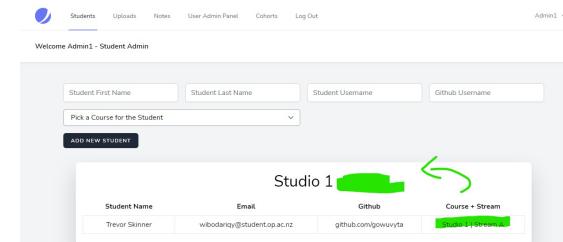
Outside of scope

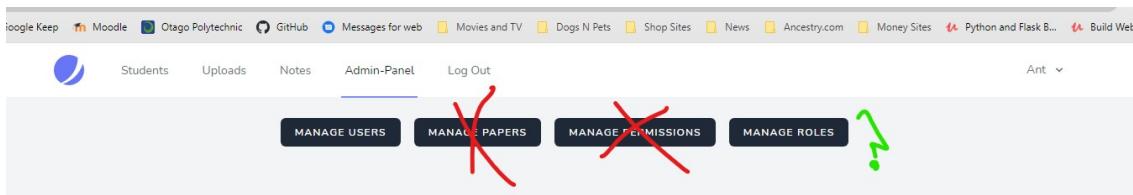
- Added a Note resource controller, transferred functions from api and pages controller
- updated all other controllers with methods from the api and pages controller
- changed the web routes to use these controller methods
- changed the form routes in cohort, note and evidence pages



| 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 generated 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 backlog 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 Transparency

Sorry guys, one thing I keep forgetting to bring up from the scrum meeting last week. Eric and Martin highlighted that there is not enough communication between each team back to the product owner (Eric) 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

Matthew Revill (0000073025) 1:09 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 Revill (0000073025) 1:14 pm
it's fine just needed to change the test for adding a student

Anthony Legg (03002726) 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, it's not something you can easily track on your own."

Matthew Revill (0000073025) 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 (03002726) 2:04 pm
actually that's a really good solution!!

2:07 pm
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

1